

**Graph Engine Service**

# **API Reference**

**Date**      2024-11-30

---

# Contents

---

|  |           |
|--|-----------|
| <b>1 Before You Start.....</b>                     | <b>1</b>  |
| 1.1 Overview.....                                  | 1         |
| 1.2 API Calling.....                               | 1         |
| 1.3 Constraints and Limitations on Using GES.....  | 1         |
| 1.3.1 Using Service Plane APIs.....                | 2         |
| 1.3.2 Naming OBS Objects.....                      | 2         |
| 1.4 Concepts.....                                  | 3         |
| 1.5 API Type or Version.....                       | 3         |
| <b>2 API Overview.....</b>                         | <b>4</b>  |
| 2.1 Management Plane APIs.....                     | 4         |
| 2.2 Service Plane API.....                         | 8         |
| 2.2.1 Memory Edition.....                          | 8         |
| 2.2.2 Database Edition.....                        | 17        |
| <b>3 Calling APIs.....</b>                         | <b>22</b> |
| 3.1 Making an API Request.....                     | 22        |
| 3.1.1 Making a Management Plane API Request.....   | 22        |
| 3.1.2 Making a Service Plane API Request.....      | 25        |
| 3.2 Authentication.....                            | 26        |
| 3.2.1 Authentication of Management Plane APIs..... | 26        |
| 3.2.2 Authentication of Service Plane APIs.....    | 27        |
| 3.3 Response.....                                  | 28        |
| <b>4 Management Plane APIs (V2).....</b>           | <b>30</b> |
| 4.1 System Management.....                         | 30        |
| 4.1.1 Querying Quotas.....                         | 30        |
| 4.2 Graph Management.....                          | 33        |
| 4.2.1 Querying the Graph List.....                 | 33        |
| 4.2.2 Querying Graph Details.....                  | 41        |
| 4.2.3 Creating a Graph.....                        | 48        |
| 4.2.4 Closing a Graph.....                         | 56        |
| 4.2.5 Starting a Graph.....                        | 58        |
| 4.2.6 Deleting a Graph.....                        | 61        |
| 4.2.7 Incrementally Importing Data to a Graph..... | 63        |

|   |            |
|---|------------|
| 4.2.8 Exporting a Graph.....                            | 68         |
| 4.2.9 Clearing a Graph.....                             | 71         |
| 4.2.10 Upgrading a Graph.....                           | 74         |
| 4.2.11 Binding an EIP.....                              | 77         |
| 4.2.12 Unbinding an EIP.....                            | 79         |
| 4.2.13 Resizing a Graph.....                            | 82         |
| 4.2.14 Restarting a Graph.....                          | 85         |
| 4.2.15 Expanding a Graph.....                           | 87         |
| 4.3 Backup Management.....                              | 90         |
| 4.3.1 Viewing the List of All Backups.....              | 90         |
| 4.3.2 Viewing the Backup List of a Graph.....           | 95         |
| 4.3.3 Adding a Backup.....                              | 99         |
| 4.3.4 Deleting a Backup.....                            | 101        |
| 4.3.5 Exporting a Backup.....                           | 103        |
| 4.3.6 Importing a Backup.....                           | 106        |
| 4.4 Metadata Management.....                            | 108        |
| 4.4.1 Constraints.....                                  | 108        |
| 4.4.2 Querying the Metadata List.....                   | 110        |
| 4.4.3 Querying Metadata.....                            | 113        |
| 4.4.4 Adding Metadata.....                              | 117        |
| 4.4.5 Deleting Metadata.....                            | 121        |
| 4.4.6 Importing Metadata from OBS.....                  | 123        |
| 4.5 Task Center.....                                    | 126        |
| 4.5.1 Querying Job Status on the Management Plane.....  | 126        |
| 4.5.2 Querying Job Details in the Job Center.....       | 131        |
| 4.6 Plugin Management.....                              | 137        |
| 4.6.1 Querying Scene Analysis Plugin Information.....   | 137        |
| 4.6.2 Subscribing to a Scene Analysis Plugin.....       | 141        |
| 4.6.3 Unsubscribing from a Scene Analysis Plugin.....   | 144        |
| <b>5 Service Plane APIs.....</b>                        | <b>148</b> |
| 5.1 Memory Edition.....                                 | 148        |
| 5.1.1 Vertex Operation APIs.....                        | 148        |
| 5.1.1.1 Querying Vertices Based on Filter Criteria..... | 148        |
| 5.1.1.2 Querying Vertex Details.....                    | 152        |
| 5.1.1.3 Adding a Vertex.....                            | 154        |
| 5.1.1.4 Deleting a Vertex.....                          | 157        |
| 5.1.1.5 Updating a Vertex Property.....                 | 159        |
| 5.1.1.6 Querying Vertices in Batches.....               | 161        |
| 5.1.1.7 Adding Vertices in Batches.....                 | 164        |
| 5.1.1.8 Deleting Vertices in Batches.....               | 167        |
| 5.1.1.9 Updating Vertex Properties in Batches.....      | 169        |
| 5.1.1.10 Adding a Vertex Label.....                     | 173        |

|  |     |
|--|-----|
| 5.1.1.11 Deleting a Vertex Label.....                | 175 |
| 5.1.1.12 Exporting Filtered Vertices.....            | 177 |
| 5.1.1.13 Deleting Filtered Vertices.....             | 180 |
| 5.1.2 Edge Operation APIs.....                       | 183 |
| 5.1.2.1 Querying Edges Based on Filter Criteria..... | 183 |
| 5.1.2.2 Querying Edge Details.....                   | 187 |
| 5.1.2.3 Adding an Edge.....                          | 190 |
| 5.1.2.4 Deleting an Edge.....                        | 194 |
| 5.1.2.5 Updating an Edge Property.....               | 196 |
| 5.1.2.6 Querying Edges in Batches.....               | 200 |
| 5.1.2.7 Adding Edges in Batches.....                 | 202 |
| 5.1.2.8 Deleting Edges in Batches.....               | 208 |
| 5.1.2.9 Updating Edge Properties in Batches.....     | 212 |
| 5.1.2.10 Exporting Filtered Edges.....               | 215 |
| 5.1.2.11 Deleting Filtered Edges.....                | 218 |
| 5.1.3 Metadata Operation APIs.....                   | 220 |
| 5.1.3.1 Adding a Label.....                          | 220 |
| 5.1.3.2 Updating a Label.....                        | 225 |
| 5.1.3.3 Querying Graph Metadata Details.....         | 229 |
| 5.1.3.4 Changing Property Names in Batches.....      | 233 |
| 5.1.3.5 Deleting a Label.....                        | 235 |
| 5.1.3.6 Adding Labels in Batches.....                | 237 |
| 5.1.3.7 Querying Schema Structure.....               | 241 |
| 5.1.3.8 Generating a Schema Structure.....           | 244 |
| 5.1.3.9 Generating Data Assets.....                  | 246 |
| 5.1.3.10 Obtaining Data Assets.....                  | 248 |
| 5.1.4 Index Operation APIs.....                      | 250 |
| 5.1.4.1 Creating an Index.....                       | 250 |
| 5.1.4.2 Deleting an Index.....                       | 254 |
| 5.1.4.3 Querying Indexes.....                        | 256 |
| 5.1.5 Gremlin Operation APIs.....                    | 258 |
| 5.1.5.1 Executing a Gremlin Query.....               | 258 |
| 5.1.6 Algorithm APIs.....                            | 260 |
| 5.1.6.1 Executing an Algorithm.....                  | 260 |
| 5.1.6.2 Algorithm API Parameter References.....      | 263 |
| 5.1.6.2.1 Common Algorithm Parameters.....           | 263 |
| 5.1.6.2.2 PageRank.....                              | 268 |
| 5.1.6.2.3 PersonalRank.....                          | 269 |
| 5.1.6.2.4 K-core.....                                | 271 |
| 5.1.6.2.5 K-Hop.....                                 | 271 |
| 5.1.6.2.6 Common Neighbors.....                      | 272 |
| 5.1.6.2.7 Common Neighbors of Vertex Sets.....       | 273 |

|   |     |
|---|-----|
| 5.1.6.2.8 Link Prediction.....                          | 275 |
| 5.1.6.2.9 Shortest Path.....                            | 275 |
| 5.1.6.2.10 All Shortest Paths.....                      | 277 |
| 5.1.6.2.11 Filtered Shortest Path.....                  | 278 |
| 5.1.6.2.12 SSSP.....                                    | 281 |
| 5.1.6.2.13 Shortest Path of Vertex Set.....             | 282 |
| 5.1.6.2.14 n-Paths.....                                 | 283 |
| 5.1.6.2.15 Filtered n-Paths.....                        | 284 |
| 5.1.6.2.16 Filtered All Pairs Shortest Paths.....       | 287 |
| 5.1.6.2.17 All Shortest Paths of Vertex Sets.....       | 289 |
| 5.1.6.2.18 Filtered All Shortest Paths.....             | 290 |
| 5.1.6.2.19 Connected Component.....                     | 292 |
| 5.1.6.2.20 Label Propagation.....                       | 293 |
| 5.1.6.2.21 Louvain.....                                 | 294 |
| 5.1.6.2.22 Node2vec.....                                | 295 |
| 5.1.6.2.23 Real-time Recommendation.....                | 296 |
| 5.1.6.2.24 Degree Correlation.....                      | 298 |
| 5.1.6.2.25 Triangle Count.....                          | 299 |
| 5.1.6.2.26 Cluster Coefficient.....                     | 299 |
| 5.1.6.2.27 Closeness Centrality.....                    | 300 |
| 5.1.6.2.28 Filtered Circle Detection.....               | 300 |
| 5.1.6.2.29 Subgraph Matching.....                       | 303 |
| 5.1.6.2.30 Topicrank.....                               | 305 |
| 5.1.7 Temporal Graph APIs.....                          | 306 |
| 5.1.7.1 Community Evolution (temporal_graph).....       | 306 |
| 5.1.7.2 Temporal BFS (temporal_bfs).....                | 309 |
| 5.1.7.3 Temporal Paths.....                             | 313 |
| 5.1.8 Path APIs.....                                    | 316 |
| 5.1.8.1 Querying Path Details.....                      | 316 |
| 5.1.8.2 Querying Tree Details.....                      | 320 |
| 5.1.8.3 Repeat-query API.....                           | 324 |
| 5.1.9 Graph Statistics APIs.....                        | 331 |
| 5.1.9.1 Querying General Information About a Graph..... | 331 |
| 5.1.9.2 Querying the Graph Version.....                 | 335 |
| 5.1.10 Graph Operation APIs.....                        | 337 |
| 5.1.10.1 Importing a Graph.....                         | 337 |
| 5.1.10.2 Exporting a Graph.....                         | 341 |
| 5.1.10.3 Clearing a Graph.....                          | 343 |
| 5.1.11 Subgraph Operation APIs.....                     | 345 |
| 5.1.11.1 Querying a Subgraph.....                       | 346 |
| 5.1.11.2 Executing an Algorithm on a Subgraph.....      | 349 |
| 5.1.12 Job Management APIs.....                         | 352 |

|   |     |
|---|-----|
| 5.1.12.1 Querying Job Status on the Service Plane.....                              | 352 |
| 5.1.12.2 Canceling a Job.....   | 355 |
| 5.1.12.3 Exporting Job Execution Results to Files.....                              | 357 |
| 5.1.12.4 Querying the Job List.....   | 363 |
| 5.1.13 Custom Operations APIs.....  | 366 |
| 5.1.13.1 Performing Custom Operations.....  | 366 |
| 5.1.14 Cypher Operation APIs.....   | 369 |
| 5.1.14.1 Executing Cypher Queries.....  | 369 |
| 5.1.14.2 Cypher Prerequisites.....  | 375 |
| 5.1.14.3 Basic Operations and Compatibility.....                                    | 375 |
| 5.1.14.4 Supported Expressions, Functions, and Procedures.....                      | 380 |
| 5.1.15 Filtered-query API.....  | 390 |
| 5.1.16 Filtered-query V2.....   | 402 |
| 5.1.17 Domain-Specific Language (DSL) Query APIs.....                               | 412 |
| 5.1.17.1 Executing the DSL Algorithm.....   | 412 |
| 5.1.17.2 DSL Syntax.....  | 415 |
| 5.1.18 Updating Specified Properties of Vertices and Edges by Importing a File..... | 428 |
| 5.1.19 Deleting Vertices and Edges by Reading Files.....                            | 433 |
| 5.1.20 Granular Permission Control APIs.....  | 436 |
| 5.1.20.1 Authorization.....   | 436 |
| 5.1.20.2 Canceling Authorization.....   | 440 |
| 5.1.20.3 Querying Authorization.....  | 443 |
| 5.1.21 O&M Monitoring APIs.....   | 444 |
| 5.1.21.1 Viewing Monitoring Metrics.....  | 444 |
| 5.1.21.2 Viewing Real-Time Requests.....  | 450 |
| 5.2 Database Edition.....   | 453 |
| 5.2.1 Specification Description.....  | 453 |
| 5.2.2 Vertex Operation APIs.....  | 455 |
| 5.2.2.1 Querying Vertex Details.....  | 455 |
| 5.2.2.2 Querying Vertices in Batches.....   | 457 |
| 5.2.2.3 Adding Vertices in Batches.....   | 460 |
| 5.2.2.4 Deleting Vertices in Batches.....   | 463 |
| 5.2.2.5 Updating Vertex Properties in Batches.....                                  | 465 |
| 5.2.3 Edge Operation APIs.....  | 468 |
| 5.2.3.1 Querying Edge Details.....  | 468 |
| 5.2.3.2 Querying Edges in Batches.....  | 471 |
| 5.2.3.3 Adding Edges in Batches.....  | 474 |
| 5.2.3.4 Deleting Edges in Batches.....  | 477 |
| 5.2.3.5 Updating Edge Properties in Batches.....                                    | 480 |
| 5.2.4 Metadata Operation APIs.....  | 483 |
| 5.2.4.1 Adding a Label.....   | 483 |
| 5.2.4.2 Updating a Label.....   | 487 |

|   |     |
|---|-----|
| 5.2.4.3 Querying Labels.....  | 490 |
| 5.2.4.4 Querying Graph Metadata Details.....  | 493 |
| 5.2.4.5 Generating Data Assets.....   | 496 |
| 5.2.4.6 Obtaining Data Assets.....  | 498 |
| 5.2.5 Index Operation APIs.....   | 500 |
| 5.2.5.1 Creating an Index.....  | 501 |
| 5.2.5.2 Deleting an Index.....  | 504 |
| 5.2.5.3 Querying Indexes.....   | 506 |
| 5.2.5.4 Creating Indexes in Batches.....  | 508 |
| 5.2.6 HyG Graph Management APIs.....  | 512 |
| 5.2.6.1 Creating a HyG Graph.....   | 512 |
| 5.2.6.2 Synchronizing HyG Graph Data.....   | 515 |
| 5.2.6.3 Querying General Information About a HyG Graph.....   | 517 |
| 5.2.6.4 Deleting a HyG Graph.....   | 520 |
| 5.2.6.5 Listing HyG Graphs.....   | 522 |
| 5.2.6.6 Importing an HyG Graph.....   | 525 |
| 5.2.7 HyG Algorithm APIs.....   | 529 |
| 5.2.7.1 Running Algorithms.....   | 529 |
| 5.2.7.2 Algorithm API Parameter Reference.....  | 531 |
| 5.2.7.2.1 Common Algorithm Parameters.....  | 531 |
| 5.2.7.2.2 Shortest Path.....  | 532 |
| 5.2.7.2.3 SSSP.....   | 535 |
| 5.2.7.2.4 K-Hop.....  | 538 |
| 5.2.7.2.5 PageRank.....   | 541 |
| 5.2.7.2.6 Connected Component.....  | 544 |
| 5.2.7.2.7 K-core.....   | 546 |
| 5.2.7.3 Algorithm Results in CSV Format.....  | 549 |
| 5.2.7.4 Executing the DSL Algorithm.....  | 565 |
| 5.2.7.5 DSL Syntax.....   | 567 |
| 5.2.7.5.1 Graph Operation APIs.....   | 567 |
| 5.2.7.5.2 API for Running Custom Algorithms (Currently, the Pregel Programming Model Is Supported)..... | 568 |
| 5.2.7.5.3 Pregel Programming API.....   | 568 |
| 5.2.7.5.4 Programming Example of Creating Custom Graph Analysis Algorithms.....                         | 569 |
| 5.2.8 HyG Job Management APIs.....  | 571 |
| 5.2.8.1 Dumping HyG Algorithm Results.....  | 571 |
| 5.2.8.2 Canceling a HyG Job.....  | 574 |
| 5.2.9 Native Algorithm APIs.....  | 576 |
| 5.2.9.1 Executing an Algorithm.....   | 576 |
| 5.2.9.2 Algorithm API Parameter References.....   | 578 |
| 5.2.9.2.1 Common Algorithm Parameters.....  | 578 |
| 5.2.9.2.2 Shortest Path.....  | 581 |
| 5.2.9.2.3 Shortest Path of Vertex Set.....  | 582 |

|  |            |
|--|------------|
| 5.2.9.2.4 Common Neighbors of Vertex Sets.....                 | 584        |
| 5.2.10 Graph Statistics APIs.....                              | 586        |
| 5.2.10.1 Querying General Information About a Graph.....       | 586        |
| 5.2.10.2 Querying the Graph Version.....                       | 588        |
| 5.2.11 Graph Operation APIs.....                               | 590        |
| 5.2.11.1 Importing a Graph.....                                | 590        |
| 5.2.11.2 Clearing a Graph.....                                 | 594        |
| 5.2.11.3 Exporting a Graph.....                                | 596        |
| 5.2.11.4 Creating a Graph.....                                 | 600        |
| 5.2.11.5 Deleting a Graph.....                                 | 602        |
| 5.2.11.6 Listing Graphs.....                                   | 604        |
| 5.2.12 Job Management APIs.....                                | 606        |
| 5.2.12.1 Querying the Job List.....                            | 606        |
| 5.2.12.2 Querying Job Status.....                              | 609        |
| 5.2.13 Cypher Operation APIs.....                              | 612        |
| 5.2.13.1 Executing Cypher Queries.....                         | 613        |
| 5.2.13.2 Basic Operations and Compatibility.....               | 619        |
| 5.2.13.3 Supported Expressions, Functions, and Procedures..... | 624        |
| <b>6 Application Examples.....</b>                             | <b>634</b> |
| 6.1 Analyzing Graphs Using HyG API Algorithms.....             | 634        |
| <b>7 Monitoring Metrics.....</b>                               | <b>637</b> |
| <b>8 Out-of-Date APIs.....</b>                                 | <b>644</b> |
| 8.1 Management Plane APIs (V1).....                            | 644        |
| 8.1.1 System Management APIs.....                              | 644        |
| 8.1.1.1 Querying Quotas.....                                   | 644        |
| 8.1.2 Graph Management APIs.....                               | 647        |
| 8.1.2.1 Querying the Graph List.....                           | 647        |
| 8.1.2.2 Querying Graph Details.....                            | 653        |
| 8.1.2.3 Creating a Graph.....                                  | 659        |
| 8.1.2.4 Closing a Graph.....                                   | 666        |
| 8.1.2.5 Starting a Graph.....                                  | 669        |
| 8.1.2.6 Deleting a Graph.....                                  | 671        |
| 8.1.2.7 Incrementally Importing Data to a Graph.....           | 673        |
| 8.1.2.8 Exporting a Graph.....                                 | 678        |
| 8.1.2.9 Clearing a Graph.....                                  | 681        |
| 8.1.2.10 Upgrading a Graph.....                                | 683        |
| 8.1.2.11 Binding an EIP.....                                   | 686        |
| 8.1.2.12 Unbinding an EIP.....                                 | 689        |
| 8.1.2.13 Resizing a Graph.....                                 | 691        |
| 8.1.2.14 Restarting a Graph.....                               | 694        |
| 8.1.2.15 Expanding a Graph.....                                | 696        |



|  |            |
|--|------------|
| 8.1.3 Backup Management APIs.....                        | 699        |
| 8.1.3.1 Viewing the List of All Backups.....             | 699        |
| 8.1.3.2 Viewing the Backup List of a Graph.....          | 703        |
| 8.1.3.3 Adding a Backup.....                             | 707        |
| 8.1.3.4 Deleting a Backup.....                           | 709        |
| 8.1.4 Metadata Management APIs.....                      | 711        |
| 8.1.4.1 Constraints.....                                 | 711        |
| 8.1.4.2 Querying the Metadata List.....                  | 713        |
| 8.1.4.3 Querying Metadata.....                           | 716        |
| 8.1.4.4 Adding Metadata.....                             | 719        |
| 8.1.4.5 Deleting Metadata.....                           | 723        |
| 8.1.4.6 Importing Metadata from OBS.....                 | 725        |
| 8.1.5 Task Center APIs.....                              | 727        |
| 8.1.5.1 Querying Job Status on the Management Plane..... | 728        |
| 8.1.5.2 Querying Job Details in the Job Center.....      | 732        |
| <b>9 Appendix.....</b>                                   | <b>739</b> |
| 9.1 Status Codes.....                                    | 739        |
| 9.2 Error Codes.....                                     | 743        |
| 9.2.1 Error Codes for Management Plane APIs.....         | 743        |
| 9.2.2 Error Codes for Service Plane APIs.....            | 750        |
| 9.3 Obtaining a Project ID.....                          | 759        |
| 9.4 Obtaining the Account Name and Account ID.....       | 760        |

# 1 Before You Start

---

## 1.1 Overview

Welcome to *Graph Engine Service API Reference*. Graph Engine Service (GES) is the first commercial self-built distributed native graph engine with independent intellectual property rights in the industry. It facilitates querying and analysis of graph structure data based on relationships. It is specifically suited for scenarios involving social applications, enterprise relationship analysis, risk control, recommendations, public opinions, and anti-fraud.

This document describes how to use application programming interfaces (APIs) to perform operations on GES resources.

- Management Plane APIs

Management plane APIs provide graph management functions, including creating, stopping, starting, restoring, and upgrading graphs, importing, exporting, and clearing data, creating, querying, and deleting graph backups, and managing metadata. You need to call the management plane APIs to perform these operations.

- Service Plane APIs

Service plane APIs provide graph service functions, including adding, deleting, querying, and modifying vertices, edges, and metadata files, performing Gremlin queries, and running algorithms. You need to call the service plane APIs to perform these operations.

Before calling APIs of GES, ensure that you are familiar with GES concepts.

## 1.2 API Calling

GES supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Making an API Request](#).

## 1.3 Constraints and Limitations on Using GES

### 1.3.1 Using Service Plane APIs

You can access the service plane APIs by anyone of the following methods:

- Through the ECS. The VPC for creating the ECS must be the same as that selected during graph creation. If the same security group is selected, you can directly access the APIs. If the security groups are not the same, enable the access permission for the ECS in the security group where the graph is created. That is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. The ingress and egress ports support HTTP and HTTPS access respectively. In this scenario, **SERVER\_URL** of the APIs is the private access address in the graph details on the GES console or the value of the **privatelp** field in the response body of the management plane API for querying graph details.
- Through the ECS. The VPC for creating the ECS is not the same as that selected during graph creation. You need to create a VPC peering connection between the VPC to which the ECS belongs and the VPC in which the graph is created. In addition, you need to enable the access permission for the ECS in the security group where the graph is created. That is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. In this scenario, **SERVER\_URL** of the APIs is the private access address in the graph details on the GES console or the value of the **privatelp** field in the response body of the management plane API for querying graph details.
- Through the public network. You need to create an EIP and enable the access permission for the client in the security group where the graph is created, that is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. In this scenario, **SERVER\_URL** of the APIs is the public access address in the graph details on the GES console or the value of the **publiclp** field in the response body of the management plane API for querying graph details (also the EIP you bind or create).

### 1.3.2 Naming OBS Objects

The OBS object names supported by GES can contain the following characters:

|                    |                 |
|--------------------|-----------------|
| Letters and digits | 0-9, a-z, A-Z   |
| Special characters | ! - _ . * ' ( ) |

The following characters are not supported:

|                           |   |
|---------------------------|---|
| <b>Special characters</b> | <b>\{\^}\%`]"&gt;[~&lt;# &amp;@:,\$=+? and spaces</b>   |
| ASCII control characters  | Range: <ul style="list-style-type: none"> <li>• 00-1F in hexadecimal form (0-31 in decimal form)</li> <li>• 7F (127 in decimal form)</li> </ul> |

## 1.4 Concepts

- **User**  
A user is created in Identity and Access Management (IAM) to use cloud services. Each user has its own identity credentials (password and access keys).  
The account name, username, and password will be required for API authentication.
- **Region**  
Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.
- **AZ**  
An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- **Project**  
Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each cloud region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and apply for resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

## 1.5 API Type or Version

The GES API version corresponds to the software version. 2.2.17 is the start version number. Other versions are updated based on the start version and are backward compatible.

# 2 API Overview

## 2.1 Management Plane APIs

GES management plane APIs comprise APIs for system management, graph management, backup management, metadata management, task center, and plugin management.

**Table 2-1** System management APIs

| Parameter                       | URL                                | Function  |
|---------------------------------|------------------------------------|---|
| <a href="#">Querying Quotas</a> | GET /v2/{project_id}/graphs/quotas | Query the quota of graphs, edges, or backups. When creating or backing up a graph, you can call this API to view the quota. |

**Table 2-2** Graph management APIs

| Parameter                               | URL                                    | Function  |
|---|--|---|
| <a href="#">Querying the Graph List</a> | GET /v2/{project_id}/graphs            | Query the list of all created graphs.   |
| <a href="#">Querying Graph Details</a>  | GET /v2/{project_id}/graphs/{graph_id} | Query details about a graph, including the graph's private and public access addresses, version number, and imported vertex and edge data sets. |

| Parameter   | URL  | Function  |
|---|--|---|
| <a href="#">Creating a Graph</a>                        | POST /v2/{project_id}/graphs                         | Create a graph after you define the metadata and vertex and edge data sets of the graph.  |
| <a href="#">Stopping a Graph</a>                        | POST /v2/{project_id}/graphs/{graph_id}/stop         | You can stop a graph at any time because service continuity is not required.  |
| <a href="#">Starting a Graph</a>                        | POST /v2/{project_id}/graphs/{graph_id}/start        | If you want to use a stopped graph, you can restore the data to the state it was before the shutdown or to a backup time point.           |
| <a href="#">Deleting a Graph</a>                        | DELETE /v2/{project_id}/graphs/{graph_id}            | Delete a graph when you do not need it.   |
| <a href="#">Incrementally Importing Data to a Graph</a> | POST /v2/{project_id}/graphs/{graph_id}/import-graph | You need to incrementally import graph data in batches.   |
| <a href="#">Exporting a Graph</a>                       | POST /v2/{project_id}/graphs/{graph_id}/export-graph | You need to export all the data in a graph as a text file.  |
| <a href="#">Clearing a Graph</a>                        | POST /v2/{project_id}/graphs/{graph_id}/clear-graph  | You need to clear all the data in a graph, including the vertex and edge data.<br><b>NOTE</b><br>The metadata will not be cleared.        |
| <a href="#">Upgrading a Graph</a>                       | POST /v2/{project_id}/graphs/{graph_id}/upgrade      | If bugs are detected in a graph of the early version or new functions need to be added, you need to upgrade the graph to the new version. |
| <a href="#">Binding an EIP</a>                          | POST /v2/{project_id}/graphs/{graph_id}/bind-eip     | To access a graph over the public network, you need to bind an EIP.   |
| <a href="#">Unbinding an EIP</a>                        | POST /v2/{project_id}/graphs/{graph_id}/unbind-eip   | You can unbind an EIP from a graph if you no longer need to access the graph over the public network.                                     |
| <a href="#">Resizing a Graph</a>                        | POST /v2/{project_id}/graphs/{graph_id}/resize       | Resize a graph instance.  |

| Parameter                          | URL   | Function   |
|------------------------------------|---|--|
| <b>Forcibly Restarting a Graph</b> | POST /v2/{project_id}/graphs/{graph_id}/restart | Forcibly start a graph.  |
| <b>Expanding a Graph</b>           | POST /v2/{project_id}/graphs/{graph_id}/expand  | This API is used to expand multiple secondary nodes dynamically. The expanded secondary nodes can process read requests, improving read performance. |

**Table 2-3** Backup management APIs

| Parameter                                 | URL   | Function   |
|---|---|--|
| <b>Viewing the List of All Backups</b>    | GET /v2/{project_id}/graphs/backups                           | View all backup details of all graphs.   |
| <b>Viewing the Backup List of a Graph</b> | GET /v2/{project_id}/graphs/{graph_id}/backups                | View details about all backups of a graph, including the backup start time and end time.   |
| <b>Adding a Backup</b>                    | POST /v2/{project_id}/graphs/{graph_id}/backups               | Backup is used to improve data reliability. It can also be used as a snapshot of a graph for you to restore data when necessary. |
| <b>Deleting a Backup</b>                  | DELETE /v2/{project_id}/graphs/{graph_id}/backups/{backup_id} | Delete backups of a graph.   |
| <b>Exporting a Backup</b>                 | POST /v2/{project_id}/graphs/{graph_id}/backups/export        | Export a GES graph instance backup to OBS.   |
| <b>Importing a Backup</b>                 | POST /v2/{project_id}/graphs/{graph_id}/backups/import        | Import a GES graph instance backup from OBS.   |

**Table 2-4** Metadata management APIs

| Parameter                          | URL  | Function   |
|------------------------------------|--|--|
| <b>Querying the Metadata List</b>  | GET /v2/{project_id}/graphs/metadatas                  | Query details about all metadata files, including the status and OBS storage path. |
| <b>Querying Metadata</b>           | GET /v2/{project_id}/graphs/metadatas/{metadata_id}    | Query details about a graph metadata file.   |
| <b>Adding Metadata</b>             | POST /v2/{project_id}/graphs/metadatas                 | Add metadata to prepare for creating a graph.                                      |
| <b>Deleting Metadata</b>           | DELETE /v2/{project_id}/graphs/metadatas/{metadata_id} | Delete a metadata file.  |
| <b>Importing Metadata from OBS</b> | POST /v2/{project_id}/graphs/metadata/upload-from-obs  | Import metadata from OBS.  |

**Table 2-5** Task center APIs

| Parameter                                     | URL   | Function  |
|---|---|---|
| <b>Querying the Job Status</b>                | GET /v2/{project_id}/graphs/{graph_id}/jobs/{job_id}/status | Graph deleting, stopping, starting, restoring, incrementally importing, clearing, and upgrading are asynchronous jobs initialized by calling these APIs. These APIs return the job IDs. You can view the execution status of each asynchronous job through the corresponding API. |
| <b>Querying Job Details in the Job Center</b> | GET /v2/{project_id}/graphs/jobs                            | View all asynchronous jobs.   |



**Table 2-6** Plugin management APIs

| API   | URL   | Function  |
|---|---|---|
| <b>Querying Scene Analysis Plugin Information</b> | GET /v2/{project_id}/graphs/scenes                        | Query information about the application analysis capability in a scene, including information about the applications, parameters, and function details.               |
| <b>Subscribing to a Scene Analysis Plugin</b>     | POST /v2/{project_id}/graphs/{graph_id}/scenes/register   | This API is used to subscribe to a scene analysis plugin so that you can use the function through the service plane APIs.   |
| <b>Unsubscribing from a Scene Analysis Plugin</b> | POST /v2/{project_id}/graphs/{graph_id}/scenes/unregister | This API is used to unsubscribe from a scene analysis plugin. After the subscription is canceled, you cannot use the function through application service plane APIs. |

## 2.2 Service Plane API

### 2.2.1 Memory Edition

Memory edition service plane APIs cover vertex operations, edge operations, metadata operations, index operations, Gremlin operations, algorithms, paths, graph statistics, graph and subgraph operations, job management, and Cypher operations.

**Table 2-7** Vertex operation APIs

| API  | URL  | Function  |
|--|--|---|
| <b>Querying Vertices That Meet Filter Criteria</b> | POST/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query       | Query vertices based on a filtering criterion. For example, if the vertex metadata contains the age property, the filter criterion can be 'age > 18'. |
| <b>Querying Vertex Details</b>                     | GET/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds={vertex_ids} | Query details about a specified vertex of a specified set of vertices, including the label information.   |
| <b>Adding a Vertex</b>                             | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices                             | Add a vertex.   |

| API  | URL  | Function   |
|--|--|--|
| <b>Deleting a Vertex</b>                     | DELETE/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}                                      | Delete a vertex.   |
| <b>Updating Vertex Properties</b>            | POST/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/properties/action?action_id={actionId} | Modify a vertex's properties, including adding, changing, and deleting properties. |
| <b>Querying Vertex Data in Batches</b>       | POST/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query                       | Query vertex details in batches.   |
| <b>Adding Vertices in Batches</b>            | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add                        | Add vertices in batches.   |
| <b>Deleting Vertices in Batches</b>          | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete                     | Delete vertices in batches based on the vertex IDs.                                |
| <b>Updating Vertex Properties in Batches</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?action_id={actionId}            | Update vertex properties in batches.   |
| <b>Adding a Vertex Label</b>                 | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels                                | Add a vertex label.  |
| <b>Deleting a Vertex Label</b>               | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels/{label_name}                 | Delete a vertex label.   |
| <b>Exporting Filtered Vertices</b>           | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export                           | Export the vertex set that meets the filter criteria.                              |
| <b>Deleting Filtered Vertices</b>            | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete                           | Delete the vertex set that meets the filter criteria.                              |

**Table 2-8** Edge operation APIs

| API  | URL   | Function   |
|--|---|--|
| <b>Querying Vertices That Meet Filter Criteria</b> | POST/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query   | Filter the edges that meet the filter criteria of edge properties.   |
| <b>Querying Edge Details</b>                       | GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?source={sourceVertex}&target={targetVertex}&index={index}                                 | Query the details about an edge based on its source and target vertices, including the edge's label information. |
| <b>Adding an Edge</b>                              | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges   | Add an edge.   |
| <b>Deleting an Edge</b>                            | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/edges?source={sourceVertex}&target={targetVertex}&index={index}                                     | Delete an edge.  |
| <b>Updating Edge Properties</b>                    | POST/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}&source={sourceVertex}&target={targetVertex}&index={index} | Modify an edge's properties, including adding, changing, and deleting properties.                                |
| <b>Querying Edge Data in Batches</b>               | POST/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query   | Query edge details in batches.   |
| <b>Adding Edges in Batches</b>                     | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add  | Add edges in batches.  |
| <b>Deleting Edges in Batches</b>                   | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete   | Delete edges in batches based on the source vertices, target vertices, and indexes of the edges.                 |
| <b>Updating Edge Properties in Batches</b>         | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}  | Update edge properties in batches.   |
| <b>Exporting Filtered Edges</b>                    | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export   | Export the edge set that meets the filter criteria.  |

| API                            | URL   | Function  |
|--------------------------------|---|---|
| <b>Deleting Filtered Edges</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete | Delete the edge set that meets the filter criteria. |

**Table 2-9** Metadata operation APIs

| API                                    | URL  | Function  |
|--|--|---|
| <b>Adding a Label</b>                  | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels                            | Add a label.  |
| <b>Updating a Label</b>                | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties    | Update a label.   |
| <b>Querying Graph Metadata Details</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema                                    | Query metadata details.   |
| <b>Deleting a Label</b>                | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{labelName}              | Delete a label as well as the vertices and edges associated with the label.   |
| <b>Adding Labels in Batches</b>        | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add | Add labels in batches.  |
| <b>Querying a Schema</b>               | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure?detail={details}         | Query a generated schema (obtained from OBS).   |
| <b>Generating a Schema</b>             | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build                   | Generate a schema where labels are represented with vertices and the relationship between the labels are represented with edges, and store the schema in an OBS bucket. |
| <b>Generating Data Assets</b>          | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets                       | Generate data assets.   |
| <b>Obtaining Data Assets</b>           | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets                        | Obtain data assets.   |

**Table 2-10** Index operation APIs

| API                      | URL   | Function         |
|--------------------------|---|------------------|
| <b>Creating an Index</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices               | Create an index. |
| <b>Deleting an Index</b> | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName} | Delete an index. |
| <b>Querying Indexes</b>  | GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices                | Query indexes.   |

**Table 2-11** Gremlin operation APIs

| API                              | URL  | Function                 |
|----------------------------------|--|--------------------------|
| <b>Executing Gremlin Queries</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query | Execute Gremlin queries. |

**Table 2-12** Algorithm APIs

| Operation                | URL  | Function        |
|--------------------------|--|-----------------|
| <b>Runing Algorithms</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm | Run algorithms. |

**Table 2-13** Temporal graph APIs

| API   | URL   | Function  |
|---|---|---|
| <b>Community Evolution (temporal_graph)</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis | The community evolution algorithm generates a temporal graph that shows structure changes of a community over time.   |
| <b>Temporal BFS (temporal_bfs)</b>          | POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis | This algorithm searches for associated vertices based on temporal message passing and temporal BFS algorithms, and outputs the visit time of each vertex and the distance from the vertex to the source start vertex. |

| API                   | URL   | Function   |
|-----------------------|---|--|
| <b>Temporal Paths</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis | Execute the temporal paths algorithm based on input parameters. Only one temporal path that meets the conditions is returned between two vertices. |

**Table 2-14** Path APIs

| Parameter                    | URL   | Function  |
|------------------------------|---|---|
| <b>Querying Path Details</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail | Query the path details.   |
| <b>Querying Tree Details</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=query-tree         | After you input a vertex, a tree starting from the vertex is output, including nodes and edges in the tree, and information about reachable paths. Parameters can be the direction (out, in, and out and in), maximum number of hops, and edge properties to be filtered. |
| <b>Repeat-query API</b>      | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=repeat-query       | Cyclically perform traversal query based on certain exit conditions.  |

**Table 2-15** Graph statistics APIs

| Operation   | URL  | Function                                 |
|---|--|--|
| <b>Querying General Information About a Graph</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary | Query general information about a graph. |
| <b>Querying the Graph Version</b>                 | GET /ges/v1.0/{project_id}/graphs/{graph_name}/version | Query the graph version.                 |

**Table 2-16** Graph operation APIs

| Operation                | URL   | Function     |
|--------------------------|---|--------------|
| <b>Importing a Graph</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph | Import data. |
| <b>Exporting a Graph</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph | Export data. |
| <b>Clearing a Graph</b>  | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph  | Clear data.  |

**Table 2-17** Subgraph operation APIs

| Operation                                   | URL  | Function   |
|---|--|--|
| <b>Querying a Subgraph</b>                  | POST/ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?action_id=query              | Query the subgraphs that are created by the specified vertices and the edges connecting them.            |
| <b>Executing an Algorithm on a Subgraph</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?action_id=execute-algorithm | Adjust the subgraph creation type based on the input and execute an algorithm on the generated subgraph. |

**Table 2-18** Job management APIs

| Operation                                       | URL   | Function  |
|---|---|---|
| <b>Querying Task Status</b>                     | GET/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset= <i>offset</i> &limit= <i>limit</i> | Query job status.   |
| <b>Canceling a Job</b>                          | DELETE https://Endpoint/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}                               | Cancel a job.   |
| <b>Exporting Job Execution Results to Files</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result                  | Export the execution result ( <b>result</b> ) of an asynchronous job ( <b>job_id</b> ) to a file. |

| Operation                    | URL  | Function  |
|------------------------------|--|---|
| <b>Querying the Job List</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset} | After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned. |

**Table 2-19** Custom operations APIs

| API                                 | URL  | Function              |
|-------------------------------------|--|-----------------------|
| <b>Performing Custom Operations</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-operation | Query the job status. |

**Table 2-20** Fine-grained permission control APIs

| API                         | URL  | Function  |
|-----------------------------|--|---|
| <b>Granting Permissions</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/rbac/action?action_id=grant  | Grant fine-grained permissions.                               |
| <b>Revoking Permissions</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/rbac/action?action_id=revoke | Revoke permissions.   |
| <b>Querying Permissions</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/rbac                          | Query all label and property permissions of the current user. |

**Table 2-21** O&M monitoring APIs

| API                               | URL  | Function   |
|-----------------------------------|--|--|
| <b>Viewing Monitoring Metrics</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?real_time=&with_performance_metrics= | View monitoring metrics, including node metrics and graph instance performance monitoring metrics. |



| API                               | URL   | Function   |
|-----------------------------------|---|--|
| <b>Viewing Real-Time Requests</b> | GET/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries?summary= | View the real-time requests on the current primary node. |

**Table 2-22** Other service plane APIs

| Operation  | URL   | Function   |
|--|---|--|
| <b>Cypher Operation APIs</b>   | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query | Run Cypher statements to query data in GES and obtain results.   |
| <b>Filtered-query API</b>  | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query       | Filter the k-hop process layer by layer, and list the k-hop vertices or edges that meet the filter criteria.       |
| <b>Filtered-query V2</b>   | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query           | This is a new version of the Filtered Query API. This API supports both Filtered Query and Repeat Query functions. |
| <b>Domain-Specific Language (DSL) Query APIs</b>                               | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=algorithm-query      | Provide flexible and controllable DSLs to help users design and run algorithms at low costs.                       |
| <b>Updating Specified Properties of Vertices and Edges by Importing a File</b> | POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties        | Update specified properties of vertices and edges by importing a file.   |
| <b>Deleting Vertices and Edges by Reading the Files</b>                        | POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file           | Delete vertices and edges by reading the files.  |

## 2.2.2 Database Edition

Database edition service plane APIs include vertex operations, edge operations, metadata operations, index operations, HyG dataset management, HyG algorithm, HyG job management, native algorithms, graph statistics, graph operations, job management, and Cypher operations.

**Table 2-23** Vertex operation APIs

| API   | URL   | Function  |
|---|---|---|
| <a href="#">Querying Vertex Details</a>               | GET/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds={vertex_ids}            | Query details about a specified vertex of a specified set of vertices, including the label information. |
| <a href="#">Querying Vertex Data in Batches</a>       | POST/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query            | Query vertex details in batches.  |
| <a href="#">Adding Vertices in Batches</a>            | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add             | Add vertices in batches.  |
| <a href="#">Deleting Vertices in Batches</a>          | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete          | Delete vertices in batches based on the vertex IDs.   |
| <a href="#">Updating Vertex Properties in Batches</a> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?action_id={actionId} | Update vertex properties in batches.  |

**Table 2-24** Edge operation APIs

| API   | URL   | Function   |
|---|---|--|
| <a href="#">Querying Edge Details</a>         | GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?source={sourceVertex}&target={targetVertex}&index={index} | Query the details about an edge based on its source and target vertices, including the edge's label information. |
| <a href="#">Querying Edge Data in Batches</a> | POST/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query                                     | Query edge details in batches.   |

| API  | URL  | Function   |
|--|--|--|
| <b>Adding Edges in Batches</b>             | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add             | Add edges in batches.  |
| <b>Deleting Edges in Batches</b>           | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete          | Delete edges in batches based on the source vertices, target vertices, and indexes of the edges. |
| <b>Updating Edge Properties in Batches</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId} | Update edge properties in batches.   |

**Table 2-25** Metadata operation APIs

| API                                    | URL   | Function                |
|--|---|-------------------------|
| <b>Adding a Label</b>                  | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels                         | Add a label.            |
| <b>Updating a Label</b>                | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties | Update a label.         |
| <b>Querying Graph Metadata Details</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema                                 | Query metadata details. |
| <b>Querying Labels</b>                 | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}               | Query labels.           |
| <b>Generating Data Assets</b>          | POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets                    | Generate data assets.   |
| <b>Obtaining Data Assets</b>           | GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets                     | Obtain data assets.     |

**Table 2-26** Index operation APIs

| API                                | URL  | Function  |
|------------------------------------|--|---|
| <b>Creating an Index</b>           | POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices                              | Create an index.  |
| <b>Deleting an Index</b>           | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName}                | Delete an index.  |
| <b>Querying Indexes</b>            | GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices                               | Query indexes.  |
| <b>Creating Indexes in Batches</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices/action?action_id=batch-build | Create indexes in batches. By doing so, the number of data scans is reduced and the overall time required is shorten. |

**Table 2-27** HyG graph management APIs

| API   | URL   | Function   |
|---|---|--|
| <b>Creating a HyG Graph</b>                           | POST /ges/v1.0/{project_id}/hyg/{graph_name}              | Create a HyG graph.  |
| <b>Synchronizing HyG Graph Data</b>                   | POST /ges/v1.0/{project_id}/hyg/{graph_name}/sync         | Synchronize graph database updates to the HyG computing engine.  |
| <b>Querying General Information About a HyG Graph</b> | GET /ges/v1.0/{project_id}/hyg/{graph_name}/summary       | Query general information about a HyG graph, such as the number of vertices, number of edges, properties, and partitioning policies. |
| <b>Deleting a HyG Graph</b>                           | DELETE /ges/v1.0/{project_id}/hyg/{graph_name}            | Delete a HyG graph.  |
| <b>Listing HyG Graphs</b>                             | GET /ges/v1.0/{project_id}/hyg                            | List HyG graphs.   |
| <b>Importing a HyG Graph</b>                          | POST /ges/v1.0/{project_id}/hyg/{graph_name}/import-graph | Import HyG graph data.   |

**Table 2-28** HyG algorithm API

| API                           | URL   | Function  |
|-------------------------------|---|---|
| <b>Runing Algorithms</b>      | POST /ges/v1.0/{project_id}/graphs/{graph_name}/hyg/algorithm | Run the HyG algorithm.  |
| <b>Running DSL Algorithms</b> | POST /ges/v1.0/{project_id}/hyg/{graph_name}/dsl              | Provide flexible DSLs to help users design and run algorithms at low costs. |

**Table 2-29** HyG job management APIs

| API                                  | URL  | Function   |
|--------------------------------------|--|--|
| <b>Dumping HyG Algorithm Results</b> | POST /ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}/export-result | Dump the execution results of the algorithm (jobId) to OBS.                    |
| <b>Canceling a HyG Job</b>           | DELETE /ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}             | Cancel a submitted HyG job. (Currently, only jobs in a queue can be canceled.) |

**Table 2-30** Native algorithm API

| API                      | URL  | Function               |
|--------------------------|--|------------------------|
| <b>Runing Algorithms</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm | Run native algorithms. |

**Table 2-31** Graph statistics APIs

| API   | URL  | Function                                 |
|---|--|--|
| <b>Querying General Information About a Graph</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary | Query general information about a graph. |
| <b>Querying the Graph Version</b>                 | GET /ges/v1.0/{project_id}/graphs/{graph_name}/version | Query the graph version.                 |

**Table 2-32** Graph operation APIs

| API                      | URL   | Function           |
|--------------------------|---|--------------------|
| <b>Importing a Graph</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph | Import graph data. |
| <b>Clearing a Graph</b>  | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph  | Clear graph data.  |
| <b>Exporting a Graph</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph | Export graph data. |
| <b>Creating a Graph</b>  | POST /ges/v1.0/{project_id}/graphs  | Create a graph.    |
| <b>Deleting a Graph</b>  | DELETE /ges/v1.0/{project_id}/graphs/{graph_name}                             | Delete a graph.    |
| <b>Listing Graphs</b>    | GET /ges/v1.0/{project_id}/graphs   | List graphs.       |

**Table 2-33** Job management APIs

| API                          | URL  | Function  |
|------------------------------|--|---|
| <b>Querying Job Status</b>   | GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset= <i>offset</i> &limit= <i>limit</i> | Query the job status.   |
| <b>Querying the Job List</b> | GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}                       | After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned. |

**Table 2-34** Cypher operation APIs

| API                          | URL   | Function   |
|------------------------------|---|--|
| <b>Cypher Operation APIs</b> | POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query | Run Cypher statements to query data in GES and obtain results. |

# 3 Calling APIs

---

## 3.1 Making an API Request

### 3.1.1 Making a Management Plane API Request

This section describes the structure of a REST API request on the management plane of GES.

#### Request URI

A request URI is in the following format:

**{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}**

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme:** Protocol used to transmit requests. All APIs use **HTTPS**.
- **Endpoint:** Endpoints vary depending on services and regions.
- **resource-path:** Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
- **query-string:** Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

#### Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET:** requests the server to return specified resources.
- **PUT:** requests the server to update specified resources.
- **POST:** requests the server to add resources or perform special operations.

- **DELETE:** requests the server to delete specified resources, for example, an object.
- **HEAD:** same as GET except that the server must return only the response header.
- **PATCH:** requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

## Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

**Table 3-1** Common request headers

| Parameter     | Mandatory   | Description   | Example                          |
|---------------|---|---|----------------------------------|
| Content-Type  | Yes   | Specifies the request body type or format. This field is mandatory and its default value is <b>application/json</b> . Other values of this field will be provided for specific APIs if any. | application/json                 |
| X-Auth-Token  | This field is mandatory only for authentication using tokens. | Specifies a user token only for token-based API authentication.   | -                                |
| X-Project-ID  | No  | Specifies a subproject ID. This parameter is mandatory only in multi-project scenarios.   | e9993fc787d94b6c886cbaa340f9c0f4 |
| Authorization | This field is mandatory for authentication using AK/SK.       | Specifies the signature authentication information. The value is obtained from the request signing result.  | -                                |



| Parameter      | Mandatory   | Description  | Example                                  |
|----------------|---|--|--|
| X-Sdk-Date     | This field is mandatory for authentication using AK/SK.                                     | Specifies the time when a request is sent. The time is in <i>YYYYMMDDTHHMMSSZ</i> format. The value is the current Greenwich Mean Time (GMT) time of the system.   | 20150907T101459Z                         |
| Host           | This field is mandatory for authentication using AK/SK.                                     | Specifies the information about the requested server. The value can be obtained from the URL of the service API. The value is in the <i>hostname[:port]</i> format. Default port used for <b>https</b> requests is port <b>443</b> . | code.test.com<br>or<br>code.test.com:443 |
| Content-Length | This field is mandatory for POST and PUT requests, but must be left blank for GET requests. | Specifies the length of the request body. The unit is byte.  | 3495                                     |
| X-Language     | No  | Request language   | en-us                                    |

 **NOTE**

In addition to token-based authentication, authentication using access key ID/secret access key (AK/SK) is also supported. During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

## Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

For the API of obtaining a user token, obtain the request parameters and parameter description in the API request. The following provides an example

request with a body included. Replace *username*, *domiannname*, *\*\*\*\*\** (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values.

If all data required for the API request is available, you can send the request to call the API through code. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

## 3.1.2 Making a Service Plane API Request

This section describes the structure of a REST API on the service plane of GES.

### Request URI

A request URI of a service plane API of GES is in the following format:

**{URI-scheme} :// {SERVER\_URL} / {resource-path} ? {query-string}**

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme:** Protocol used to transmit requests. All APIs use **HTTPS**.
- **SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- **resource-path:** Access path of an API for performing a specified operation. Obtain the value from the URI module of the API, for example, **ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?action\_id=query**.
- **query-string:** Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

### Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET:** requests the server to return specified resources.
- **PUT:** requests the server to update specified resources.
- **POST:** requests the server to add resources or perform special operations.
- **DELETE:** requests the server to delete specified resources, for example, an object.
- **HEAD:** same as GET except that the server must return only the response header.
- **PATCH:** requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

### Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

**Table 3-2** Common request headers

| Parameter    | Mandatory | Description   | Example          |
|--------------|-----------|---|------------------|
| Content-Type | Yes       | Specifies the request body type or format. This field is mandatory and its default value is <b>application/json</b> . Other values of this field will be provided for specific APIs if any. | application/json |
| X-Auth-Token | Yes       | Specifies a user token only for token-based API authentication.   | -                |
| X-Language   | Yes       | Request language  | en-us            |

## Request Message Body

The body of a request is often sent in a structured format as specified in **Content-Type**. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the **GET** and **DELETE** methods.

For the API of obtaining a user token, obtain the request parameters and parameter description in the API request. The following provides an example request with a body included. Replace *username*, *domiannname*, *\*\*\*\*\** (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values.

If all data required for the API request is available, you can send the request to call the API through code. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. which can be used to authenticate the calling of other APIs.

## 3.2 Authentication

### 3.2.1 Authentication of Management Plane APIs

Requests for calling a management plane API of GES can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure.

## Token-based Authentication

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

When calling the API to obtain a user token, you must set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

### NOTE

- The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
- To obtain the token, the GES **scope** must be **project** (cannot be **domain**).

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request:

## 3.2.2 Authentication of Service Plane APIs

Calling a service plane API of GES can only be authenticated using tokens.

### Token-based Authentication

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

When calling the API to obtain a user token, you must set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
```

```

    "password": {
      "user": {
        "name": "username",
        "password": "*****",
        "domain": {
          "name": "domainname"
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}

```

 **NOTE**

- The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
- To obtain the token, the GES **scope** must be **project** (cannot be **domain**).

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request:

## 3.3 Response

### Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Status Codes](#).

### Response Header

Similar to a request, a response also has a header, for example, **Content-Type**. [Table 3-3](#) list the response header parameters.

**Table 3-3** Response header parameters

| Parameter      | Description  |
|----------------|--|
| Content-Length | Specifies the length (in bytes) of the response body.                  |
| Date           | Specifies the time when a response is returned.                        |
| Content-type   | Specifies the MIME type of the response body.                          |
| TraceID        | Specifies the ID returned by the request, facilitating fault locating. |

## Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

# 4 Management Plane APIs (V2)

---

## 4.1 System Management

### 4.1.1 Querying Quotas

#### Function

This API is used to query tenant quotas.

#### URI

GET /v2/{project\_id}/graphs/quotas

**Table 4-1** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 4-2** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token.<br>It is used to obtain the permission to call APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-3** Response body parameters

| Parameter | Type                          | Description  |
|-----------|-------------------------------|--|
| quotas    | <a href="#">quotas</a> Object | Resource type list. This field is left blank when the request fails. |

**Table 4-4** quotas

| Parameter | Type                                       | Description             |
|-----------|--|-------------------------|
| resources | Array of <a href="#">resources</a> objects | GES resource quota list |

**Table 4-5** resources

| Parameter | Type    | Description  |
|-----------|---------|--|
| type      | String  | Quota type<br>Available values are as follows: <ul style="list-style-type: none"> <li>"graph"</li> <li>"backup"</li> <li>"metadata"</li> </ul> |
| available | Integer | Number of available graphs   |



| Parameter   | Type    | Description   |
|-------------|---------|---|
| edge_volume | Integer | Number of available edges. The parameter value is valid only when <b>type</b> is <b>graph</b> . |

**Status code: 400**

**Table 4-6** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| error_msg  | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>         |

## Example Request

Query tenant quotas.

GET https://Endpoint/v2/{project\_id}/graphs/quotas

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "quotas": {
    "resources": [ {
      "type": "graph",
      "available": 10,
      "edge_volume": 7300
    }, {
      "type": "backup",
      "available": 100
    }, {
      "type": "metadata",
      "available": 86
    }
  ]
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

# 4.2 Graph Management

## 4.2.1 Querying the Graph List

### Function

This API is used to query all graphs of the current tenant.

### URI

GET /v2/{project\_id}/graphs

**Table 4-7** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 4-8** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |

## Request Parameters

**Table 4-9** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token<br>It is used to obtain the permission to call APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-10** Response body parameters

| Parameter   | Type                                    | Description   |
|-------------|---|---|
| graph_count | Integer                                 | Total number of graphs. Graph object. If the request fails, this parameter is left empty. |
| graphs      | Array of <a href="#">graphs</a> objects | Graph list. Graph object. If the request fails, this parameter is left empty.             |

**Table 4-11** graphs

| Parameter | Type   | Description |
|-----------|--------|-------------|
| id        | String | Graph ID    |

| Parameter               | Type             | Description  |
|-------------------------|------------------|--|
| name                    | String           | Graph name   |
| created_by              | String           | IAM username   |
| is_multi_az             | String           | Whether cross-AZ HA will be enabled  |
| region_code             | String           | Region code  |
| az_code                 | String           | AZ code  |
| edgeset_format          | String           | Format of the edge data file   |
| edgeset_default_label   | String           | Default label of the edge data file  |
| vertexset_format        | String           | Format of the vertex data file   |
| vertexset_default_label | String           | Default label of the vertex data file  |
| data_store_version      | String           | Graph version  |
| sys_tags                | Array of strings | Enterprise project information. If this parameter is not specified, this function is disabled (default). |

| Parameter             | Type   | Description  |
|-----------------------|--------|--|
| status                | String | <p>Status code of a graph</p> <ul style="list-style-type: none"> <li>● <b>100</b>: Indicates that a graph is being prepared.</li> <li>● <b>200</b>: indicates that a graph is running.</li> <li>● <b>201</b>: indicates that a graph is upgrading.</li> <li>● <b>202</b>: indicates that a graph is being imported.</li> <li>● <b>203</b>: indicates that a graph is being rolled back.</li> <li>● <b>204</b>: indicates that a graph is being exported.</li> <li>● <b>205</b>: indicates that a graph is being cleared.</li> <li>● <b>206</b>: indicates that the system is preparing for resize.</li> <li>● <b>207</b>: indicates that the resize is in progress.</li> <li>● <b>208</b>: Indicates that the resize is being rolled back.</li> <li>● <b>210</b>: Preparing for expansion</li> <li>● <b>211</b>: Expanding</li> <li>● <b>300</b>: indicates that a graph is faulty.</li> <li>● <b>303</b>: indicates that a graph fails to be created.</li> <li>● <b>400</b>: indicates that a graph is deleted.</li> <li>● <b>800</b>: indicates that a graph is frozen.</li> <li>● <b>900</b>: indicates that a graph is stopped.</li> <li>● <b>901</b>: indicates that a graph is being stopped.</li> <li>● <b>920</b>: indicates that a graph is being started.</li> </ul> |
| action_progress       | String | <p>Progress of graph creation</p> <p><b>NOTE</b><br/>This field is returned only when <b>status</b> is <b>100</b>.</p>   |
| graph_size_type_index | String | <p>Graph size type index</p> <ul style="list-style-type: none"> <li>● <b>0</b>: indicates 10 thousand edges.</li> <li>● <b>1</b>: indicates 1 million edges.</li> <li>● <b>2</b>: indicates 10 million edges.</li> <li>● <b>3</b>: indicates 100 million edges.</li> <li>● <b>4</b>: indicates 1 billion edges.</li> <li>● <b>5</b>: indicates 10 billion edges.</li> <li>● <b>6</b>: indicates the database edition.</li> <li>● <b>401</b>: indicates 1 billion enhanced edges.</li> </ul>  |
| vpc_id                | String | VPC ID   |

| Parameter              | Type                         | Description  |
|------------------------|------------------------------|--|
| subnet_id              | String                       | Subnet ID in the VPC   |
| security_group_id      | String                       | Security group ID  |
| replication            | Integer                      | Number of replicas. The default value is <b>1</b> .  |
| created                | String                       | Time when a graph is created   |
| updated                | String                       | Time when a graph is updated   |
| private_ip             | String                       | Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.   |
| public_ip              | String                       | Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.   |
| arch                   | String                       | Graph instance's CPU architecture type. The value can be <b>x86_64</b> or <b>aarch64</b> .   |
| master_key_id          | String                       | User master key ID   |
| master_key_name        | String                       | User master key name   |
| enable_rbac            | Boolean                      | Whether to enable granular permission control  |
| enable_full_text_index | Boolean                      | Whether to enable full-text indexes  |
| enable_hyg             | Boolean                      | Whether to enable HyG. This parameter is available only for database edition graphs.   |
| traffic_ip_list        | Array of strings             | Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.  |
| crypt_algorithm        | String                       | Graph instance cryptography algorithm. Available values are as follows: <ul style="list-style-type: none"> <li><b>generalCipher</b>: Chinese cryptographic algorithm</li> <li><b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul> |
| enable_https           | Boolean                      | Whether to enable the security mode. This mode may damage GES performance greatly.   |
| tags                   | Array of <b>tags</b> objects | Tag list. Each tag is in <key,value> format.   |

| Parameter      | Type                         | Description   |
|----------------|------------------------------|---|
| product_type   | String                       | Graph type. The value can be <b>InMemory</b> (default value) or <b>Persistence</b> . If <b>graph_size_type_index</b> is <b>6</b> , the value must be <b>Persistence</b> . <ul style="list-style-type: none"> <li>• <b>InMemory</b>: memory edition</li> <li>• <b>Persistence</b>: database version</li> </ul> |
| vertex_id_type | <b>vertex_id_type</b> object | ID type of vertices. This parameter is mandatory only for database edition graphs. <p><b>NOTE</b><br/>The vertex ID type cannot be changed once set. Exercise caution when setting this parameter.</p>  |

**Table 4-12** tags

| Parameter | Type   | Description |
|-----------|--------|-------------|
| key       | String | Tag key     |
| value     | String | Tag value   |

**Table 4-13** vertex\_id\_type

| Parameter | Type    | Description   |
|-----------|---------|---|
| id_type   | String  | Vertex ID type. The value can be <b>fixedLengthString</b> or <b>hash</b> . <ul style="list-style-type: none"> <li>• <b>fixedLengthString</b>: Vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs.</li> <li>• <b>hash</b>: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately <math>10^{(-43)}</math>, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to <b>Hash</b>.</li> </ul> |
| id_length | Integer | This parameter is mandatory if <b>id_type</b> is <b>fixedLengthString</b> . The value ranges from 1 to 128.   |

**Status code: 400**

**Table 4-14** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |

### Example Request

View the first 10 graphs of the current tenant.

```
GET https://Endpoint/v2/{project_id}/graphs?offset=0&limit=10
```

### Example Response

**Status code: 200**

Example response for a successful request

```
{
  "graph_count" : 2,
  "graphs" : [ {
    "id" : "146227d2-bfac-499a-97df-df475349e43d",
    "name" : "TenThousand_Charging",
    "created_by" : "ei_ges_j00356469_01",
    "is_multi_az" : "false",
    "region_code" : "XXX",
    "az_code" : "XXX",
    "edgeset_format" : "",
    "edgeset_default_label" : "",
    "vertexset_format" : "",
    "vertexset_default_label" : "",
    "data_store_version" : "2.3.2",
    "sys_tags" : [ "0" ],
    "status" : "200",
    "graph_size_type_index" : "0",
    "vpc_id" : "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id" : "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id" : "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication" : 1,
    "created" : "2022-04-26T02:19:54",
    "private_ip" : "192.168.0.228",
    "arch" : "x86_64",
    "encrypted" : false,
    "master_key_id" : "",
    "master_key_name" : "",
    "enable_rbac" : false,
    "enable_full_text_index" : false,
    "enable_hyg" : false,
    "traffic_ip_list" : [ "192.168.0.228" ],
    "crypt_algorithm" : "generalCipher",
  } ]
}
```



```

    "enable_https" : false
  }, {
    "id" : "1172f16c-63c7-4746-89b0-78972eddf706",
    "name" : "GES_UI_2_0_1",
    "created_by" : "ei_ges_j00356469_01",
    "is_multi_az" : "false",
    "region_code" : "XXX",
    "az_code" : "XXX",
    "edgeset_format" : "",
    "edgeset_default_label" : "",
    "vertexset_format" : "",
    "vertexset_default_label" : "",
    "data_store_version" : "2.0.1",
    "sys_tags" : [ "0" ],
    "status" : "900",
    "graph_size_type_index" : "0",
    "vpc_id" : "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id" : "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id" : "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication" : 1,
    "created" : "2022-04-26T02:10:52",
    "private_ip" : "192.168.0.123",
    "encrypted" : false,
    "enable_rbac" : false,
    "enable_full_text_index" : false,
    "enable_hyg" : false,
    "traffic_ip_list" : [ "192.168.0.123" ],
    "crypt_algorithm" : "generalCipher",
    "enable_https" : false
  }
}

```

**Status code: 400**

Example response for a failed request

```

{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 4.2.2 Querying Graph Details

### Function

This API is used to query details about a graph based on the graph ID.

### URI

GET /v2/{project\_id}/graphs/{graph\_id}

**Table 4-15** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

### Request Parameters

**Table 4-16** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token.<br>It is used to obtain the permission to call APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

### Response Parameters

Status code: 200

**Table 4-17** Response body parameters

| Parameter | Type                         | Description   |
|-----------|------------------------------|---|
| graph     | <a href="#">graph</a> object | Graph object. If the request fails, this parameter is left empty. |

**Table 4-18** graph

| Parameter               | Type             | Description  |
|-------------------------|------------------|--|
| id                      | String           | Graph ID   |
| name                    | String           | Graph name   |
| created_by              | String           | IAM username   |
| is_multi_az             | String           | Whether to enable cross-AZ HA  |
| region_code             | String           | Region code  |
| az_code                 | String           | AZ code  |
| edgeset_format          | String           | Format of the edge data file   |
| edgeset_default_label   | String           | Default label of the edge data file  |
| vertexset_format        | String           | Format of the vertex data file   |
| vertexset_default_label | String           | Default label of the vertex data file  |
| data_store_version      | String           | Graph version  |
| sys_tags                | Array of strings | Enterprise project information. If this parameter is not specified, this function is disabled (default). |

| Parameter             | Type   | Description   |
|-----------------------|--------|---|
| status                | String | <p>Status code of a graph:</p> <ul style="list-style-type: none"> <li>● <b>100</b>: Indicates that a graph is being prepared.</li> <li>● <b>200</b>: indicates that a graph is running.</li> <li>● <b>201</b>: indicates that a graph is upgrading.</li> <li>● <b>202</b>: indicates that a graph is being imported.</li> <li>● <b>203</b>: indicates that a graph is being rolled back.</li> <li>● <b>204</b>: indicates that a graph is being exported.</li> <li>● <b>205</b>: indicates that a graph is being cleared.</li> <li>● <b>206</b>: indicates that the system is preparing for resize.</li> <li>● <b>207</b>: indicates that the resize is in progress.</li> <li>● <b>208</b>: Indicates that the resize is being rolled back.</li> <li>● <b>210</b>: Preparing for expansion</li> <li>● <b>211</b>: Expanding</li> <li>● <b>300</b>: indicates that a graph is faulty.</li> <li>● <b>303</b>: indicates that a graph fails to be created.</li> <li>● <b>400</b>: indicates that a graph is deleted.</li> <li>● <b>800</b>: indicates that a graph is frozen.</li> <li>● <b>900</b>: indicates that a graph is stopped.</li> <li>● <b>901</b>: indicates that a graph is being stopped.</li> <li>● <b>920</b>: indicates that a graph is being started.</li> </ul> |
| action_progress       | String | <p>Progress of graph creation</p> <p><b>NOTE</b><br/>This field is returned only when <b>status</b> is <b>100</b>.</p>  |
| graph_size_type_index | String | <p>Graph size type index:</p> <ul style="list-style-type: none"> <li>● <b>0</b>: indicates 10 thousand edges.</li> <li>● <b>1</b>: indicates 1 million edges.</li> <li>● <b>2</b>: indicates 10 million edges.</li> <li>● <b>3</b>: indicates 100 million edges.</li> <li>● <b>4</b>: indicates 1 billion edges.</li> <li>● <b>5</b>: indicates 10 billion edges.</li> <li>● <b>6</b>: indicates the database edition.</li> <li>● <b>401</b>: indicates 1 billion enhanced edges.</li> </ul>  |
| vpc_id                | String | VPC ID  |

| Parameter              | Type                         | Description  |
|------------------------|------------------------------|--|
| subnet_id              | String                       | Subnet ID in the VPC   |
| security_group_id      | String                       | Security group ID  |
| replication            | Integer                      | Number of replicas. The default value is <b>1</b> .  |
| created                | String                       | Time when a graph is created   |
| updated                | String                       | Time when a graph is updated   |
| private_ip             | String                       | Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.   |
| public_ip              | String                       | Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.   |
| arch                   | String                       | Graph instance's CPU architecture type. The value can be <b>x86_64</b> or <b>aarch64</b> .   |
| master_key_id          | String                       | User master key ID   |
| master_key_name        | String                       | User master key name   |
| enable_rbac            | Boolean                      | Whether to enable granular permission control  |
| enable_full_text_index | Boolean                      | Whether to enable full-text indexes  |
| enable_hyg             | Boolean                      | Whether to enable HyG. This parameter is available only for database edition graphs.   |
| traffic_ip_list        | Array of strings             | Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.  |
| crypt_algorithm        | String                       | Graph instance cryptography algorithm. Available values are as follows: <ul style="list-style-type: none"> <li><b>generalCipher</b>: Chinese cryptographic algorithm</li> <li><b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul> |
| enable_https           | Boolean                      | Whether to enable the security mode. This mode may damage GES performance greatly.   |
| tags                   | Array of <b>tags</b> objects | Tag list. Each tag is in <key,value> format.   |

| Parameter                    | Type                                  | Description   |
|------------------------------|---------------------------------------|---|
| product_type                 | String                                | Graph type. The value can be <b>InMemory</b> (default value) or <b>Persistence</b> . If <b>graph_size_type_index</b> is <b>6</b> , the value must be <b>Persistence</b> . <ul style="list-style-type: none"> <li>• <b>InMemory</b>: memory edition</li> <li>• <b>Persistence</b>: database version</li> </ul> |
| vertex_id_type               | <a href="#">vertex_id_type</a> object | ID type of vertices. This parameter is mandatory only for database edition graphs.<br><b>NOTE</b><br>The vertex ID type cannot be changed once set. Exercise caution when setting this parameter.   |
| origin_graph_size_type_index | String                                | Initial size of a graph.  |
| expand_time                  | String                                | Time when a graph is expanded.  |
| resize_time                  | String                                | Time when a graph is resized.   |
| enable_multi_label           | Boolean                               | Whether multi-labeling is enabled.  |
| capacity_ratio               | Integer                               | Capacity rate of a graph. This parameter is returned only for 10-billion-edge graphs of the database edition.   |

**Table 4-19** tags

| Parameter | Type   | Description |
|-----------|--------|-------------|
| key       | String | Tag key.    |
| value     | String | Tag value   |

**Table 4-20** vertex\_id\_type

| Parameter | Type    | Description   |
|-----------|---------|---|
| id_type   | String  | <p>Vertex ID type. The value can be <b>fixedLengthString</b> or <b>hash</b>.</p> <ul style="list-style-type: none"> <li>• <b>fixedLengthString</b>: Vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs.</li> <li>• <b>hash</b>: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately <math>10^{(-43)}</math>, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to <b>Hash</b>.</li> </ul> |
| id_length | Integer | <p>This parameter is mandatory if <b>id_type</b> is <b>fixedLengthString</b>. The value ranges from 1 to 128.</p>   |

**Status code: 400**

**Table 4-21** Response body parameters

| Parameter  | Type   | Description   |
|------------|--------|---|
| error_code | String | <p>System prompt code.</p> <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | <p>System prompt code.</p> <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

**Example Request**

Query details about a graph based on the graph ID.

```
GET https://Endpoint/v2/{project_id}/graphs/{graph_id}
```

## Example Response

### Status code: 200

Example response for a successful request

```
{
  "graph": {
    "id": "1172f16c-63c7-4746-89b0-78972eddf706",
    "name": "GES_UI_2_0_1",
    "created_by": "ei_ges_j00356469_01",
    "is_multi_az": "false",
    "region_code": "XXX",
    "az_code": "XXX",
    "edgeset_format": "",
    "edgeset_default_label": "",
    "vertexset_format": "",
    "vertexset_default_label": "",
    "data_store_version": "2.0.1",
    "sys_tags": [ "0" ],
    "status": "900",
    "graph_size_type_index": "0",
    "vpc_id": "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id": "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id": "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication": 0,
    "created": "2022-04-26T02:10:52",
    "updated": "2022-04-26T02:10:52",
    "private_ip": "192.168.0.123",
    "encrypted": false,
    "enable_rbac": false,
    "enable_full_text_index": false,
    "enable_hyg": false,
    "traffic_ip_list": [ "192.168.0.123" ],
    "crypt_algorithm": "generalCipher",
    "enable_https": false,
    "enable_multi_label": false,
    "origin_graph_size_type_index": "1",
    "expand_time": "2023-08-03T02:10:52",
    "resize_time": "2023-08-02T02:10:52"
  }
}
```

### Status code: 400

Example response for a failed request

```
{
  "error_code": "GES.7006",
  "error_msg": "An internal error occurs in the underlying service of the graph engine."
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |



| Return Value            | Description          |
|-------------------------|----------------------|
| 503 Service Unavailable | Service unavailable. |

## Error Codes

See [Error Codes](#).

## 4.2.3 Creating a Graph

### Function

This API is used to create a graph.

### URI

POST /v2/{project\_id}/graphs

**Table 4-22** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

### Request Parameters

**Table 4-23** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-24** Request body parameter

| Parameter | Mandatory | Type                | Description |
|-----------|-----------|---------------------|-------------|
| graph     | Yes       | <b>graph</b> object | Graph type  |

**Table 4-25** graph

| Parameter             | Mandatory | Type   | Description   |
|-----------------------|-----------|--------|---|
| name                  | Yes       | String | A graph name must start with a letter and contains 4 to 50 characters consisting of letters, digits, hyphens (-), and underscores (_). It cannot contain special characters.  |
| graph_size_type_index | Yes       | String | Graph size type index: <ul style="list-style-type: none"> <li>• <b>0</b>: indicates 10 thousand edges.</li> <li>• <b>1</b>: indicates 1 million edges.</li> <li>• <b>2</b>: indicates 10 million edges.</li> <li>• <b>3</b>: indicates 100 million edges.</li> <li>• <b>4</b>: indicates 1 billion edges.</li> <li>• <b>5</b>: indicates 10 billion edges.</li> <li>• <b>6</b>: indicates the database edition.</li> <li>• <b>401</b>: indicates 1 billion enhanced edges.</li> </ul> |
| arch                  | No        | String | Graph instance's CPU architecture type. The value can be <b>x86_64</b> or <b>aarch64</b> . The default value is <b>x86_64</b> . <ul style="list-style-type: none"> <li>• <b>x86_64</b>: x86 64-bit architecture</li> <li>• <b>aarch64</b>: Arm 64-bit architecture</li> </ul>   |
| vpc_id                | Yes       | String | VPC ID  |
| subnet_id             | Yes       | String | Subnet ID in the VPC  |
| security_group_id     | Yes       | String | Security group ID   |

| Parameter           | Mandatory | Type                                       | Description   |
|---------------------|-----------|--|---|
| public_ip           | No        | <a href="#">public_ip</a> object           | Public IP address. If the parameter is not specified, public connection is not used by default.   |
| enable_multi_az     | No        | Boolean                                    | Whether the created graph supports the cross-AZ mode. The default value is <b>false</b> . If the value is <b>true</b> , the system will create the ECSs in the graph in two AZs.<br><br>If this parameter is not specified when you create a graph, all ECSs in the graph are created in one AZ.  |
| lts_operation_trace | No        | <a href="#">lts_operation_trace</a> object | Whether to enable audit logs. This function is disabled by default.   |
| sys_tags            | No        | Array of <a href="#">sys_tags</a> objects  | Enterprise project information. If this parameter is not specified, this function is disabled (default).  |
| tags                | No        | Array of <a href="#">tags</a> objects      | TMS tags for expenses. This function is disabled by default.  |
| enable_rbac         | No        | Boolean                                    | Whether to enable granular permission control for the created graph. The default value is <b>false</b> , indicating that granular permission control is disabled. If this parameter is set to <b>true</b> , no user has the permission to access the graph. To access the graph, you need to call the granular permission control API of the service plane to set the required permissions. |

| Parameter              | Mandatory | Type    | Description  |
|------------------------|-----------|---------|--|
| enable_full_text_index | No        | Boolean | <p>Whether to enable full-text index control for the created graph. The default value is <b>false</b>, indicating that full-text index control is disabled. If this parameter is set to <b>true</b>, full-text indexes are available for 1-billion-edge-pro graphs, and a Cloud Search Service (CSS) cluster will be created when you create a graph.</p> <p><b>NOTE</b><br/>If you enable full-text indexes: If the CSS has been deployed, the system automatically creates a CSS cluster during the creation of the graph instance, which will take a long time. If the CSS is not deployed, the graph creation will fail.</p> |
| enable_hyg             | No        | Boolean | Whether to enable HyG for the graph. This parameter is available for database edition graphs only.   |
| crypt_algorithm        | Yes       | String  | <p>Graph instance cryptography algorithm. Available values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>generalCipher</b>: Chinese cryptographic algorithm</li> <li>• <b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul>  |
| enable_https           | Yes       | Boolean | Whether to enable the security mode. This mode may damage GES performance greatly.   |
| product_type           | No        | String  | <p>Graph type. The value can be <b>InMemory</b> (default value) or <b>Persistence</b>. If <b>graph_size_type_index</b> is 6, the value must be <b>Persistence</b>.</p> <ul style="list-style-type: none"> <li>• <b>InMemory</b>: memory edition</li> <li>• <b>Persistence</b>: database version</li> </ul>   |

| Parameter          | Mandatory | Type                                  | Description   |
|--------------------|-----------|---------------------------------------|---|
| vertex_id_type     | No        | <a href="#">vertex_id_type</a> object | ID type of vertices. This parameter is mandatory only for database edition graphs.<br><br><b>NOTE</b><br>The vertex ID type cannot be changed once set. Exercise caution when setting this parameter. |
| enable_multi_label | No        | Boolean                               | Whether multi-labeling is enabled.  |
| capacity_ratio     | No        | Integer                               | Capacity rate of a graph. This parameter is returned only for 10-billion-edge graphs of the database edition.   |

**Table 4-26** public\_ip

| Parameter        | Mandatory | Type   | Description   |
|------------------|-----------|--------|---|
| public_bind_type | No        | String | Binding type of an EIP. The value can be either of the following: <ul style="list-style-type: none"> <li>• <b>auto_assign</b></li> <li>• <b>bind_existing</b></li> </ul>  |
| eip_id           | No        | String | ID of an EIP. When <b>publicBindType</b> is set to <b>bind_existing</b> , its value is the ID of an EIP that has been created but has not been bound. When <b>publicBindType</b> is set to <b>auto_assign</b> , its value is set to null. |

**Table 4-27** lts\_operation\_trace

| Parameter            | Mandatory | Type    | Description  |
|----------------------|-----------|---------|--|
| enable_audit         | No        | Boolean | Whether to enable graph audit. The default value is <b>false</b> . |
| audit_log_group_name | No        | String  | LTS log group name   |

**Table 4-28** sys\_tags

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| key       | No        | String | Key of the enterprise project. The value is <b>_sys_enterprise_project_id</b> . |
| value     | No        | String | Enterprise project ID. You can obtain it from the enterprise project.           |

**Table 4-29** tags

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| key       | No        | String | Tag key     |
| value     | No        | String | Tag value   |

**Table 4-30** vertex\_id\_type

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| id_type   | Yes       | String | <p>Vertex ID type. The value can be <b>fixedLengthString</b> or <b>hash</b>.</p> <ul style="list-style-type: none"> <li>• <b>fixedLengthString</b>: Vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs.</li> <li>• <b>hash</b>: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately <math>10^{(-43)}</math>, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to <b>Hash</b>.</li> </ul> |

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| id_length | No        | Integer | This parameter is mandatory if <b>id_type</b> is <b>fixedLengthString</b> . The value ranges from 1 to 128. |

## Response Parameters

**Status code: 200**

**Table 4-31** Response body parameters

| Parameter | Type   | Description |
|-----------|--------|-------------|
| id        | String | Graph ID    |
| name      | String | Graph name  |

**Status code: 400**

**Table 4-32** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |

## Example Request

Create a graph. The graph name is **demo**, the graph size is 10-thousand-edge, the graph instance CPU architecture type is x86 64-bit, the VPC ID is **f07ed478-c31e-47ed-92d4-8fae6fd16981**, the subnet ID is **70bc9d11-b172-479b-8bce-1ad1ade02224**, and the security group ID is **65de00b0-45f5-432f-8346-d8bb5f30940d**.

```
{
  "graph": {
    "name": "demo",
```

```

"graph_size_type_index": 0,
"capacity_ratio": "",
"arch": "x86_64",
"vpc_id": "f07ed478-c31e-47ed-92d4-8fae6fd16981",
"subnet_id": "70bc9d11-b172-479b-8bce-1ad1ade02224",
"security_group_id": "65de00b0-45f5-432f-8346-d8bb5f30940d",
"public_ip": {
  "public_bind_type": "bind_existing",
  "eip_id": "b6e3bc7d-1ced-4690-9219-ef41d25e171e"
},
"enable_multi_az": false,

"lts_operation_trace": {
  "enable_audit": true,
  "audit_log_group_name": "test"
},
"sys_tags": [
  {
    "key": "_sys_enterprise_project_id",
    "value": "0"
  }
],
"enable_hyg": false,
"crypt_algorithm": "generalCipher",
"enable_https": false
}

```

## Example Response

### Status code: 200

Example response (successful request)

```

{
  "id": "f1529b88-c958-493e-8452-fccfe932cde1",
  "name": "demo"
}

```

### Status code: 400

Example response (failed request)

```

{
  "error_code": "GES.7016",
  "error_msg": "The parameter [subnetId] is not exist."
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |



## Error Codes

See [Error Codes](#).

## 4.2.4 Closing a Graph

### Function

This API is used to close a graph. After the graph is created, you can disable it if it is not used temporarily.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/stop

**Table 4-33** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

### Request Parameters

**Table 4-34** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

### Response Parameters

Status code: 200

**Table 4-35** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---|
| job_id    | String | ID of the graph stopping job. Graph object. If the request fails, this parameter is left empty. |

**Table 4-36** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |

## Example Request

Close a graph.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/stop
{ }
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_code" : "GES.7001",
  "error_msg" : "The graph is not running."
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.5 Starting a Graph

### Function

This API is used to start a graph. You can disable a graph if it is not used temporarily.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/start

**Table 4-37** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-38** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token.<br>It is used to obtain the permission to call APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-39** Request body parameters

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| graph_backup_id | No        | String | Backup ID associated during graph startup. If this parameter is configured, the graph starts from the backup. If this parameter is left blank, the graph starts from the status when it was closed last time. |

## Response Parameters

**Status code: 200**

**Table 4-40** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---|
| job_id    | String | ID of the graph startup job. This parameter is left blank when the request fails. |

**Status code: 400**

**Table 4-41** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Start a graph whose associated backup ID is **08a898ae-3ff8-40e8-a7ed-03afe05aedbb**.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/start
{
  "graph_backup_id" : "08a898ae-3ff8-40e8-a7ed-03afe05aedbb"
}
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

### Status Code

| Return Value     | Description              |
|------------------|--------------------------|
| 400 Bad Request  | Request error            |
| 401 Unauthorized | Authorization failed     |
| 403 Forbidden    | No operation permissions |

| Return Value              | Description           |
|---------------------------|-----------------------|
| 404 Not Found             | No resources found    |
| 500 Internal Server Error | Internal server error |
| 503 Service Unavailable   | Service unavailable   |

## Error Code

See [Error Code](#).

## 4.2.6 Deleting a Graph

### Function

This API is used to delete a graph.

### URI

DELETE /v2/{project\_id}/graphs/{graph\_id}

**Table 4-42** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 4-43** Query parameters

| Parameter   | Mandatory | Type    | Description   |
|-------------|-----------|---------|---|
| keep_backup | No        | Boolean | Whether to retain the backups of a graph after it is deleted. By default, one automatic backup and two manual backups are retained. If this parameter is left empty, no backups are retained. |

## Request Parameters

**Table 4-44** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

**Table 4-45** Response body parameters

| Parameter | Type   | Description  |
|-----------|--------|--|
| job_id    | String | ID of the graph deletion job. This parameter is left blank when the request fails. |

**Status code: 400**

**Table 4-46** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Delete a graph.

```
DELETE https://Endpoint/v2/{project_id}/graphs/{graph_id}
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_code" : "GES.7000",
  "error_msg" : "The graph does not exist or has been deleted."
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.7 Incrementally Importing Data to a Graph

### Function

This API is used to import data to graphs incrementally.

#### NOTE

To prevent failures in restoring the imported graph data during system restarting, do not delete the data stored on OBS when the graph is in use.



## URI

POST /v2/{project\_id}/graphs/{graph\_id}/import-graph

**Table 4-47** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-48** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-49** Request body parameters

| Parameter        | Mandatory | Type   | Description   |
|------------------|-----------|--------|---|
| edgeset_path     | No        | String | Edge file directory or name   |
| edgeset_format   | No        | String | Format of the edge data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.   |
| vertexset_path   | No        | String | Vertex file directory or name   |
| vertexset_format | No        | String | Format of the vertex data set. Currently, only the CSV format is supported.<br>The CSV format is used by default. |

| Parameter     | Mandatory | Type                                 | Description  |
|---------------|-----------|--------------------------------------|--|
| schema_path   | No        | String                               | Path for storing the metadata file of the new data.  |
| log_dir       | No        | String                               | Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.   |
| parallel_edge | No        | <a href="#">parallel edge</a> object | How to process repetitive edges.   |
| delimiter     | No        | String                               | Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).   |
| trim_quote    | No        | String                               | Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.  |
| offline       | No        | Boolean                              | <p>Whether offline import is selected. The value is <b>true</b> or <b>false</b>, and the default value is <b>false</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import.</li> <li>• <b>false</b>: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.</li> </ul> |

**Table 4-50** parallel\_edge

| Parameter    | Mandatory | Type    | Description  |
|--------------|-----------|---------|--|
| action       | No        | String  | <p>Processing mode of repetitive edges. The value can be <b>allow</b>, <b>ignore</b>, or <b>override</b>. The default value is <b>allow</b>.</p> <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul>   |
| ignore_label | No        | Boolean | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: Indicates that the repetitive edge definition contains the label. That is, the &lt;source vertex, target vertex, label&gt; indicates an edge.</li> </ul> |

## Response Parameters

**Status code: 200**

**Table 4-51** Response body parameters

| Parameter | Type   | Description               |
|-----------|--------|---------------------------|
| job_id    | String | ID of an asynchronous job |

**Status code: 400**

**Table 4-52** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Incrementally import graph data. The edge file directory is **testbucket/demo\_movie/edges/** and the edge data set is in CSV format. The vertex file directory is **testbucket/demo\_movie/vertices/** and the vertex data set is in CSV format.

POST http://Endpoint/v2/{project\_id}/graphs/{graph\_id}/import-graph

```
{
  "edgeset_path" : "testbucket/demo_movie/edges/",
  "edgeset_format" : "csv",
  "vertexset_path" : "testbucket/demo_movie/vertices/",
  "vertexset_format" : "csv",
  "schema_path" : "testbucket/demo_movie/incremental_data_schema.xml",
  "log_dir" : "testbucket/importlogdir",
  "parallel_edge" : {
    "action" : "override",
    "ignore_label" : true
  },
  "delimiter" : ";",
  "trim_quote" : "\"",
  "offline" : false
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_msg" : "parameter format error",
  "error_code" : "GES.8013"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 4.2.8 Exporting a Graph

### Function

This API is used to unbind an EIP.

#### NOTE

If you choose to export CSV files to your local host, the files are opened using the spreadsheet software by default. You are advised to open the files in a text editor. If the data contains special characters such as plus signs (+), minus signs (-), equal signs (=), and at signs (@), the data will be parsed into formulas by the software. To ensure system security, pay attention to the following when opening such files:

1. Do not select **Enable Dynamic Data Exchange Server Launch (not recommended)**.
2. Do not select **Enable** or **Yes** if a dialog box indicating a security issue is displayed.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/export-graph

**Table 4-53** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-54** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-55** Request body parameters

| Parameter         | Mandatory | Type                            | Description  |
|-------------------|-----------|---------------------------------|--|
| graph_export_path | Yes       | String                          | OBS path to which a graph is exported  |
| edge_set_name     | Yes       | String                          | Exported edge file name  |
| vertex_set_name   | Yes       | String                          | Exported vertex file name  |
| schema_name       | Yes       | String                          | Name of the exported metadata file   |
| paginate          | No        | <a href="#">paginate</a> object | Pagination-related parameters. In version 2.3.11 or later, graphs are exported on multiple pages by default. |

**Table 4-56** paginate

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| enable    | No        | Boolean | Whether to enable pagination. The default value is <b>true</b> . If pagination is not required, set this parameter to <b>false</b> .<br><b>NOTE</b><br>This parameter is unavailable for database editions. |

| Parameter          | Mandatory | Type    | Description   |
|--------------------|-----------|---------|---|
| row_count_per_file | No        | Integer | Maximum number of rows in each file when graphs are exported by page. The default value is <b>1000000</b> . |
| num_thread         | No        | Integer | Number of concurrent threads when graphs are exported by page. The default value is <b>8</b> .              |

## Response Parameters

Status code: 200

Table 4-57 Response body parameters

| Parameter | Type   | Description               |
|-----------|--------|---------------------------|
| job_id    | String | ID of an asynchronous job |

Status code: 400

Table 4-58 Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Export a graph. The OBS path for exporting the graph is **demo\_movie/**, the name of the exported edge file is **set\_edge.csv**, the name of the exported vertex file is **set\_vertex.csv**, and the name of the exported metadata file is **set\_schema.xml**.

```
POST http://Endpoint/v2/{project_id}/graphs{graph_id}/export-graph
{
```

```
"graph_export_path" : "demo_movie/",  
"edge_set_name" : "set_edge.csv",  
"vertex_set_name" : "set_vertex.csv",  
"schema_name" : "set_schema.xml"  
}
```

## Example Response

### Status code: 200

Example response for a successful request

```
{  
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"  
}
```

### Status code: 400

Example response for a failed request

```
{  
  "error_msg" : "graph [demo] is not found",  
  "error_code" : "GES.8011"  
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.9 Clearing a Graph

### Function

This API is used to clear all data in a graph.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/clear-graph



**Table 4-59** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 4-60** Query parameters

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| clear_metadata | No        | Boolean | Whether to clear graph metadata. Set this parameter to <b>true</b> . The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> . <ul style="list-style-type: none"> <li><b>true</b>: The metadata will be cleared.</li> <li><b>false</b>: The metadata will not be cleared.</li> </ul> |

## Request Parameters

**Table 4-61** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-62** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---|
| job_id    | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

**Status code: 400**

**Table 4-63** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Clear all data in a graph.

```
POST http://Endpoint/v2/{project_id}/graphs/{graph_id}/clear-graph?clear_metadata=true
{ }
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg" : "graph [demo] is not found",
  "error_code" : "GES.8012"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.10 Upgrading a Graph

### Function

This API is used to upgrade a graph. The GES version is periodically upgraded. You can upgrade your graphs as required.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/upgrade

**Table 4-64** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-65** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-66** Request body parameters

| Parameter       | Mandatory | Type    | Description  |
|-----------------|-----------|---------|--|
| upgrade_version | Yes       | String  | Target version, which must be later than the current version   |
| force_upgrade   | No        | Boolean | Whether to upgrade forcibly. The value is <b>true</b> or <b>false</b> , and the default value is <b>false</b> . <ul style="list-style-type: none"> <li><b>true</b>: forcible upgrades, which will interrupt running tasks, such as long algorithm execution tasks. As a result, a small number of requests may fail.</li> <li><b>false</b>: non-forcible upgrades, which will wait for running services to complete. The upgrade process may be slow.</li> </ul> |

## Response Parameters

Status code: 200

**Table 4-67** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---|
| job_id    | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

**Status code: 400**

**Table 4-68** Response body parameters

| Parameter  | Type   | Description   |
|------------|--------|---|
| error_code | String | System prompt code.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Upgrade a graph to version 1.1.8 as needed.

```
POST http://Endpoint/v2/{project_id}/graphs{graph_id}/upgrade
{
  "upgrade_version" : "1.1.8",
  "force_upgrade" : false
}
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_msg" : "graph [demo] is not found",
}
```

```
"error_code" : "GES.8011"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.11 Binding an EIP

### Function

This API enables you to access GES by binding an elastic IP (EIP).

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/bind-eip

**Table 4-69** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-70** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-71** Request body parameters

| Parameter | Mandatory | Type   | Description                   |
|-----------|-----------|--------|-------------------------------|
| eip_id    | Yes       | String | ID of the elastic IP address. |

## Response Parameters

**Status code: 200**

None

**Status code: 400**

**Table 4-72** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Bind an EIP to access GES. The ID of the EIP is **02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042**.

```
POST http://Endpoint/v2/{project_id}/graphs{graph_id}/bind-eip
{
  "eip_id" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{ }
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg" : "graph [demo] is not found",
  "error_code" : "GES.8011"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.12 Unbinding an EIP

### Function

If you do not need to use the EIP, this API enabled you to unbind the EIP to release network resources.

### URI

```
POST /v2/{project_id}/graphs/{graph_id}/unbind-eip
```



**Table 4-73** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-74** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-75** Request body parameters

| Parameter | Mandatory | Type   | Description                   |
|-----------|-----------|--------|-------------------------------|
| eip_id    | Yes       | String | ID of the elastic IP address. |

## Response Parameters

**Status code: 200**

None

**Status code: 400**

**Table 4-76** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Unbind an EIP to release network resources. The ID of the EIP is **02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042**.

```
POST http://Endpoint/v2/{project_id}/graphs{graph_id}/unbind-eip
{
  "eip_id": "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{ }
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg": "graph [demo] is not found",
  "error_code": "GES.8011"
}
```

### Status Code

| Return Value     | Description              |
|------------------|--------------------------|
| 400 Bad Request  | Request error            |
| 401 Unauthorized | Authorization failed     |
| 403 Forbidden    | No operation permissions |
| 404 Not Found    | No resources found       |

| Return Value              | Description           |
|---------------------------|-----------------------|
| 500 Internal Server Error | Internal server error |
| 503 Service Unavailable   | Service unavailable   |

## Error Code

See [Error Code](#).

## 4.2.13 Resizing a Graph

### Function

This API is used to resize a graph instance.

#### NOTE

After the graph is resized, you need to re-create all indexes.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/resize

**Table 4-77** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

### Request Parameters

**Table 4-78** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token.<br>It is used to obtain the permission to call APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-79** Request body parameter

| Parameter | Mandatory | Type                          | Description                                      |
|-----------|-----------|-------------------------------|--|
| resize    | Yes       | <a href="#">resize</a> object | Graph specifications after the graph is resized. |

**Table 4-80** resize

| Parameter             | Mandatory | Type   | Description  |
|-----------------------|-----------|--------|--|
| graph_size_type_index | Yes       | String | Graph flavor. Currently, the value can be <b>2</b> , <b>3</b> , <b>4</b> , or <b>5</b> , indicating <b>10-million-edge</b> , <b>100-million-edge</b> , <b>1-billion-edge</b> , or <b>10-billion-edge</b> , respectively. |

## Response Parameters

**Status code: 200**

**Table 4-81** Response body parameters

| Parameter | Type   | Description  |
|-----------|--------|--|
| job_id    | String | Indicates the ID of the resize job. This parameter is left blank when the request fails.<br><br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

**Status code: 400**

**Table 4-82** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Resize a graph instance. The graph size is **ten-million-edge**.

```
POST http://Endpoint/v2/{project_id}/graphs/{graph_id}/resize
{
  "resize" : {
    "graph_size_type_index" : "2"
  }
}
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg" : "graph [demo] is not found",
  "error_code" : "GES.8012"
}
```

### Status Codes

| Return Value     | Description           |
|------------------|-----------------------|
| 400 Bad Request  | Request error.        |
| 401 Unauthorized | Authorization failed. |

| Return Value              | Description               |
|---------------------------|---------------------------|
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 4.2.14 Restarting a Graph

#### Function

This API is used to forcibly start a graph, especially for running graphs or those are being imported, exported, and cleared. If a graph is forcibly restarted, asynchronous tasks of the graph are failed state and the graph is stopped and started.

#### URI

POST /v2/{project\_id}/graphs/{graph\_id}/restart

**Table 4-83** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-84** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

**Table 4-85** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---|
| job_id    | String | ID of a forcible restart job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |

**Status code: 400**

**Table 4-86** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Forcibly start a graph.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/restart
{ }
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg" : "The request is invalid.",
  "error_code" : "GES.7016"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.2.15 Expanding a Graph

### Function

This API is used to expand multiple secondary nodes dynamically. The expanded secondary nodes can process read requests, improving read performance.



 NOTE

1. This API is not supported by graphs of the 10,000-edge and 10-billion-edge types.
2. Graphs cannot be resized after expansion.
3. If you want to resize and expand the graph, resize the graph before you expand it.

## URI

POST /v2/{project\_id}/graphs/{graph\_id}/expand

**Table 4-87** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-88** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-89** Request body parameter

| Parameter | Mandatory | Type                          | Description                 |
|-----------|-----------|-------------------------------|-----------------------------|
| expand    | Yes       | <a href="#">expand</a> object | <b>expand</b> is an object. |

**Table 4-90** expand

| Parameter   | Mandatory | Type    | Description                   |
|-------------|-----------|---------|-------------------------------|
| replication | Yes       | Integer | Number of new nodes to expand |

## Response Parameters

**Status code: 200**

**Table 4-91** Response body parameters

| Parameter | Type   | Description  |
|-----------|--------|--|
| job_id    | String | ID of the expansion job. This parameter is left blank when the request fails.<br><br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |

**Status code: 400**

**Table 4-92** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Expand a graph. The number of new replicas is 1.

```
POST http://Endpoint/v2/{project_id}/graphs/{graph_id}/expand
{
  "expand":{
    "replication": 1
```

```
}  
}
```

## Example Response

### Status code: 200

Example response (successful request)

```
{  
  "job_id" : "ff8080816025a0a1016025a5a2700007"  
}
```

### Status code: 400

Example response (failed request)

```
{  
  "error_code" : "GES.7015",  
  "error_msg" : "The graph is not running or stopped."  
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 4.3 Backup Management

### 4.3.1 Viewing the List of All Backups

#### Function

This API is used to query a backup list according to search criteria. Before using this API:

#### URI

GET /v2/{project\_id}/graphs/backups

**Table 4-93** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 4-94** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |

## Request Parameters

**Table 4-95** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-96** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| backup_count | Integer | Total number of backups. This parameter is left blank if the request fails. |

| Parameter   | Type   | Description  |
|-------------|--|--|
| backup_list | Array of <a href="#">backup_list</a> objects | List of all backups under the current project ID. This parameter is left blank if the request fails. |

**Table 4-97** backup\_list

| Parameter             | Type   | Description   |
|-----------------------|--------|---|
| id                    | String | Indicates the backup ID.  |
| name                  | String | Backup name   |
| backup_method         | String | Backup method. The value can be <b>auto</b> , <b>manual</b> , or <b>import</b> .  |
| graph_id              | String | ID of the graph associated with the backup  |
| graph_name            | String | Name of the graph associated with the backup  |
| graph_status          | String | Status of the graph associated with the backup  |
| graph_size_type_index | String | Size of the graph associated with the backup  |
| data_store_version    | String | Version of the graph associated with the backup   |
| arch                  | String | CPU architecture of the graph node associated with the backup   |
| status                | String | Backup status: <ul style="list-style-type: none"> <li>● <b>backing_up</b>: indicates that a graph is being backed up.</li> <li>● <b>success</b>: indicates that a graph is successfully backed up.</li> <li>● <b>failed</b>: indicates that a graph fails to be backed up.</li> </ul> |
| start_timestamp       | Long   | Start timestamp of a backup job   |
| start_time            | String | Start time of a backup job.   |
| end_timestamp         | Long   | End timestamp of a backup job   |
| end_time              | String | Indicates the backup end time.  |
| size                  | Long   | Backup file size (MB)   |
| duration              | Long   | Backup duration (seconds)   |

**Status code: 400**

**Table 4-98** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

**Example Request**

Query the list of all backups.

```
GET https://Endpoint/v2/{project_id}/graphs/backups?offset=0&limit=3
```

**Example Response**

**Status code: 200**

Example response (successful request)

```
{
  "backup_count" : 3,
  "backup_list" : [ {
    "id" : "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name" : "ges060803_nodelete-20210608135513",
    "backup_method" : "manual",
    "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name" : "ges060803_nodelete",
    "graph_status" : "200",
    "graph_size_type_index" : "1",
    "data_store_version" : "2.2.21",
    "arch" : "x86_64",
    "status" : "success",
    "start_timestamp" : 1623160513000,
    "start_time" : "2021-06-08T13:55:13",
    "end_timestamp" : 1623160568000,
    "end_time" : "2021-06-08T13:56:08",
    "size" : 1,
    "duration" : 54,
    "encrypted" : false
  }, {
    "id" : "7ed3f51d-816d-4651-9129-fe21b64b5c91",
    "name" : "ges060803_nodelete_20210609203323_auto",
    "backup_method" : "auto",
    "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name" : "ges060803_nodelete",
    "graph_status" : "200",
    "graph_size_type_index" : "1",
    "data_store_version" : "2.2.21",
```

```

"arch" : "x86_64",
"status" : "success",
"start_timestamp" : 1623242004000,
"start_time" : "2021-06-09T12:33:24",
"end_timestamp" : 1623242004000,
"end_time" : "2021-06-09T12:33:24",
"size" : 1,
"duration" : 0,
"encrypted" : false
}, {
  "id" : "604bfb46-04dd-45fc-a9ae-df24a0705b9d",
  "name" : "ges060802_nodelete-20210608135523",
  "backup_method" : "manual",
  "graph_id" : "9b9a05c2-0cdb-41ac-b55f-93caffb0519a",
  "graph_name" : "ges060802_nodelete",
  "graph_status" : "400",
  "graph_size_type_index" : "0",
  "data_store_version" : "2.2.23",
  "arch" : "x86_64",
  "status" : "success",
  "start_timestamp" : 1623160524000,
  "start_time" : "2021-06-08T13:55:24",
  "end_timestamp" : 1623160577000,
  "end_time" : "2021-06-08T13:56:17",
  "size" : 1,
  "duration" : 53,
  "encrypted" : false
}]
}

```

**Status code: 400**

Example response (failed request)

```

{
  "error_code" : "GES.7006",
  "error_msg" : "The underlying graph engine has internal error."
}

```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.3.2 Viewing the Backup List of a Graph

### Function

This API is used to query the backup list of a graph.

### URI

GET /v2/{project\_id}/graphs/{graph\_id}/backups

**Table 4-99** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 4-100** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |

### Request Parameters

**Table 4-101** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |



## Response Parameters

Status code: 200

**Table 4-102** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--|--|
| backup_count | Integer                                      | Total number of backups. This parameter is left blank if the request fails.                          |
| backup_list  | Array of <a href="#">backup_list</a> objects | List of all backups under the current project ID. This parameter is left blank if the request fails. |

**Table 4-103** backup\_list

| Parameter             | Type   | Description   |
|-----------------------|--------|---|
| id                    | String | Backup ID   |
| name                  | String | Backup name   |
| backup_method         | String | Backup method. The value can be <b>auto</b> , <b>manual</b> , or <b>import</b> .  |
| graph_id              | String | ID of the graph associated with the backup  |
| graph_name            | String | Name of the graph associated with the backup  |
| graph_status          | String | Status of the graph associated with the backup  |
| graph_size_type_index | String | Size of the graph associated with the backup  |
| data_store_version    | String | Version of the graph associated with the backup   |
| arch                  | String | CPU architecture of the graph node associated with the backup   |
| status                | String | Backup status: <ul style="list-style-type: none"> <li>● <b>backing_up</b>: indicates that a graph is being backed up.</li> <li>● <b>success</b>: indicates that a graph is successfully backed up.</li> <li>● <b>failed</b>: indicates that a graph fails to be backed up.</li> </ul> |
| start_timestamp       | Long   | Start timestamp of a backup job   |
| start_time            | String | Start time of a backup job.   |

| Parameter     | Type   | Description                    |
|---------------|--------|--------------------------------|
| end_timestamp | Long   | End timestamp of a backup job  |
| end_time      | String | Indicates the backup end time. |
| size          | Long   | Backup file size (MB)          |
| duration      | Long   | Backup duration (seconds)      |

**Status code: 400**

**Table 4-104** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Query the backup list of a graph.

```
GET https://Endpoint/v2/{project_id}/graphs/{graph_id}/backups?offset=0&limit=2
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "backup_count" : 2,
  "backup_list" : [ {
    "id" : "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name" : "ges060803_nodelete-20210608135513",
    "backup_method" : "manual",
    "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name" : "ges060803_nodelete",
    "graph_status" : "200",
    "graph_size_type_index" : "1",
    "data_store_version" : "2.2.21",
    "arch" : "x86_64",
    "status" : "success",
```

```

"start_timestamp" : 1623160513000,
"start_time" : "2021-06-08T13:55:13",
"end_timestamp" : 1623160568000,
"end_time" : "2021-06-08T13:56:08",
"size" : 1,
"duration" : 54,
"encrypted" : false
}, {
  "id" : "7ed3f51d-816d-4651-9129-fe21b64b5c91",
  "name" : "ges060803_nodelete_20210609203323_auto",
  "backup_method" : "auto",
  "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
  "graph_name" : "ges060803_nodelete",
  "graph_status" : "200",
  "graph_size_type_index" : "1",
  "data_store_version" : "2.2.21",
  "arch" : "x86_64",
  "status" : "success",
  "start_timestamp" : 1623242004000,
  "start_time" : "2021-06-09T12:33:24",
  "end_timestamp" : 1623242004000,
  "end_time" : "2021-06-09T12:33:24",
  "size" : 1,
  "duration" : 0,
  "encrypted" : false
} ]
}

```

**Status code: 400**

Example response (failed request)

```

{
  "error_code" : "GES.7000",
  "error_msg" : "The graph does not exist or has been deleted."
}

```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

### 4.3.3 Adding a Backup

#### Function

Add a backup. If data in the current graph is incorrect or faulty, you can start the backup graph to restore the data.

#### URI

POST /v2/{project\_id}/graphs/{graph\_id}/backups

**Table 4-105** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

#### Request Parameters

**Table 4-106** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

#### Response Parameters

Status code: 200

**Table 4-107** Response body parameters

| Parameter | Type   | Description  |
|-----------|--------|--|
| job_id    | String | ID of the graph backup job.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |
| backup_id | String | Graph backup ID  |

**Status code: 400**

**Table 4-108** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Add a graph backup.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/backups
{ }
```

### Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id" : "ff8080815f9a3c84015f9a438ff70001"
  "backup_id":"0000b54f-2f1d-47a5-94bc-04f0d8a2e2db"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_code" : "GES.7000",
```

```
"error_msg" : "The graph does not exist or has been deleted."
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

### 4.3.4 Deleting a Backup

#### Function

This API is used to delete a backup.

#### URI

DELETE /v2/{project\_id}/graphs/{graph\_id}/backups/{backup\_id}

**Table 4-109** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| backup_id  | Yes       | String | Graph backup ID  |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-110** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

None

**Status code: 400**

**Table 4-111** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Delete a backup.

```
DELETE https://Endpoint/v2/{project_id}/graphs/{graph_id}/backups/{backupId}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{ }
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg" : "Parameter error!",
  "error_code" : "GES.0001"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.3.5 Exporting a Backup

### Function

This API is used to export a GES graph instance backup to OBS.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/backups/export

**Table 4-112** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |



## Request Parameters

**Table 4-113** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-114** Request body parameters

| Parameter   | Mandatory | Type   | Description        |
|-------------|-----------|--------|--------------------|
| backup_id   | Yes       | String | Backup ID          |
| export_path | Yes       | String | Export path in OBS |

## Response Parameters

**Status code: 200**

**Table 4-115** Response body parameter

| Parameter | Type   | Description                |
|-----------|--------|----------------------------|
| job_id    | String | ID of the asynchronous job |

**Status code: 400**

**Table 4-116** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| error_msg | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Export a backup.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/backups/export
{
  "backup_id": "{backup_id}",
  "export_path": "{obs_path}"
}
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id": "ff8080815f9a3c84015f9a438ff70001"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_msg": "Parameter error!",
  "error_code": "GES.0001"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 4.3.6 Importing a Backup

### Function

This API is used to import a GES graph instance backup from OBS.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/backups/import

**Table 4-117** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

### Request Parameters

**Table 4-118** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-119** Request body parameter

| Parameter   | Mandatory | Type   | Description                                |
|-------------|-----------|--------|--|
| import_path | Yes       | String | OBS path the backup is to be imported from |

### Response Parameters

Status code: 200

**Table 4-120** Response body parameter

| Parameter | Type   | Description                |
|-----------|--------|----------------------------|
| job_id    | String | ID of the asynchronous job |

**Status code: 400**

**Table 4-121** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |
| error_msg  | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |

### Example Request

Import a backup.

```
POST https://Endpoint/v2/{project_id}/graphs/{graph_id}/backups/import
{
  "import_path" : "{obs_path}"
}
```

### Example Response

**Status code: 200**

Example response for a successful request

```
{
  "job_id" : "ff8080815f9a3c84015f9a438ff70001"
}
```

**Status code: 400**

Example response for a failed request

```
{
  "error_msg" : "Parameter error!",
  "error_code" : "GES.0001"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

# 4.4 Metadata Management

## 4.4.1 Constraints

[Table 4-122](#) and [Table 4-123](#) list the metadata types.

**Table 4-122** Metadata property constraints

| Data Type  | Constraints   |
|------------|---|
| char       | <ul style="list-style-type: none"> <li>• Less than '&lt;'</li> <li>• Greater than '&gt;'</li> <li>• Equal to '='</li> <li>• Not equal to '!='</li> <li>• In range 'range'</li> <li>• Greater than or equal to '&gt;='</li> <li>• Less than or equal to '&lt;='</li> </ul> |
| char array | <ul style="list-style-type: none"> <li>• Less than '&lt;'</li> <li>• Greater than '&gt;'</li> <li>• Equal to '='</li> <li>• Not equal to '!='</li> <li>• In range 'range'</li> <li>• Greater than or equal to '&gt;='</li> <li>• Less than or equal to '&lt;='</li> </ul> |

| Data Type | Constraints  |
|-----------|--|
| float     | <ul style="list-style-type: none"> <li>● Less than '&lt;'</li> <li>● Greater than '&gt;'</li> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> <li>● In range 'range'</li> <li>● Greater than or equal to '&gt;='</li> <li>● Less than or equal to '&lt;='</li> </ul> |
| double    | <ul style="list-style-type: none"> <li>● Less than '&lt;'</li> <li>● Greater than '&gt;'</li> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> <li>● In range 'range'</li> <li>● Greater than or equal to '&gt;='</li> <li>● Less than or equal to '&lt;='</li> </ul> |
| bool      | <ul style="list-style-type: none"> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> </ul>   |
| long      | <ul style="list-style-type: none"> <li>● Less than '&lt;'</li> <li>● Greater than '&gt;'</li> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> <li>● In range 'range'</li> <li>● Greater than or equal to '&gt;='</li> <li>● Less than or equal to '&lt;='</li> </ul> |
| int       | <ul style="list-style-type: none"> <li>● Less than '&lt;'</li> <li>● Greater than '&gt;'</li> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> <li>● In range 'range'</li> <li>● Greater than or equal to '&gt;='</li> <li>● Less than or equal to '&lt;='</li> </ul> |
| date      | <ul style="list-style-type: none"> <li>● Less than '&lt;'</li> <li>● Greater than '&gt;'</li> <li>● Equal to '='</li> <li>● Not equal to '! ='</li> <li>● In range 'range'</li> <li>● Greater than or equal to '&gt;='</li> <li>● Less than or equal to '&lt;='</li> </ul> |

| Data Type | Constraints   |
|-----------|---|
| enum      | <ul style="list-style-type: none"> <li>• Equal to '='</li> <li>• Not equal to '!='</li> </ul>   |
| string    | <ul style="list-style-type: none"> <li>• Less than '&lt;'</li> <li>• Greater than '&gt;'</li> <li>• Equal to '='</li> <li>• Not equal to '!='</li> <li>• In range 'range'</li> <li>• Greater than or equal to '&gt;='</li> <li>• Less than or equal to '&lt;='</li> </ul> |

**Table 4-123** Property-level constraints

| Property Level                   | Constraints | Description                     |
|----------------------------------|-------------|---------------------------------|
| Single value/<br>Multiple values | has         | This property is contained.     |
| Single value/<br>Multiple values | hasNot      | This property is not contained. |

## 4.4.2 Querying the Metadata List

### Function

Query the metadata list.

### URI

GET /v2/{project\_id}/graphs/metadatas

**Table 4-124** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 4-125** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |

## Request Parameters

**Table 4-126** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

**Table 4-127** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--|---|
| schema_count | Integer                                      | Number of returned metadata files. This parameter is left blank if the request fails.                       |
| schema_list  | Array of <a href="#">schema_list</a> objects | List of all metadata files under the current project ID. This parameter is left blank if the request fails. |

**Table 4-128** schema\_list

| Parameter | Type   | Description |
|-----------|--------|-------------|
| id        | String | Metadata ID |



| Parameter        | Type   | Description                       |
|------------------|--------|-----------------------------------|
| name             | String | Metadata name                     |
| start_time       | String | Metadata creation time            |
| last_update_time | String | Last update time of the metadata  |
| description      | String | Metadata description              |
| metadata_path    | String | Metadata path                     |
| status           | String | Whether the metadata is available |

**Status code: 500**

**Table 4-129** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Query the metadata list.

```
GET https://Endpoint/v2/{project_id}/graphs/metadatas?offset=0&limit=2
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "schema_count" : 2,
  "schema_list" : [ {
    "start_time" : "2022-01-21T10:13:31",
    "last_update_time" : "2022-01-21T10:13:31",
    "encrypted" : true,
    "name" : "schema_748e",
```

```

    "description" : "xxxxx",
    "id" : "6634c50e-13aa-4395-8088-6b327f7da694",
    "metadata_path" : "devdata/schema_748e.xml",
    "status" : "200"
  }, {
    "start_time" : "2022-04-12T03:15:17",
    "last_update_time" : "2022-11-16T08:18:32",
    "encrypted" : false,
    "name" : "unionsdk_schema",
    "id" : "6b74069d-3cf3-4cc0-9118-2478e23b87aa",
    "metadata_path" : "devdata/unionsdk/unionsdk_schema.xml",
    "status" : "200"
  }
}

```

**Status code: 500**

Example response (failed request)

```

{
  "error_code" : "GES.7006",
  "error_msg" : "The underlying graph engine has internal error."
}

```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

### 4.4.3 Querying Metadata

#### Function

This API is used to query the metadata of a graph.

#### URI

GET /v2/{project\_id}/graphs/metadatas/{metadata\_id}

**Table 4-130** URI parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| project_id  | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| metadata_id | Yes       | String | Metadata ID  |

## Request Parameters

**Table 4-131** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-132** Response body parameters

| Parameter    | Type                                | Description                                      |
|--------------|-------------------------------------|--|
| ges_metadata | <a href="#">ges_metadata</a> object | Object for storing metadata message information. |

**Table 4-133** ges\_metadata

| Parameter | Type                                    | Description              |
|-----------|---|--------------------------|
| labels    | Array of <a href="#">labels</a> objects | Label data structure set |

**Table 4-134** labels

| Parameter  | Type                                | Description  |
|------------|-------------------------------------|--------------|
| name       | String                              | Label name   |
| properties | Array of Map<String,String> objects | Property map |

**Status code: 400**

**Table 4-135** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Query metadata of a graph.

```
GET https://Endpoint/v2/{project_id}/graphs/metadatas/{metadata_id}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "encrypted": false,
  "ges_metadata": {
    "labels": [ {
      "name": "friends"
    }, {
      "name": "movie",
      "properties": [ {
        "data_type": "string",
        "name": "ChineseTitle",
        "cardinality": "single"
      }, {
        "data_type": "int",
        "name": "Year",

```

```
    "cardinality" : "single"
  }, {
    "data_type" : "string",
    "name" : "Genres",
    "cardinality" : "set"
  }
]
}, {
  "name" : "user",
  "properties" : [ {
    "data_type" : "string",
    "name" : "ChineseName",
    "cardinality" : "single"
  }, {
    "type_name1" : "F",
    "type_name2" : "M",
    "data_type" : "enum",
    "name" : "Gender",
    "type_name_count" : "2",
    "cardinality" : "single"
  }, {
    "type_name1" : "Under 18",
    "type_name2" : "18-24",
    "type_name3" : "25-34",
    "type_name4" : "35-44",
    "type_name5" : "45-49",
    "type_name6" : "50-55",
    "type_name7" : "56+",
    "data_type" : "enum",
    "name" : "Age",
    "type_name_count" : "7",
    "cardinality" : "single"
  }, {
    "data_type" : "string",
    "name" : "Occupation",
    "cardinality" : "single"
  }, {
    "data_type" : "char array",
    "name" : "Zip-code",
    "max_data_size" : "12",
    "cardinality" : "single"
  }
]
}, {
  "name" : "rate",
  "properties" : [ {
    "data_type" : "int",
    "name" : "Score",
    "cardinality" : "single"
  }, {
    "data_type" : "date",
    "name" : "Datetime",
    "cardinality" : "single"
  }
]
}
}
```

**Status code: 400**

## Example response (failed request)

```
{
  "error_msg" : "6b74069d-3cf3-4cc0-9118-2478e23b87a does not exist.",
  "error_code" : "GES.2067"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.4.4 Adding Metadata

### Function

Add a metadata file.

### URI

POST /v2/{project\_id}/graphs/metadatas

**Table 4-136** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 4-137** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-138** Request body parameters

| Parameter     | Mandatory | Type                                | Description  |
|---------------|-----------|-------------------------------------|--|
| metadata_path | Yes       | String                              | Path for storing the metadata  |
| name          | Yes       | String                              | Metadata name, which contains 1 to 64 characters consisting of only letters, digits, and underscores (_) |
| description   | Yes       | String                              | Metadata description   |
| is_overwrite  | Yes       | Boolean                             | Whether to overwrite the file.   |
| ges_metadata  | Yes       | <a href="#">ges_metadata</a> object | Object for storing metadata message information.   |

**Table 4-139** ges\_metadata

| Parameter | Mandatory | Type                                    | Description |
|-----------|-----------|---|-------------|
| labels    | Yes       | Array of <a href="#">labels</a> objects | Label list  |

**Table 4-140** labels

| Parameter | Mandatory | Type   | Description     |
|-----------|-----------|--------|-----------------|
| name      | No        | String | Name of a label |

| Parameter  | Mandatory | Type                                | Description        |
|------------|-----------|-------------------------------------|--------------------|
| properties | No        | Array of Map<String,String> objects | Label property map |

## Response Parameters

Status code: 200

Table 4-141 Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---------------|
| id        | String | Metadata ID   |
| name      | String | Metadata name |

Status code: 400

Table 4-142 Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Add metadata. The metadata is stored in **devdata/unionsdk/unionsdk\_test.xml**. The metadata does not overwrite existing data, the metadata name is **unionsdk\_test**, and the metadata description is **test**.

```
POST https://Endpoint/v2/{project_id}/graphs/metadatas
{
  "metadata_path" : "devdata/unionsdk/unionsdk_test.xml",
  "is_overwrite" : false,
  "name" : "unionsdk_test",
  "description": " test",
}
```



```

"ges_metadata" : {
  "labels" : [ {
    "name" : "friends"
  }, {
    "name" : "movie",
    "properties" : [ {
      "dataType" : "string",
      "name" : "ChineseTitle",
      "cardinality" : "single"
    }, {
      "dataType" : "int",
      "name" : "Year",
      "cardinality" : "single"
    }, {
      "dataType" : "string",
      "name" : "Genres",
      "cardinality" : "set"
    }
  ]
}, {
  "name" : "user",
  "properties" : [ {
    "dataType" : "string",
    "name" : "ChineseName",
    "cardinality" : "single"
  }, {
    "typeName1" : "F",
    "typeName2" : "M",
    "typeNameCount" : "2",
    "dataType" : "enum",
    "name" : "Gender",
    "cardinality" : "single"
  }, {
    "typeName1" : "Under 18",
    "typeName2" : "18-24",
    "typeName3" : "25-34",
    "typeName4" : "35-44",
    "typeNameCount" : "7",
    "dataType" : "enum",
    "name" : "Age",
    "typeName5" : "45-49",
    "typeName6" : "50-55",
    "cardinality" : "single",
    "typeName7" : "56+"
  }, {
    "dataType" : "string",
    "name" : "Occupation",
    "cardinality" : "single"
  }, {
    "dataType" : "char array",
    "name" : "Zip-code",
    "maxDataSize" : "12",
    "cardinality" : "single"
  }
  ]
}, {
  "name" : "rate",
  "properties" : [ {
    "dataType" : "int",
    "name" : "Score",
    "cardinality" : "single"
  }, {
    "dataType" : "date",
    "name" : "Datetime",
    "cardinality" : "single"
  }
  ]
}
}
}

```

## Example Response

### Status code: 200

Example response (successful request)

```
{  
  "id" : "ff8080815f9a3c84015f9a438ff70001",  
  "name" : "movie_schema"  
}
```

### Status code: 400

Example response (failed request)

```
{  
  "error_msg" : "The metadata file already exists.",  
  "error_code" : "GES.2067"  
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.4.5 Deleting Metadata

### Function

Delete a metadata file.

### URI

DELETE /v2/{project\_id}/graphs/metadatas/{metadata\_id}

**Table 4-143** URI parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| project_id  | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| metadata_id | Yes       | String | Metadata ID  |

## Request Parameters

**Table 4-144** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 400**

None

**Status code: 200**

**Table 4-145** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

## Example Request

Delete metadata of a graph.

```
DELETE /v2/{project_id}/graphs/metadatas/{metadata_id}
```

## Example Response

**Status code: 200**

OK

```
{}
```

**Status code: 400**

Bad Request

```
{  
  "error_code" : "GES.7024",  
  "error_msg" : "The metadata is not exist or has been deleted."  
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.4.6 Importing Metadata from OBS

### Function

This API is used to import metadata from OBS.

### URI

```
POST /v2/{project_id}/graphs/metadata/upload-from-obs
```

**Table 4-146** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 4-147** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-148** Request body parameters

| Parameter     | Mandatory | Type   | Description                   |
|---------------|-----------|--------|-------------------------------|
| metadata_path | Yes       | String | Path for storing the metadata |
| name          | Yes       | String | Metadata name                 |
| description   | No        | String | Metadata description          |

## Response Parameters

**Status code: 200**

**Table 4-149** Response body parameters

| Parameter | Type   | Description   |
|-----------|--------|---------------|
| id        | String | Metadata ID   |
| name      | String | Metadata name |

**Status code: 400**

**Table 4-150** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

**Example Request**

Import metadata from OBS. The metadata is stored in **devdata/unionsdk/schema.xml** and the metadata name is **test\_schema**.

```
POST https://Endpoint/v2/{project_id}/graphs/metadata/upload-from-obs
{
  "metadata_path" : "devdata/unionsdk/schema.xml",
  "name" : "test_schema",
  "description" : "xx"
}
```

**Example Response**

**Status code: 200**

Example response for a successful request

```
{
  "id" : "d30d2e94-f2ee-4344-af49-eb27fd002eea",
  "name" : "test_schema"
}
```

**Status code: 400**

Example response for failed request

```
{
  "error_msg" : "test_schema The name already exists.",
  "error_code" : "GES.2067"
}
```

**Status Code**

| Return Value    | Description   |
|-----------------|---------------|
| 400 Bad Request | Request error |

| Return Value              | Description              |
|---------------------------|--------------------------|
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

# 4.5 Task Center

## 4.5.1 Querying Job Status on the Management Plane

### Function

This interface is used to query the execution status of a task, such as ECS creation, ECS deletion, ECS batch operation, and NIC operation. Asynchronous APIs that are used to create, stop, start, delete, and import graphs will return job IDs after commands are sent. You can query the job execution status according to the job IDs.

### URI

GET /v2/{project\_id}/graphs/{graph\_id}/jobs/{job\_id}/status

**Table 4-151** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |
| job_id     | Yes       | String | ID of the asynchronous job   |

## Request Parameters

**Table 4-152** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-153** Response body parameters

| Parameter     | Type                              | Description  |
|---------------|-----------------------------------|--|
| job_id        | String                            | Job ID   |
| status        | String                            | The task status. <ul style="list-style-type: none"> <li>• <b>pending</b></li> <li>• <b>running</b></li> <li>• <b>success</b></li> <li>• <b>failed</b></li> </ul> |
| job_type      | String                            | Task type.   |
| job_name      | String                            | Task name  |
| related_graph | String                            | Associated graph name  |
| begin_time    | String                            | Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.   |
| end_time      | String                            | Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.   |
| job_detail    | <a href="#">job_detail</a> object | This parameter is returned only when <b>jobName</b> is set to <b>ImportGraph</b> and is used to display graph import details.                                    |
| fail_reason   | String                            | Job failure cause  |
| job_progress  | Double                            | Job execution progress. It is a reserved field, and not used currently.  |



**Table 4-154** job\_detail

| Parameter      | Type  | Description                          |
|----------------|---|--------------------------------------|
| schema_path    | Array of <a href="#">schema_path</a> objects    | Path for storing metadata            |
| edgeset_path   | Array of <a href="#">edgeset_path</a> objects   | Path for storing the edge data set   |
| vertexset_path | Array of <a href="#">vertexset_path</a> objects | Path for storing the vertex data set |

**Table 4-155** schema\_path

| Parameter        | Type   | Description  |
|------------------|--------|--|
| path             | String | OBS storage path   |
| log              | String | Import log   |
| status           | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |
| cause            | String | Import failure cause   |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version.  |

**Table 4-156** edgeset\_path

| Parameter | Type   | Description      |
|-----------|--------|------------------|
| path      | String | OBS storage path |
| log       | String | Import log       |

| Parameter        | Type   | Description  |
|------------------|--------|--|
| status           | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause            | String | Import failure cause   |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version.  |

**Table 4-157** vertexset\_path

| Parameter        | Type   | Description  |
|------------------|--------|--|
| path             | String | OBS storage path   |
| log              | String | Import log   |
| status           | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause            | String | Import failure cause   |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version.  |

**Status code: 400**

**Table 4-158** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Query the execution status of a job.

```
GET https://Endpoint/v2/{project_id}/graphs/{graph_id}/jobs/{job_id}/status
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_id" : "ff80808167f09aaa0167f19b35ec0305",
  "status" : "success",
  "job_type" : "GraphManagement",
  "job_name" : "ImportGraph",
  "related_graph" : "GES_UI_AUTO",
  "begin_time" : "2018-11-27T21:39:00",
  "end_time" : "2018-11-27T21:39:56",
  "job_detail" : {
    "vertexset_path" : [ {
      "path" : "ges-ui/auDatas/list_set_vertex.csv",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ],
    "edgeset_path" : [ {
      "path" : "ges-ui/auDatas/list_set_edge.csv",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ],
    "schema_path" : [ {
      "path" : "ges-ui/auDatas/list_set_schema.xml",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ]
  },
  "job_progress" : 0
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg": "can not find job, jobId is ff808081646e81d40164c5fb414b2b1a1",
  "error_code": "GES.8301"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.5.2 Querying Job Details in the Job Center

### Function

This API is used to query asynchronous job details in the job center on the management plane. Asynchronous jobs include creating, closing, starting, deleting, adding, importing, exporting, and upgrading graphs, as well as adding backups.

### URI

GET /v2/{project\_id}/graphs/jobs

**Table 4-159** URI parameters

| Parameter  | Mandatory | Type   | Description                                       |
|------------|-----------|--------|---|
| project_id | Yes       | String | Project ID, which is used for resource isolation. |

**Table 4-160** Query parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| end_time   | No        | String | Job end date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.                               |
| graph_name | No        | String | Associated graph name   |
| limit      | No        | String | Maximum number of resources displayed on a single page. The default value is <b>10</b> .  |
| offset     | No        | String | Start position of the request. The default value is <b>0</b> .  |
| start_time | No        | String | Job start date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.                             |
| status     | No        | String | The task status. Options: <ul style="list-style-type: none"> <li>• running</li> <li>• waiting</li> <li>• success</li> <li>• failed</li> </ul> |

## Request Parameters

**Table 4-161** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 4-162** Response body parameters

| Parameter | Type                                      | Description          |
|-----------|---|----------------------|
| job_count | Integer                                   | Total number of jobs |
| job_list  | Array of <a href="#">job_list</a> objects | Task list            |

**Table 4-163** job\_list

| Parameter     | Type                              | Description  |
|---------------|-----------------------------------|--|
| job_id        | String                            | Job ID   |
| status        | String                            | The task status. <ul style="list-style-type: none"> <li>• <b>pending</b></li> <li>• <b>running</b></li> <li>• <b>success</b></li> <li>• <b>failed</b></li> </ul> |
| job_type      | String                            | Task type.   |
| job_name      | String                            | Task name  |
| related_graph | String                            | Associated graph name  |
| begin_time    | String                            | Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.   |
| end_time      | String                            | Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.   |
| job_detail    | <a href="#">job_detail</a> object | This parameter is returned only when <b>jobName</b> is set to <b>ImportGraph</b> and is used to display graph import details.                                    |
| fail_reason   | String                            | Job failure cause  |
| job_progress  | Double                            | Job execution progress. It is a reserved field, and not used currently.  |

**Table 4-164** job\_detail

| Parameter   | Type   | Description               |
|-------------|--|---------------------------|
| schema_path | Array of <a href="#">schema_path</a> objects | Path for storing metadata |

| Parameter      | Type  | Description                          |
|----------------|---|--------------------------------------|
| edgeset_path   | Array of <a href="#">edgeset_path</a> objects   | Path for storing the edge data set   |
| vertexset_path | Array of <a href="#">vertexset_path</a> objects | Path for storing the vertex data set |

**Table 4-165** schema\_path

| Parameter        | Type   | Description  |
|------------------|--------|--|
| path             | String | OBS storage path   |
| log              | String | Import log   |
| status           | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |
| cause            | String | Import failure cause   |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version.  |

**Table 4-166** edgeset\_path

| Parameter | Type   | Description  |
|-----------|--------|--|
| path      | String | OBS storage path   |
| log       | String | Import log   |
| status    | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |

| Parameter        | Type   | Description   |
|------------------|--------|---|
| cause            | String | Import failure cause  |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.                 |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.                    |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version. |

**Table 4-167** vertexset\_path

| Parameter        | Type   | Description  |
|------------------|--------|--|
| path             | String | OBS storage path   |
| log              | String | Import log   |
| status           | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause            | String | Import failure cause   |
| total_lines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failed_lines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successful_lines | Long   | Number of lines that are successfully imported. The value <b>-1</b> indicates that this field is not returned in the current version.  |

**Status code: 400**



**Table 4-168** Response body parameters

| Parameter  | Type   | Description  |
|------------|--------|--|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| error_msg  | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

### Example Request

Query asynchronous job details in the job center on the management plane.

```
GET https://Endpoint/v2/{project_id}/graphs/jobs?offset=0&limit=100
```

### Example Response

**Status code: 200**

Example response (successful request)

```
{
  "job_count" : 2,
  "job_list" : [ {
    "job_id" : "ff80808167bb90340167bc3c7b5b026a",
    "status" : "success",
    "job_type" : "GraphManagement",
    "job_name" : "ImportGraph",
    "related_graph" : "test1217",
    "begin_time" : "2018-12-17T12:55:40",
    "end_time" : "2018-12-17T12:56:32",
    "job_detail" : {
      "vertexset_path" : null,
      "edgeset_path" : [ {
        "path" : "hkmovie/edge.csv",
        "log" : null,
        "cause" : null,
        "status" : "success"
      } ],
      "schema_path" : [ {
        "path" : "hkmovie/schema.xml",
        "log" : null,
        "cause" : null,
        "status" : "success"
      } ]
    }
  }, {
    "job_id" : "ff80808167bb90340167bc5d0b1d0358",
    "status" : "success",
    "job_type" : "GraphManagement",
    "job_name" : "DeleteGraph",
    "related_graph" : "test1218",
  } ]
}
```

```
"begin_time" : "2018-12-17T13:31:14",
"end_time" : "2018-12-17T13:34:48",
"job_progress" : 0
}]
}
```

**Status code: 400**

Example response (failed request)

```
{
"error_msg" : "failed",
"error_code" : "GES.9999"
}
```

**Status Code**

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

**Error Code**

See [Error Code](#).

## 4.6 Plugin Management

### 4.6.1 Querying Scene Analysis Plugin Information

**Function**

This API is used to query the information about the application analysis capability in a scene, including information about the applications, parameters, and function details.

**URI**

GET /v2/{project\_id}/graphs/scenes

**Table 4-169** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 4-170** Query parameters

| Parameter        | Mandatory | Type   | Description  |
|------------------|-----------|--------|--|
| scene_name       | No        | String | Scene name. If only <b>scene_name</b> is specified, all application details in the specified scene will be returned. If only <b>scene_name</b> and <b>application_name</b> are specified, details about the applications requested by <b>application_name</b> will be returned. If <b>scene_name</b> , <b>application_name</b> , and <b>graph_id</b> are left empty, details of all scene applications will be returned. |
| application_name | No        | String | Application name. If only <b>scene_name</b> and <b>application_name</b> are specified, details about the applications requested by <b>application_name</b> will be returned. If <b>scene_name</b> , <b>application_name</b> , and <b>graph_id</b> are left empty, details of all scene applications will be returned.  |
| graph_id         | No        | String | Graph ID. If only <b>graph_id</b> is set, details about all the subscribed applications of the graph ID will be returned. If <b>scene_name</b> , <b>application_name</b> , and <b>graph_id</b> are left empty, details of all scene applications will be returned.   |

## Request Parameters

**Table 4-171** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

**Table 4-172** Response body parameters

| Parameter | Type                                     | Description                       |
|-----------|--|-----------------------------------|
| results   | Array of <a href="#">results</a> objects | Scene analysis plugin information |

**Table 4-173** results

| Parameter   | Type                                    | Description                              |
|-------------|---|--|
| scene       | String                                  | Scene name                               |
| name        | String                                  | Application name                         |
| params      | Array of <a href="#">params</a> objects | Parameter list                           |
| description | String                                  | Description of an application in a scene |

**Table 4-174** params

| Parameter | Type   | Description    |
|-----------|--------|----------------|
| name      | String | Parameter name |

| Parameter     | Type   | Description   |
|---------------|--------|---|
| type          | String | Parameter type. The value range is ["string","int"]. Currently, only <b>"string"</b> is supported.            |
| default_value | String | The value can be left empty or a specified value. If you left the value empty, the parameter is not nullable. |

**Status code: 400**

**Table 4-175** Response body parameters

| Parameter  | Type   | Description   |
|------------|--------|---|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| error_msg  | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |

## Example Request

Query the information about the application analysis capability in a scene.

```
GET /v2/{project_id}/graphs/scenes?scene_name=xxx&application_name=xxx&graph_id=xxx
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "results": [ {
    "name": "movie_recommendation",
    "description": "Recommend movies that friends are interested in. Graph constraints: (user)-[friends]->(user), (user)-[rates]->(movie)",
    "params": [ {
      "name": "user",
      "default_value": "",
      "type": "string"
    } ],
    "scene": "MovieSocialNetwork_V2"
  }, {
    "name": "friend_recommendation",
```

```

    "description": "Recommend people who you may be interested in (considering the relationships between
    potential friends and movie preference). Graph constraints: (user) -[friends]-> (user), (user) -[rates]->
    (movie)",
    "params": [ {
      "name": "user",
      "default_value": "",
      "type": "string"
    } ],
    "scene": "MovieSocialNetwork_V2"
  } ]
}

```

**Status code: 400**

Example response (failed request)

```

{
  "error_msg": "The request body or header is invalid.",
  "error_code": "GES.7016"
}

```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.6.2 Subscribing to a Scene Analysis Plugin

### Function

This API is used to subscribe to a scene analysis plugin so that you can use the function through the service plane APIs.

#### NOTE

A subscribed plugin cannot be subscribed to repeatedly. To update the plugin, cancel the subscription and subscribe to it again after update.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/scenes/register

**Table 4-176** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-177** Request header parameter

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| X-Auth-Token | Yes       | String | User token<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-178** Request body parameters

| Parameter | Mandatory | Type                                    | Description                    |
|-----------|-----------|---|--------------------------------|
| scenes    | No        | Array of <a href="#">scenes</a> objects | Scene you want to subscribe to |

**Table 4-179** scenes

| Parameter    | Mandatory | Type             | Description  |
|--------------|-----------|------------------|--|
| name         | No        | String           | Scene name   |
| applications | No        | Array of strings | List of applications you want to subscribe to (not supported currently). |

## Response Parameters

Status code: 200

**Table 4-180** Response body parameters

| Parameter | Type   | Description         |
|-----------|--------|---------------------|
| result    | String | Subscription result |

**Status code: 400**

**Table 4-181** Response body parameters

| Parameter  | Type   | Description   |
|------------|--------|---|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| error_msg  | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |

## Example Request

Subscribe to an application scenario analysis plugin. The scenario name is **Service ticket association analysis**, and the list of applications is **In-depth analysis of dynamic high-dimension relationships**.

```
POST /v2/{project_id}/graphs/{graph_id}/scenes/register
```

```
{
  "scenes": [ {
    "name": "Service ticket association analysis",
    "applications": ["In-depth analysis of dynamic high-dimension relationships"]
  } ]
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg": "graph [demo] is not found",
}
```



```
"error_code" : "GES.8402"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

## 4.6.3 Unsubscribing from a Scene Analysis Plugin

### Function

This API is used to unsubscribe from a scene analysis plugin. After the subscription is canceled, you cannot use the function through application service plane APIs.

### URI

POST /v2/{project\_id}/graphs/{graph\_id}/scenes/unregister

**Table 4-182** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 4-183** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 4-184** Request body parameters

| Parameter | Mandatory | Type                                    | Description                                 |
|-----------|-----------|---|---|
| scenes    | No        | Array of <a href="#">scenes</a> objects | List of scenes you want to unsubscribe from |

**Table 4-185** scenes

| Parameter    | Mandatory | Type             | Description                                       |
|--------------|-----------|------------------|---|
| name         | No        | String           | Scene name  |
| applications | No        | Array of strings | List of applications you want to unsubscribe from |

## Response Parameters

**Status code: 200**

**Table 4-186** Response body parameters

| Parameter | Type             | Description  |
|-----------|------------------|--|
| success   | Array of strings | Scene application that is successfully unsubscribed from |
| failure   | Array of strings | Scene application that fails to be unsubscribed from     |

**Status code: 400**

**Table 4-187** Response body parameters

| Parameter  | Type   | Description   |
|------------|--------|---|
| error_code | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| error_msg  | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |

**Example Request**

Unsubscribe from a scenario analysis plugin. The scenario name is **Service ticket association analysis**, and the list of applications is **In-depth analysis of dynamic high-dimension relationships**.

```
POST /v2/{project_id}/graphs/{graph_id}/scenes/unregister
{
  "scenes": [ {
    "name": "Service ticket association analysis",
    "applications": ["In-depth analysis of dynamic high-dimension relationships"]
  } ]
}
```

**Example Response**

**Status code: 200**

Example response (successful request)

```
{
  "success": [ "testCase" ],
  "failure": [ "xxx" ]
}
```

**Status code: 400**

Example response (failed request)

```
{
  "error_msg": "graph [demo] is not found",
  "error_code": "GES.8402"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

## Error Code

See [Error Code](#).

# 5 Service Plane APIs

---

## 5.1 Memory Edition

### 5.1.1 Vertex Operation APIs

#### 5.1.1.1 Querying Vertices Based on Filter Criteria

##### Function

This API is used to query vertices that meet filter criteria.

##### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?action\_id=query

**Table 5-1** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-2** Request body parameters

| Parameter     | Mandatory  | Type    | Description  |
|---------------|--|---------|--|
| labels        | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | String  | Filter criteria of the vertex type   |
| vertexFilters | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | Object  | Filter criteria, in JSONArray format. Vertices are filtered by property.                 |
| offset        | No   | Integer | Start position of the request  |
| limit         | No   | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| sorts         | No   | Object  | Result sorting property, in JSONArray format   |

**Table 5-3** sorts parameter description

| Parameter    | Mandatory  | Type   | Description  |
|--------------|--|--------|--|
| key          | Either <b>Key</b> or <b>propertyName</b> is mandatory. | String | Possible values are <b>id</b> , <b>label</b> , and <b>property</b> . These values indicate that IDs, labels, or properties are sorted.           |
| propertyName | Either <b>Key</b> or <b>propertyName</b> is mandatory. | String | Property name  |
| orderValue   | No   | String | Possible values are <b>incr</b> and <b>decr</b> , which indicate ascending and descending order respectively. The default value is <b>incr</b> . |

**Table 5-4** vertexFilters parameter structure

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---------------|
| propertyName | Yes       | String | Property name |

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| predicate | Yes       | String | Predicate. Available values are =, <, >, <=, >=, <b>range</b> , <b>has</b> , <b>hasNot</b> .<br><b>NOTE</b><br>If the property is of the composite type, such as list or set, the predicate can only be <b>has</b> or <b>hasNot</b> . |
| values    | No        | String | Property value.   |
| type      | No        | String | Logical operator of the filter criteria. Possible values are <b>and</b> and <b>or</b> . The default value is <b>and</b> .   |

## Response Parameters

**Table 5-5** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String  | System prompt code.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>  |
| jobId        | String  | ID of the vertex query job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType      | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

Query vertices that meet filter criteria. The start position of the request is **0**, the maximum number of resources on each page is **2**, the attribute criteria for filtering are **movie** and **user**, and the attribute name for filtering is **Age**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query
{
  "offset":0,
  "limit":2,
  "labels": ["movies",
            "user"],
  "vertexFilters":[{"propertyName":"Age",
                    "predicate":"=",
                    "values":["18-24"]}
]
}
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example 1 for vertexFilters

```
[
  {
    "propertyName":"Gender",
    "predicate":"=",
    "values":["F"]
  },
  {
    "propertyName":"Age",
    "predicate":"range",
    "values":["18-24","56+"],
    "type":"or"
  }
]
```

## Example Response

**Status code: 200**

Example of a successful response

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

**Status code: 400**

Example of a failed response

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |



| Return Value              | Description            |
|---------------------------|------------------------|
| 404 Not Found             | No resources found.    |
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

### 5.1.1.2 Querying Vertex Details

## Function

This API is used to query the vertex information (such as the label and property) based on the vertex ID.

## URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/detail?  
vertexIds={vertex\_ids}

**Table 5-6** URI parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .                              |
| graph_name | Yes       | String | Graph name  |
| vertex_ids | Yes       | String | IDs of the vertices to be queried. When multiple IDs are specified by <b>vertexIds</b> , separate the IDs with commas (,) in the URL. |

## Request Parameters

**Table 5-7** Request body parameter

| Parameter | Type | Description  |
|-----------|------|--|
| data      | List | Vertex details. For details, see <a href="#">Table 5-8</a> . |

**Table 5-8** data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| vertices  | List | Vertex result set. If no corresponding vertices are found, the value of <b>vertices</b> is empty. |

## Response Parameters

**Table 5-9** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results.  |

## Example Request

Query node information by node ID and return node details.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds=Ray
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example of a successful response

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "Ray",
        "label": "user",
        "properties": {
          "Occupation": [
```

```

        "college/grad student"
      ],
      "Name": [
        "Lei"
      ],
      "Zip-code": [
        "90241"
      ],
      "Gender": [
        "M"
      ],
      "Age": [
        "18-24"
      ]
    }
  ]
}

```

**Status code: 400**

Example of a failed response

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8204"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.1.3 Adding a Vertex**

**Function**

This API is used to add a vertex.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices

**Table 5-10** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-11** Request body parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| vertex     | Yes       | String | Vertex name   |
| label      | Yes       | String | Label of a vertex. If no label exists, set it to <b>__DEFAULT__</b> . |
| properties | No        | Json   | Value of each property  |

## Response Parameters

**Table 5-12** Parameter description

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Add a vertex named **Lily** and set the vertex label to **user**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices
{
  "vertex": "Lily",
  "label": "user",
  "properties": {
    "Age": [
      "under 18"
    ],
    "Gender": [
      "F"
    ],
    "Occupation": [
      "artist"
    ],
    "Zip-code": [
      "98133"
    ]
  }
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] already exists",
  "errorCode": "GES.8000"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.1.4 Deleting a Vertex

## Function

This API is used to delete a vertex.

## URI

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/{vertex\_id}

**Table 5-13** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| vertex_id  | Yes       | String | Vertex ID  |

## Response Parameters

**Table 5-14** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> . |

## Example Request

Delete a vertex.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/Lily
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.1.5 Updating a Vertex Property

#### Function

This API is used to update vertex property values. The operations include ADD, UPDATE, and DEL.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/{vertex\_id}/properties/  
action?action\_id={actionId}

**Table 5-15** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .   |
| graph_name | Yes       | String | Graph name   |
| vertex_id  | Yes       | String | Vertex ID  |
| actionId   | Yes       | String | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>UPDATE</b>: Update the value of a property.</li> <li>• <b>ADD</b>: Add the value to a property. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of <b>UPDATE</b>. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>DEL</b>: Delete a property value.</li> </ul> |

#### Request Parameters

**Table 5-16** Request body parameters

| Parameter  | Mandatory | Type   | Description            |
|------------|-----------|--------|------------------------|
| properties | Yes       | Object | Value of each property |
| label      | No        | String | Name of a label        |



## Response Parameters

**Table 5-17** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update the property value of a vertex. The value of the **Age** property is **under 18**, and the value of the **Gender** property is **F**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/Lily/properties/action?
action_id={actionId}
{
  "properties":{
    "Age":[
      "under 18"
    ],
    "Gender":[
      "F"
    ]
  }
}
```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8220"
}
```

**Status Code**

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

**Error Code**

See [Error Code](#).

**5.1.1.6 Querying Vertices in Batches**

**Function**

This API is used to query the vertex data (such as the labels and properties) in batches based on the vertex IDs.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-query

**Table 5-18** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-19** Request body parameters

| Parameter | Mandatory | Type | Description                              |
|-----------|-----------|------|--|
| vertices  | Yes       | List | Vertex IDs you use to query the vertices |

## Response Parameters

**Table 5-20** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | The <b>data</b> field is contained when the query is successful, and the <b>data</b> field contains the vertices query result.  |
| result       | String | Query results. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .   |

## Example Request

Query nodes in batches by node ID. The vertex IDs to be queried are **27003509\_Station Building** and **39636392\_Badaling Great Wall**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query
{
  "vertices":
  ["27003509_Station Building",
  "39636392_Badaling Great Wall"]
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "27003509_Station Building",
        "label": "tag",
        "properties": {
          "popularity": [
            0
          ],
          "name": [
            "Station Building"
          ],
          "alias": [
            "Guanghua Road Office",
            "Headquarters",
            "Giant Underpants",
            "Headquarters Building"
          ]
        }
      },
      {
        "id": "39636392_Badaling Great Wall",
        "label": "tag",
        "properties": {
          "popularity": [
            0
          ],
          "name": [
            "Badaling Great Wall"
          ],
          "alias": [
            "Great Wall"
          ]
        }
      }
    ]
  }
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

## Status Code

| Return Value     | Description            |
|------------------|------------------------|
| 400 Bad Request  | Request error.         |
| 401 Unauthorized | Authentication failed. |

| Return Value              | Description              |
|---------------------------|--------------------------|
| 403 Forbidden             | No operation permission. |
| 404 Not Found             | No resources found.      |
| 500 Internal Server Error | Internal service error.  |
| 503 Service Unavailable   | Service unavailable.     |

## Error Code

See [Error Code](#).

### 5.1.1.7 Adding Vertices in Batches

#### Function

This API is used to add vertices in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-add

**Table 5-21** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-22** Request body parameters

| Parameter | Mandatory | Type | Description  |
|-----------|-----------|------|--|
| vertices  | Yes       | List | Vertices you want to add. A maximum of 10,000 vertices can be added at a time. For details about this array, see the <a href="#">vertices parameters</a> . |

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| overrideExists | No        | Boolean | Whether to overwrite the existing vertices in the <b>vertices</b> parameter. The default value is <b>false</b> . <ul style="list-style-type: none"> <li>If this parameter is set to <b>false</b>, existing vertices are ignored.</li> <li>If this parameter is set to <b>true</b>, the existing vertices in the <b>vertices</b> parameter are overwritten.</li> </ul> |

**Table 5-23** vertices parameter description

| Parameter  | Mandatory | Type   | Description            |
|------------|-----------|--------|------------------------|
| vertex     | Yes       | String | Vertex ID              |
| label      | Yes       | String | Vertex label           |
| properties | No        | JSON   | Value of each property |

## Response Parameters

**Table 5-24** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> . |

## Example Request

Add vertices in batches. The names of the vertices to be added are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "label": "movie",
      "properties": {
        "movieid": [
          "6"
        ],
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ],
  "overrideExists": true
}
```

### NOTE

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertex **6** already exists in the graph, properties of vertex **6** are overwritten.

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
```

```
"result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] already exists",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.1.8 Deleting Vertices in Batches

#### Function

This API is used to delete vertices in batches based on the vertex IDs.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-delete

**Table 5-25** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |



## Request Parameters

**Table 5-26** Request body parameters

| Parameter | Mandatory | Type   | Description                   |
|-----------|-----------|--------|-------------------------------|
| vertices  | Yes       | String | Vertex ID array to be deleted |

## Response Parameters

**Table 5-27** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Delete nodes in batches by node ID. The vertex IDs to be deleted are **Vivian** and **46**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete
{
  "vertices": [
    "Vivian",
    "46"
  ]
}
```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error.           |
| 401 Unauthorized          | Authentication failed.   |
| 403 Forbidden             | No operation permission. |
| 404 Not Found             | No resources found.      |
| 500 Internal Server Error | Internal service error.  |
| 503 Service Unavailable   | Service unavailable.     |

## Error Code

See [Error Code](#).

### 5.1.1.9 Updating Vertex Properties in Batches

#### Function

This API is used to update vertex properties in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/properties/action?  
action\_id={actionId}

**Table 5-28** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .   |
| graph_name | Yes       | String | Graph name   |
| actionId   | Yes       | String | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>batch-update</b>: Update the value of a property.</li> <li>• <b>batch-add</b>: Add the value to a property. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of <b>batch-update</b>. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>batch-del</b>: Delete a property value.</li> </ul> |

## Request Parameters

**Table 5-29** Request body parameters

| Parameter   | Mandatory | Type    | Description   |
|-------------|-----------|---------|---|
| vertices    | Yes       | Json    | Vertex array to be updated For details about this array, see the <a href="#">vertices parameters</a> .  |
| ignoreError | No        | Boolean | Whether to ignore the update error of specific vertices. The default value is <b>false</b> . <ul style="list-style-type: none"> <li>• The value <b>false</b> indicates that if an error that causes the update failure is detected, for example, the vertex to be updated does not exist, an error is reported and no vertex will be updated.</li> <li>• The value <b>true</b> indicates that similar errors will be ignored and other vertex properties without errors will be updated.</li> </ul> |

**Table 5-30** vertices parameter description

| Parameter  | Mandatory | Type   | Description                          |
|------------|-----------|--------|--------------------------------------|
| vertex     | Yes       | String | Vertex ID                            |
| label      | No        | String | Vertex label                         |
| properties | Yes       | Json   | Value of each property to be updated |

## Response Parameters

**Table 5-31** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update vertex properties in batches. The vertex names to be updated are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?
action_id=batch-update
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
```

```

        "testmoive"
      ],
      "genres": [
        "Comedy"
      ]
    }
  },
  {
    "vertex": "6",
    "properties": {
      "title": [
        "testmoive_exist_id"
      ],
      "genres": [
        "Comedy"
      ]
    }
  }
],
"ignoreError": true
}

```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```

Http Status Code: 200
{
  "result": "success"
}

```

### Status code: 400

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8220"
}

```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.1.10 Adding a Vertex Label

#### Function

This API is used to add a vertex label.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/{vertex\_id}/labels

**Table 5-32** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| vertex_id  | Yes       | String | Vertex ID  |

#### Request Parameters

**Table 5-33** Request body parameter

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--------------|
| label     | Yes       | String | Vertex label |

#### Response Parameters

**Table 5-34** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

Add a vertex label named **user**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels
{
  "label":"user"
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Same label [user] already exists",
  "errorCode": "GES.8213"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.1.11 Deleting a Vertex Label

## Function

This API is used to delete a vertex label.

## URI

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/{vertex\_id}/labels/{label\_name}

**Table 5-35** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| vertex_id  | Yes       | String | Vertex ID  |
| label_name | Yes       | String | Vertex label   |

## Response Parameters

**Table 5-36** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |



| Parameter | Type   | Description  |
|-----------|--------|--|
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> . |

## Example Request

Delete a vertex label.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/46/labels/movie
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Vertex [46] does not have label [movie]",
  "errorCode": "GES.8182"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.1.12 Exporting Filtered Vertices

#### Function

This API is used to export the vertex set that meets the filter criteria.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?action\_id=export

**Table 5-37** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

Export vertices that meet filter criteria (only the asynchronous mode is supported). The export path is **demo\_movie/**, and the export file name is **export\_movie\_and\_user.csv**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export
{
  "labels": [
    "movies",
    "user"
  ],
  "vertexFilters": [
    {
      "propertyName": "Age",
      "predicate": "=",
      "values": [
        "18-24"
      ]
    }
  ],
  "exportPath": "demo_movie/",
  "fileName": "export_movie_and_user.csv",
  "obsParameters": {
    "accessKey": "XXXX",
    "secretKey": "XXXX"
  }
}
```

## Request Parameters

**Table 5-38** Request body parameters

| Parameter     | Mandatory  | Type   | Description  |
|---------------|--|--------|--|
| labels        | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | String | Filter criteria of the vertex type   |
| vertexFilters | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | Json   | Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see <a href="#">vertexFilters parameters</a> . |
| exportPath    | Yes  | String | Export path  |
| fileName      | No   | String | Name of the exported file  |
| obsParameters | Yes  | String | OBS authentication parameters. For details, see <a href="#">obsParameters</a> .  |

**Table 5-39** vertexFilters parameters

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| propertyName | Yes       | String | Property name   |
| predicate    | Yes       | String | Predicate. Available values are =, <, >, <=, >=, <b>range</b> , <b>has</b> , <b>hasNot</b> .                              |
| values       | No        | String | Property value.   |
| type         | No        | String | Logical operator of the filter criteria. Possible values are <b>and</b> and <b>or</b> . The default value is <b>and</b> . |

**Table 5-40** obsParameters parameter description

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | string | AK          |
| secretKey | Yes       | string | SK          |

 NOTE

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

## Response Parameters

**Table 5-41** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of the edge query job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                  |

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.1.13 Deleting Filtered Vertices

#### Function

This API is used to delete the vertex set that meets the filter criteria.

**Table 5-42** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?action\_id=delete

## Request Parameters

**Table 5-43** Request body parameters

| Parameter     | Mandatory  | Type   | Description   |
|---------------|--|--------|---|
| labels        | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | String | Filter criteria of the vertex type  |
| vertexFilters | Either <b>labels</b> or <b>vertexFilters</b> is mandatory. | Object | Filter criteria, in JSONArray format. Vertices are filtered by property.<br>For details, see <a href="#">vertexFilters parameters</a> . |

**Table 5-44** vertexFilters parameters

| Parameter    | Mandatory   | Type   | Description   |
|--------------|---|--------|---|
| propertyName | Yes   | String | Property name   |
| predicate    | Yes   | String | Predicate. Available values are =, <, >, <=, >=, <b>range</b> , <b>has</b> , <b>hasNot</b> .                                |
| values       | No  | String | Property value.   |
| type         | No  | String | Logical operator of the filter criteria. Possible values are <b>and</b> and <b>or</b> . The default value is <b>and</b> .   |
| degree       | Either <b>propertyName</b> or <b>degree</b> is mandatory. | String | Degree of a vertex. The options are <b>in</b> , <b>out</b> , and <b>both</b> . The predicate can be =, <, >, <=, >=, or !=. |

## Response Parameters

**Table 5-45** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId     | String | ID of the vertex query job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |

## Example Request

Delete vertices that meet filter criteria (only the asynchronous mode is supported). The vertex type filter criteria are **movies** and **user**, and the property name is **Age**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete
{
  "labels": [
    "movies",
    "user"
  ],
  "vertexFilters": [
    {
      "propertyName": "Age",
      "predicate": "=",
      "values": [
        "18-24"
      ]
    }
  ]
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.1.2 Edge Operation APIs

### 5.1.2.1 Querying Edges Based on Filter Criteria

#### Function

This API is used to query edges that meet filter criteria.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=query

**Table 5-46** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |



## Request Parameters

**Table 5-47** Request body parameters

| Parameter   | Mandatory  | Type    | Description  |
|-------------|--|---------|--|
| labels      | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String  | Filter criteria of the relationship type                                 |
| edgeFilters | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String  | Filter criteria, in JSONArray format. Vertices are filtered by property. |
| offset      | No   | Integer | Start position of the request  |
| limit       | No   | Integer | Expected number of edges returned by a query                             |
| sorts       | No   | Object  | Result sorting property. It is in JSONArray format.                      |

**Table 5-48** sorts parameter description

| Parameter    | Mandatory  | Type   | Description  |
|--------------|--|--------|--|
| key          | Either <b>Key</b> or <b>propertyName</b> is mandatory. | String | Possible values are <b>label</b> and <b>property</b> , which indicate that labels or properties are sorted.                                      |
| propertyName | Either <b>Key</b> or <b>propertyName</b> is mandatory. | String | Property name  |
| orderValue   | No   | String | Possible values are <b>incr</b> and <b>decr</b> , which indicate ascending and descending order respectively. The default value is <b>incr</b> . |

**Table 5-49** edgeFilters parameter structure

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---------------|
| propertyName | Yes       | String | Property name |

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| predicate | Yes       | String | Logical relationship. Possible values are =, <, >, <=, >=, <b>range</b> , <b>has</b> , <b>hasNot</b><br><b>NOTE</b><br>If the property is of the composite type, such as list or set, the predicate can only be <b>has</b> or <b>hasNot</b> . |
| values    | No        | String | Property value.   |
| type      | No        | String | Logical relationship of filter criteria. Possible values are <b>and</b> and <b>or</b> . The default value is <b>and</b> .   |

## Response Parameters

**Table 5-50** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of the edge query job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                     |

## Example Request

Query edges that meet filter criteria. The start position of the request is **0**, the number of edges to be returned is **20**, and the filter criterion of the relationship type is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query
{
  "offset":"0",
  "limit":"20",
  "labels":[
    "rate"
  ],
}
```

```

"edgeFilters":[
  {
    "propertyName":"Score",
    "predicate":">=",
    "values":[
      "2"
    ]
  },
  {
    "propertyName":"Datetime",
    "predicate":"range",
    "values":[
      "1998-12-27 01:00:00",
      "2000-12-31 00:12:38"
    ],
    "type":"or"
  }
]

```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```

Http Status Code: 200
{
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",
  "jobType": 0
}

```

### Status code: 400

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and edgeFilters cannot all be null",
  "errorCode": "GES.8103"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.2.2 Querying Edge Details

#### Function

This API is used to query the detailed information about an edge based on the source vertex, target vertex, and index of the edge. Information about edges and properties is returned.

#### URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source={sourceVertex}&target={targetVertex}&index={index}
```

**Table 5-51** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-52** Request body parameters

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| project_id   | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name   | Yes       | String | Graph name   |
| sourceVertex | Yes       | String | Source vertex of an edge   |
| targetVertex | Yes       | String | Target vertex of an edge   |

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| index     | No        | Integer | Edge index. If this parameter is not set, all edges between the source and target vertices are queried |

## Response Parameters

**Table 5-53** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. If the query is successful, the query result will be returned. If the query fails, this parameter will be left blank.  |

**Table 5-54** data parameter

| Parameter | Mandatory | Type | Description  |
|-----------|-----------|------|--|
| edges     | Yes       | List | Edge result set. If no edge is found, this parameter will be left blank. |

## Example Request

Query details about an edge.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source=Ray&target=Rocky&index=6
```

 NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example successful response

Http Status Code: 200

```
{
  "data": {
    "edges": [
      {
        "index": "6",
        "source": "Ray",
        "label": "rate",
        "properties": {
          "Score": [
            3
          ],
          "Datetime": [
            "2000-11-22 19:23:05"
          ]
        }
      },
      "target": "Rocky"
    ]
  }
}
```

### Status code: 400

Example response (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8107"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.2.3 Adding an Edge

#### Function

This API is used to add an edge.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges

**Table 5-55** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-56** Request body parameters

| Parameter    | Mandatory | Type                                | Description  |
|--------------|-----------|-------------------------------------|--|
| source       | Yes       | String                              | Source vertex name   |
| target       | Yes       | String                              | Target vertex name   |
| label        | No        | String                              | Label of an edge. If no label exists, set it to <b>__DEFAULT__</b> . |
| properties   | No        | Object                              | Value of each property   |
| parallelEdge | No        | <a href="#">parallelEdge</a> Object | How to process repetitive edges                                      |

**Table 5-57** parallelEdge parameters

| Parameter        | Mandatory | Type                             | Description  |
|------------------|-----------|----------------------------------|--|
| action           | No        | String                           | <p>Processing mode of repetitive edges. The value can be <b>allow</b>, <b>ignore</b>, or <b>override</b>. The default value is <b>allow</b>.</p> <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul>   |
| ignoreLabel      | No        | Boolean                          | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: indicates that the repetitive edge definition contains the label. That is, &lt;Source vertex, Target vertex, Label&gt; indicates an edge.</li> </ul>   |
| targetProperties | No        | <b>targetProperties</b><br>Array | <p>List of properties used to determine repetitive edges. If this parameter is specified, <b>ignoreLabel</b> is set to <b>false</b> to define repetitive edges that contain properties in the list specified by this parameter.</p> <p>(This parameter is mandatory only when <b>action</b> is set to <b>override</b>.)</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• The current version supports only overwriting by property and does not support ignoring by property. In addition, only one property takes effect for each label.</li> <li>• Properties of the non-single type are considered unequal.</li> </ul> |

**Table 5-58** targetProperties parameter description

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| label     | Yes       | String | Labels whose repetitive edges need to be determined by property |



|            |     |       |   |
|------------|-----|-------|---|
| properties | Yes | Array | List of properties whose repetitive edges need to be determined by property. Currently, only one property is supported. If multiple properties are entered, the first property is used. |
|------------|-----|-------|---|

## Response Parameters

**Table 5-59** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |
| data         | Object | Query results. If the query is successful, the query result will be returned. If the query fails, this parameter will be left blank.  |

## Example Request

Add an edge. The source name is **Lily**, the target name is **Rocky**, and the label name of the edge is **rate**.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges
```

```
{
  "source": "Lily",
  "target": "Rocky",
  "label": "rate",
  "properties": {
    "Score": [
      5
    ],
    "Datetime": [
      "2018-01-01 20:30:05"
    ]
  },
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true,
  }
}
```

```

"targetProperties": [
  {
    "label": "rate",
    "properties": [
      "Datetime"
    ]
  },
  {
    "label": "superclassOf",
    "properties": [
      "popularity"
    ]
  }
]
}

```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```

Http Status Code: 200
{
  "result": "success",
  "data": {"index": "0"}
}

```

### Status code: 400

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}

```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

## 5.1.2.4 Deleting an Edge

### Function

This API is used to delete an edge based on the specified property value or index.

### URI

```
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/edges?
source={sourceVertex}&target={targetVertex}&index={index}&label={label}&property={name}&value={value}
```

### Request Parameters

Table 5-60 Request body parameters

| Parameter  | Mandatory | Type    | Description   |
|------------|-----------|---------|---|
| project_id | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name | Yes       | String  | Graph name  |
| source     | Yes       | String  | Source vertex name  |
| target     | Yes       | String  | Target vertex name  |
| index      | No        | Integer | Edge index <ul style="list-style-type: none"> <li>• If <b>property</b> has been set, ignore this parameter.</li> <li>• If <b>property</b> is not set, the edge is deleted based on <b>index</b>.</li> <li>• If neither <b>property</b> nor <b>index</b> is set, all edges between <b>source</b> and <b>target</b> are deleted.</li> </ul> |
| label      | No        | String  | Indicates the label of an edge, which can accelerate the search of property values. This parameter must be used together with <b>property</b> .   |
| property   | No        | String  | Property name of the edge to be deleted. This parameter must be used together with <b>value</b> .   |

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| value     | No        | String | Indicates the property value of the edge to be deleted. This parameter must be used together with <b>property</b> . |

## Example Request

Delete an edge. The source vertex name of the edge to be deleted is **Vivian**, the target vertex name is **Lethal**, the edge index value is **0**, the edge label value is **rate**, the property name is **Score**, and the property value is **5**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges?
source=Vivian&target=Lethal Weapon&index=0&label=rate&property=Score&value=5
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Response Parameters

**Table 5-61** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
```

```
"result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

**Status Code**

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

**Error Code**

See [Error Code](#).

**5.1.2.5 Updating an Edge Property**

**Function**

This API is used to update edge property values. The operations include ADD, UPDATE, and DEL.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/properties/action? action\_id={actionId}&source={sourceVertex}&target={targetVertex}&index={index}

**Table 5-62** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

| Parameter    | Mandatory | Type    | Description   |
|--------------|-----------|---------|---|
| actionId     | Yes       | String  | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>update</b>: Update a property value.</li> <li>• <b>add</b>: Add a property value. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of UPDATE. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>del</b>: Delete a property value.</li> </ul> |
| sourceVertex | Yes       | String  | Source vertex of an edge  |
| targetVertex | Yes       | String  | Target vertex of an edge  |
| index        | No        | Integer | Edge index. If this parameter is not set, properties of the first edge between the vertices will be modified.   |

## Request Parameters

Table 5-63 Request body parameters

| Parameter        | Mandatory | Type   | Description  |
|------------------|-----------|--------|--|
| properties       | Yes       | Object | Value of each property   |
| label            | No        | String | Name of a label  |
| targetProperties | No        | Array  | Properties used to determine duplicate edges. <ul style="list-style-type: none"> <li>• If this parameter is not left blank, other properties of duplicate edges (with the same source vertex and target vertex) that has the same property value as the input property value will be overwritten. If there are multiple specified properties, the properties of the first edge that is matched based on the property input sequence are modified.</li> <li>• If this parameter is left blank or no property is specified for the input edge, the first edge that meets the criteria is updated.</li> </ul> For details about the property elements, see <a href="#">Table 5-64</a> . |

**Table 5-64** targetProperty parameter description

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| label      | Yes       | String | Label name. The label of duplicate edges is determined by the property.   |
| properties | Yes       | Array  | Value of each property. The property list of duplicate edges is determined by the property. Currently, only a single property is supported. If multiple properties are entered, the first property is used. |

## Example Request

Update the property value of an edge. The value of property **Rating** is **7**, the value of property **Datetime** is **2020-12-27 23:44:41**, and the label name is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=update&source=Lily&target=Tom&index=1
{
  "properties": {
    "Rating": ["7"],
    "Datetime":["2020-12-27 23:44:41"]
  },
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ]
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Response Parameters

**Table 5-65** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "edge [Lily-Tom-1] does not exist",
  "errorCode": "GES.8221"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).



### 5.1.2.6 Querying Edges in Batches

#### Function

This API is used to query the detailed information about edges in batches based on the source vertices, target vertices, and indexes of the edges. Information about edges and properties is returned.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-query

**Table 5-66** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-67** Request body parameter

| Parameter | Mandatory | Type   | Description              |
|-----------|-----------|--------|--------------------------|
| edges     | Yes       | Object | Edge array to be queried |

**Table 5-68** edges parameter description

| Parameter | Mandatory | Type   | Description              |
|-----------|-----------|--------|--------------------------|
| source    | Yes       | String | Source vertex of an edge |
| target    | Yes       | String | Target vertex of an edge |
| index     | No        | String | Edge index               |

## Response Parameters

**Table 5-69** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | String | The <b>data</b> field is contained when the query is successful, and the <b>data</b> field contains the edges query result.   |

## Example Request

Query details about edges in batches based on the source vertex, target vertex, and index. The source nodes of the edges to be queried are **39631050\_Landscape** and **27803870\_Landmark building**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query
{
  "edges": [{
    "source": "39631050_Landscape",
    "target": "27803870_Landmark building"
  },{
    "index": "0",
    "source": "27803870_Landmark building",
    "target": "27661363_Jiuhua Hot Spring"
  }
]
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example successful response

```
{
  "data": {
    "edges": [
      {
        "index": "24",
```

```

    "source": "39631050_Landscape",
    "label": "superclassOf",
    "properties": {
      "popularity": [
        0
      ]
    },
    "target": "27803870_Landmark building"
  },
  {
    "index": "0",
    "source": "27803870_Landmark building",
    "label": "superclassOf",
    "properties": {
      "popularity": [
        0
      ]
    },
    "target": "27661363_Jiuhua Hot Spring"
  }
]
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "parameter does not contain 'source'",
  "errorCode": "GES.8000"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.2.7 Adding Edges in Batches**

**Function**

This API is used to add edges in batches.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-add

**Table 5-70** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

| Parameter      | Mandatory | Type                                | Description  |
|----------------|-----------|-------------------------------------|--|
| edges          | Yes       | Object                              | Edge array to be added   |
| parallelEdge   | No        | <a href="#">parallelEdge</a> Object | How to process repetitive edges  |
| createNotExist | No        | Boolean                             | Whether to add source or target vertices that do not exist in the <b>edges</b> parameter before adding edges. The default value is <b>false</b> , which does not affect the original functions and semantics.<br>If this parameter is set to <b>true</b> , source or target vertices that do not exist in the <b>edges</b> parameter are added prior to the edges. |

**Table 5-71** edges parameter description

| Parameter | Mandatory | Type   | Description              |
|-----------|-----------|--------|--------------------------|
| source    | Yes       | String | Source vertex of an edge |
| target    | Yes       | String | Target vertex of an edge |

| Parameter  | Mandatory | Type   | Description            |
|------------|-----------|--------|------------------------|
| label      | Yes       | String | Edge label             |
| properties | No        | Object | Value of each property |

**Table 5-72** parallelEdge

| Parameter   | Mandatory | Type    | Description  |
|-------------|-----------|---------|--|
| action      | No        | String  | <p>Processing mode of repetitive edges. The value can be <b>allow</b>, <b>ignore</b>, or <b>override</b>. The default value is <b>allow</b>.</p> <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul>   |
| ignoreLabel | No        | Boolean | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: indicates that the repetitive edge definition contains the label. That is, &lt;Source vertex, Target vertex, Label&gt; indicates an edge.</li> </ul> |

| Parameter        | Mandatory | Type                             | Description  |
|------------------|-----------|----------------------------------|--|
| targetProperties | No        | <b>targetProperties</b><br>Array | <p>List of properties used to determine repetitive edges. If this parameter is specified, <b>ignoreLabel</b> is set to <b>false</b> to define repetitive edges that contain properties in the list specified by this parameter.</p> <p>(This parameter is mandatory only when <b>action</b> is set to <b>override</b>.)</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>The current version supports only overwriting by property and does not support ignoring by property. In addition, only one property takes effect for each label.</li> <li>Properties of the non-single type are considered unequal.</li> </ul> |

**Table 5-73** targetProperties

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| label      | Yes       | String | Labels whose repetitive edges need to be determined by property   |
| properties | Yes       | Array  | List of properties whose repetitive edges need to be determined by property. Currently, only one property is supported. If multiple properties are entered, the first property is used. |

## Response Parameters

**Table 5-74** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | <p>System prompt.</p> <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Add edges in batches. The source vertex is **46**, the target vertices are **39** and **38**, and the edge label is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
{
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": false,
    "targetProperties": [
      {
        "label": "rate",
        "properties": [
          "Datetime"
        ]
      },
      {
        "label": "superclassOf",
        "properties": [
          "popularity"
        ]
      }
    ]
  },
  "createNotExists": true,
  "edges": [
    {
      "source": "46",
      "target": "39",
      "label": "rate",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-01 20:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "label": "rate",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-01 20:30:05"
        ]
      }
    }
  ]
}
```

```

    ]
  }
}
]
}

```

 **NOTE**

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertices **46** and **39** are not in the original graph, create vertices **46** and **39**, retain the default value of each label, and add an edge.

## Example Response

### Status code: 200

Example response (successful request)

```

Http Status Code: 200
{
  "result": "success",
  "data": {
    "edges": [
      {
        "index": "7",
        "source": "46",
        "target": "39"
      },
      {
        "index": "0",
        "source": "46",
        "target": "38"
      }
    ]
  }
}

```

### Status code: 400

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |



## Error Codes

See [Error Codes](#).

### 5.1.2.8 Deleting Edges in Batches

#### Function

This API is used to delete edges in batches based on the source vertices, target vertices, and indexes of the edges.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-delete

**Table 5-75** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

Delete edges in batches. The source vertices of the edges are **39631050\_Landscape** and **27803870\_Landmark building**, and the target vertices of the edges are **27803870\_Landmark building** and **27661363\_Villa hot spring**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete
{
  "edges": [
    {
      "source": "39631050_Landscape",
      "target": "27803870_Landmark building"
    },
    {
      "index": "0",
      "source": "27803870_Landmark building",
      "target": "27661363_Villa hot spring"
    }
  ],
  "ignoreError": true
}
```

#### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Request Parameters

**Table 5-76** Request body parameters

| Parameter     | Mandatory | Type    | Description  |
|---------------|-----------|---------|--|
| edges         | Yes       | Object  | Edge array to be deleted   |
| executionMode | No        | String  | <b>sync</b> indicates the synchronous mode, and <b>async</b> indicates the asynchronous mode. The default value is <b>sync</b> .   |
| ignoreError   | No        | Boolean | Whether to ignore errors, for example, the edge to delete does not exist. The default value is <b>false</b> , indicating that errors will not be ignored. Errors in JSON format cannot be ignored. |

**Table 5-77** edges parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| source    | Yes       | String | Source vertex of an edge   |
| target    | Yes       | String | Target vertex of an edge   |
| index     | No        | String | Edge index   |
| label     | No        | String | Label of an edge. If the <b>index</b> parameter is set, this parameter is ignored. If the <b>index</b> parameter is not set, an edge that meets the <b>source</b> , <b>target</b> , and <b>label</b> conditions is deleted. If the specified <b>label</b> value does not exist in the schema or the edge with the same <b>label</b> does not exist, no edge will be deleted. |

## Response Parameters

- Synchronous call

**Table 5-78** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter does not contain 'source'",
  "errorCode": "GES.8000"
}
```

- Asynchronous call

**Table 5-79** Response body parameters

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| errorMessage | No        | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| errorCode | No        | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId     | No        | String  | ID of the deletion job.<br>This parameter is left blank if the request fails. This ID can be used as a request parameter to obtain the deletion result through the API for querying the job status.           |
| jobType   | No        | Integer | Task type. This parameter is left blank if the request fails.   |

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",
  "jobType": 3
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [test_117d] is not found",
  "errorCode": "GES.8402"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.2.9 Updating Edge Properties in Batches

#### Function

This API is used to update edge properties in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/properties/action?  
action\_id={actionId}

**Table 5-80** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .   |
| graph_name | Yes       | String | Graph name   |
| actionId   | Yes       | String | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>batch-update</b>: Update the value of a property.</li> <li>• <b>batch-add</b>: Add the value to a property. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of <b>batch-update</b>. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>batch-del</b>: Delete a property value.</li> </ul> |

#### Request Parameters

**Table 5-81** Request body parameters

| Parameter | Mandatory | Type | Description              |
|-----------|-----------|------|--------------------------|
| edges     | Yes       | Json | Edge array to be updated |

| Parameter   | Mandatory | Type    | Description   |
|-------------|-----------|---------|---|
| ignoreError | No        | Boolean | Whether to ignore the update error of specific edges. The default value is <b>false</b> , indicating that an error that causes the update failure must be detected. For example, if the edge to be updated does not exist, an error is reported and no edge is updated.<br><br>If the value is <b>true</b> , similar errors are ignored and other edge properties without errors are updated. |

**Table 5-82** edges parameter description

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| source     | Yes       | String | Source vertex of an edge  |
| target     | Yes       | String | Target vertex of an edge  |
| index      | No        | String | Edge index. If this parameter is not set, the first edge between vertices is updated. |
| properties | Yes       | Object | Value of each property  |

## Response Parameters

**Table 5-83** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update edge properties in batches. The source vertex of the edge is **46**, and the target vertices of the edge are **39** and **38**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=batch-update
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "index": "0",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ],
  "ignoreError": true
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "edge source vertex [46] does not exist",
  "errorCode": "GES.8221"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.2.10 Exporting Filtered Edges

#### Function

This API is used to export the edge set that meets the filter criteria.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=export

#### Request Parameters

**Table 5-84** Request body parameters

| Parameter     | Mandatory  | Type   | Description   |
|---------------|--|--------|---|
| exportPath    | Yes  | String | Export path   |
| fileName      | No   | String | Name of the exported file   |
| obsParameters | Yes  | String | OBS authentication parameters. For details, see <a href="#">Table 5-272</a> . |
| labels        | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String | Filter criteria of the relationship type                                      |



| Parameter   | Mandatory  | Type   | Description  |
|-------------|--|--------|--|
| edgeFilters | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String | Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see <a href="#">Table 5-49</a> . |

## Response Parameters

**Table 5-85** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String  | ID of the edge query job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                  |
| jobType      | Integer | Type of an asynchronous job   |

## Example Request

Export edges that meet filter criteria (only the asynchronous mode is supported). The relationship type filter criterion is **rate**, the property names are **Score** and **Datetime**, and the export path is **demo\_movie/**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export
{
  "labels": [
    "rate"
  ],
  "edgeFilters": [
    {
      "propertyName": "Score",
      "predicate": ">=",
      "values": [
        "2"
      ]
    }
  ]
}
```

```
{
  "propertyName": "Datetime",
  "predicate": "range",
  "values": [
    "1998-12-27 01:00:00",
    "2000-12-31 00:12:38"
  ],
  "type": "or"
},
"exportPath": "demo_movie/",
"fileName": "export_rate.csv",
"obsParameters": {
  "accessKey": "XXXXXXX",
  "secretKey": "XXXXXXX"
}
}
```

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 0
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8103"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.2.11 Deleting Filtered Edges

#### Function

This API is used to delete the edge set that meets the filter criteria.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=delete

#### Request Parameters

**Table 5-86** Request body parameters

| Parameter   | Mandatory  | Type   | Description   |
|-------------|--|--------|---|
| labels      | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String | Filter criteria of the relationship type  |
| edgeFilters | Either <b>labels</b> or <b>edgeFilters</b> is mandatory. | String | Filter criteria, in JSONArray format. Vertices are filtered by property.<br>For details, see <a href="#">Table 5-49</a> . |

#### Response Parameters

**Table 5-87** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| job_id       | String | ID of the edge query job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                  |

| Parameter | Type    | Description                 |
|-----------|---------|-----------------------------|
| jobType   | Integer | Type of an asynchronous job |

## Example Request

Delete edges that meet filter criteria (only the asynchronous mode is supported). The relationship type filter criterion is **rate**, and the property names are **Score** and **Datetime**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete
{
  "labels": [
    "rate"
  ],
  "edgeFilters": [
    {
      "propertyName": "Score",
      "predicate": ">=",
      "values": [
        "2"
      ]
    },
    {
      "propertyName": "Datetime",
      "predicate": "range",
      "values": [
        "1998-12-27 01:00:00",
        "2000-12-31 00:12:38"
      ]
    }
  ],
  "type": "or"
}
```

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete
{
  "labels": [
    "rate"
  ],
  "edgeFilters": [
    {
      "propertyName": "Score",
      "predicate": ">=",
      "values": [
        "2"
      ]
    },
    {
      "propertyName": "Datetime",
      "predicate": "range",
      "values": [
        "1998-12-27 01:00:00",
        "2000-12-31 00:12:38"
      ]
    }
  ],
  "type": "or"
}
```

## Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",
  "jobType": 0
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and edgeFilters cannot all be null",
  "errorCode": "GES.8103"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

## 5.1.3 Metadata Operation APIs

### 5.1.3.1 Adding a Label

#### Function

This API is used to add a label.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/labels

**Table 5-88** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

- Request parameters (OBS scenario)

**Table 5-89** Request body parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| name       | Yes       | String | Name of a label<br>A label name can contain a maximum of 256 characters.<br>Only letters, digits, spaces, and special characters %, @, #, \$, :, ;, *, ., +, - are allowed. |
| properties | Yes       | Object | Properties you want to add to the label. For details about the parameters, see <a href="#">Table 5-90</a> .   |

**Table 5-90** properties parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| property  | No        | Object | Label properties. For details about the parameters, see <a href="#">Table 5-91</a> . |

**Table 5-91 property parameters**

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| name       | Yes       | String | Property name<br>1. A property name can contain a maximum of 256 characters.<br>2. A property name cannot contain <, >, &, ASCII 14,15 or 30.<br>3. The property under a label must be unique.                                    |
| type       | No        | String | Label type, indicating that the label is used for vertices or edges. The value can be <b>vertex</b> , <b>edge</b> , or <b>all</b> .<br>The default value is <b>all</b> , indicating that the label applies to vertices and edges. |
| properties | Yes       | Object | Properties you want to add to the label. For details about the parameters, see <a href="#">Table 5-90</a> .   |

**Table 5-92 properties elements**

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| property  | No        | Object | Label properties. For details about the parameters, see <a href="#">Table 5-93</a> . |

**Table 5-93 property parameter description**

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| name      | Yes       | String | Property name<br>1. A property name can contain a maximum of 256 characters.<br>2. A property name cannot contain <, >, &, ASCII 14,15 or 30.<br>3. The property under a label must be unique. |

| Parameter     | Mandatory   | Type   | Description  |
|---------------|---|--------|--|
| cardinality   | Yes   | String | Cardinality type of a property. Possible values: <ul style="list-style-type: none"> <li>• single</li> <li>• list</li> <li>• set</li> </ul>   |
| dataType      | Yes   | String | Data type of a property. For details, see the metadata types in <a href="#">Table 4-122</a> .  |
| typeNameCount | No<br>(This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Total number of parameters of the <b>enum</b> type. This parameter controls the <b>typeName</b> quantity.  |
| typeName*     | No<br>(This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Names of parameters of the <b>enum</b> type. For example, if the value of <b>typeNameCount</b> is 2, the parameter contains <b>typeName1:science</b> and <b>typeName2:literature</b> . |

## Response Parameters

**Table 5-94** Parameter description

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |



| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Add a label. The label name is **book**. The label has three properties to add.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
{
  "name": "book",
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Category",
        "typeName1": "science",
        "typeName2": "literature",
        "typeNameCount": "2",
        "cardinality": "single",
        "dataType": "enum"
      }
    }
  ]
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
```

```
"result": "success"
}
```

**Status code: 400**

Example response (failed request)

Http Status Code: 400

```
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.3.2 Updating a Label

#### Function

In the current version, this API only allows appending properties to existing labels, but not deleting existing properties or updating the order of existing properties. It can either append properties to existing labels or overwrite the entire label.

#### URI

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties
```

**Table 5-95** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

| Parameter  | Mandatory | Type   | Description |
|------------|-----------|--------|-------------|
| label_name | Yes       | String | Label name  |

## Request Parameters

**Table 5-96** Request body parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| properties | Yes       | Object | Property array to be appended. For details about the parameters, see <a href="#">Table 5-97</a> .  |
| override   | No        | Bool   | The default value is <b>false</b> . If the value is <b>true</b> , the entire label is overwritten. |

**Table 5-97** properties parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| property  | No        | Object | Label properties. For details about the parameters, see <a href="#">Table 5-98</a> . |

**Table 5-98** property parameter description

| Parameter   | Mandatory | Type   | Description   |
|-------------|-----------|--------|---|
| name        | Yes       | String | Property name<br><ol style="list-style-type: none"> <li>1. A property name can contain a maximum of 256 characters.</li> <li>2. A property name cannot contain &lt;, &gt;, &amp;, ASCII 14,15 and 30.</li> <li>3. The property under a label must be unique.</li> </ol> |
| cardinality | Yes       | String | Composite type of a property. Possible values: <ul style="list-style-type: none"> <li>• single</li> <li>• list</li> <li>• set</li> </ul>  |

| Parameter     | Mandatory  | Type   | Description  |
|---------------|--|--------|--|
| dataType      | Yes  | String | Data type of a property. For details, see the metadata types in <a href="#">Table 4-122</a> .  |
| typeNameCount | No (This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Total number of parameters of the <b>enum</b> type. This parameter controls the <b>typeName</b> quantity.  |
| typeName*     | No (This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Names of parameters of the <b>enum</b> type. For example, if the value of <b>typeNameCount</b> is <b>2</b> , the parameter contains <b>typeName1:science</b> and <b>typeName2:literature</b> . |

## Response Parameters

**Table 5-99** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update a label. The label name is **book**. The label has three properties to update.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties
{
  "name": "book",
  "override": "true",
  "properties": [
    {
      "property": {
        "name": "Title",
```

```

        "cardinality": "single",
        "dataType": "string"
    }
  },
  {
    "property": {
      "name": "Version",
      "cardinality": "single",
      "dataType": "string"
    }
  },
  {
    "property": {
      "name": "Category",
      "typeName1": "science",
      "typeName2": "literature",
      "typeNameCount": "2",
      "cardinality": "single",
      "dataType": "enum"
    }
  }
]
}

```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```

Http Status Code: 200
{
  "result": "success"
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.3.3 Querying Graph Metadata Details

#### Function

This API is used to query graph metadata details.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema

**Table 5-100** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Response Parameters

**Table 5-101** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. This parameter is left blank when the request fails.   |

**Table 5-102** data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| schema    | List | Definitions of each label and associated property field |

## Example Request

Query metadata details of a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "schema": [
      {
        "label": "rate",
        "type": "all",
        "properties": [
          {
            "name": "Rating",
            "type": "int",
            "cardinality": "single"
          },
          {
            "name": "Datetime",
            "type": "string",
            "cardinality": "single"
          }
        ]
      },
      {
        "label": "user",
        "type": "all",
        "properties": [
          {
            "name": "userid",
            "type": "int",
            "cardinality": "single"
          },
          {
            "name": "gender",
            "probableValue": [
              "F",
              "M"
            ],
            "type": "enum",
            "cardinality": "single"
          },
          {
            "name": "age",
```

```

    "probableValue": [
      "Under 18",
      "18-24",
      "25-34",
      "35-44",
      "45-49",
      "50-55",
      "56+"
    ],
    "type": "enum",
    "cardinality": "single"
  },
  {
    "name": "occupation",
    "probableValue": [
      "other or not specified",
      "academic/educator",
      "artist",
      "clerical/admin",
      "college/grad student",
      "customer service",
      "doctor/health care",
      "executive/managerial",
      "farmer",
      "homemaker",
      "K-12 student",
      "lawyer",
      "programmer",
      "retired",
      "sales/marketing",
      "scientist",
      "self-employed",
      "technician/engineer",
      "tradesman/craftsman",
      "unemployed",
      "writer"
    ],
    "type": "enum",
    "cardinality": "single"
  },
  {
    "name": "Zip-code",
    "maxDataSize": 12,
    "type": "char array",
    "cardinality": "single"
  }
]
},
{
  "label": "book",
  "type": "vertex",
  "properties": [
    {
      "name": "Title",
      "type": "string",
      "cardinality": "single"
    },
    {
      "name": "Version",
      "type": "string",
      "cardinality": "single"
    },
    {
      "name": "Category",
      "probableValue": [
        "science",
        "literature"
      ],
      "type": "enum",

```



```
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "default",
    "type": "all"
  },
  {
    "label": "movie",
    "type": "all",
    "properties": [
      {
        "name": "movieid",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "title",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "genres",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "__DEFAULT__",
    "type": "all"
  }
]
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8003"
}
```

### Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.3.4 Changing Property Names in Batches

#### Function

This API is used to change property names in batches.

#### URI

PUT /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/labels/properties

**Table 5-103** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-104** Request body parameters

| Parameter           | Mandatory | Type      | Description            |
|---------------------|-----------|-----------|------------------------|
| labels              | Yes       | JsonArray | Label array            |
| label               | Yes       | String    | Name of a label        |
| originPropertyName  | Yes       | String    | Original property name |
| updatedPropertyName | Yes       | String    | New property name      |

## Response Parameters

**Table 5-105** Response body parameters

| Parameter     | Type    | Description   |
|---------------|---------|---|
| errorMessage  | String  | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode     | String  | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| updated_count | Integer | Number of properties that are successfully updated  |

## Example Request

Change label property names in batches. The label name is **movie**, the original property name is **title**, and the new property name is **movie\_title**. The label name is **movie**, the original property name is **newProperty**, and the new property name is **xxxxProperty**. The label name is **user**, the original property name is **gender**, and the new property name is **sexuality**.

```
PUT http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/properties
{
  "labels": [{
    "label": "movie",
    "originPropertyName": "title",
    "updatedPropertyName": "movie_title"
  },
  {
    "label": "movie",
    "originPropertyName": "newProperty",
    "updatedPropertyName": "xxxxProperty"
  },
  {
    "label": "user",
    "originPropertyName": "gender",
    "updatedPropertyName": "sexuality"
  }
  ]
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "updated_count": 4
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label name does not exist",
  "errorCode": "GES.8807"
}
```

### Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

### Error Code

See [Error Code](#).

### 5.1.3.5 Deleting a Label

#### Function

This API is used to delete a label as well as the vertices and edges associated with the label.

#### URI

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/labels/{label\_name}

**Table 5-106** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

| Parameter  | Mandatory | Type   | Description     |
|------------|-----------|--------|-----------------|
| graph_name | Yes       | String | Graph name      |
| label_name | Yes       | String | Name of a label |

## Response Parameters

**Table 5-107** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. This parameter is left blank when the request fails.   |

**Table 5-108** data parameter description

| Parameter | Type    | Description  |
|-----------|---------|--|
| outputs   | Integer | Number of deleted vertices or edges when a label is deleted. |

## Example Request

Delete a label as well as the vertices and edges associated with the label.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "72506da3-d2e2-4709-8ca3-a5376a42db23001234245",
  "jobType": 3
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8003"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.3.6 Adding Labels in Batches

#### Function

This API is used to add labels in batches.

#### URI

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add
```

**Table 5-109** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-110** Request body parameters

| Parameter | Mandatory | Type   | Description     |
|-----------|-----------|--------|-----------------|
| labels    | Yes       | String | Metadata labels |

**Table 5-111** label parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| name       | Yes       | String | Name of a label. The value can contain a maximum of 256 characters, including numbers, spaces, and the following special characters: % @ # \$: ? * . + - _. |
| properties | Yes       | Object | Property array to be added. The array element is property. For details about the parameters, see <a href="#">Table 5-112</a> .                              |

**Table 5-112** property parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| name      | Yes       | String | Property name <ol style="list-style-type: none"> <li>1. A property name can contain a maximum of 256 characters.</li> <li>2. A property name cannot contain &lt;, &gt;, &amp;, ASCII 14, 15 or 30.</li> <li>3. The property under a label must be unique.</li> </ol> |

| Parameter     | Mandatory  | Type   | Description  |
|---------------|--|--------|--|
| cardinality   | Yes  | String | Cardinality type of a property. Possible values: <ul style="list-style-type: none"> <li>• single</li> <li>• list</li> <li>• set</li> </ul>   |
| dataType      | Yes  | String | Data type of a property. For details, see the metadata types in <a href="#">Table 8-103</a> .  |
| typeNameCount | No (This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Total number of parameters of the <b>enum</b> type. This parameter controls the <b>typeName</b> quantity.  |
| typeName*     | No (This parameter is mandatory if <b>dataType</b> is <b>enum</b> .) | String | Names of parameters of the <b>enum</b> type. For example, if the value of <b>typeNameCount</b> is <b>2</b> , the parameter contains <b>typeName1:science</b> and <b>typeName2:literature</b> . |

## Response Parameters

**Table 5-113** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |



| Parameter | Type   | Description   |
|-----------|--------|---|
| data      | Object | If some labels fail to be added, this field contains the names of the failed labels and the failure causes. |

## Example Request

Add metadata labels and their properties in batches. The names of the metadata labels are **book** and **movie**.

POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add`

```
{
  "labels": [
    {
      "name": "book",
      "type": "vertex",
      "properties": [
        {
          "property": {
            "name": "title",
            "cardinality": "single",
            "dataType": "string"
          }
        }
      ]
    },
    {
      "name": "movie",
      "type": "vertex",
      "properties": [
        {
          "property": {
            "name": "movieid",
            "cardinality": "single",
            "dataType": "int"
          }
        }
      ]
    }
  ]
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

Example response for a partially successful request

```
Http Status Code: 200
{
```

```

"result": "partial success",
"data": {
  "failed": [
    {
      "cause": "label name is invalid which can only contain letters, digits, space,%,@,#,$,;,?*,.,+,- and",
      "labelName": "book<"
    }
  ]
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}

```

**Status Code**

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

**Error Code**

See [Error Code](#).

**5.1.3.7 Querying Schema Structure**

**Function**

This API is used to query the structure of the generated schema (obtained from OBS).

**URI**

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/structure?detail={details}

**Table 5-114** URI parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name | Yes       | String | Graph name  |
| details    | No        | String | Detailed information of the schema structure. The value can be <b>SIMPLE</b> or <b>FULL</b> . If you set this parameter to <b>SIMPLE</b> , only the labels of vertices or edges are returned. If you set it to <b>FULL</b> , the number of vertices or edges is returned in addition to the labels. If this parameter is left empty, the default value <b>SIMPLE</b> is used. |

## Response Parameters

**Table 5-115** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| schema       | Object | Schema structure. For details about the parameters, see <a href="#">Table 5-116</a> .   |

**Table 5-116** schema parameter description

| Parameter | Type   | Description  |
|-----------|--------|--|
| vertices  | String | Vertex result set. If the graph is empty, the return value is empty. For details about the parameters, see <a href="#">Table 5-117</a> . |

| Parameter | Type   | Description  |
|-----------|--------|--|
| edges     | String | Edge result set. If the graph is empty, the return value is empty. For details about the parameters, see <a href="#">Table 5-118</a> . |

**Table 5-117** vertices parameter description

| Parameter | Type   | Description                            |
|-----------|--------|--|
| vertex    | String | Label name                             |
| weight    | String | Number of vertices that have the label |

**Table 5-118** edges parameter description

| Parameter | Type   | Description                         |
|-----------|--------|-------------------------------------|
| source    | String | Label the start vertex.             |
| target    | String | Label the end vertex.               |
| relation  | String | Relationship label                  |
| weight    | String | Number of edges that have the label |

## Example Request

Query a generated schema (obtained from OBS).

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure?detail=SIMPLE
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "schema": {
    "vertices": [
      {
        "vertex": "user",
        "weight": 100
      },
      {
        "vertex": "movie",
        "weight": 46
      }
    ],
    "edges": [
      {
        "weight": 1209,
        "source": "user",
        "target": "movie",

```

```

        "relation": "rate"
      },
      {
        "weight": 450,
        "source": "user",
        "target": "user",
        "relation": "default"
      }
    ]
  }
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter [detail] cannot be null.",
  "errorCode": "GES.8813"
}

```

**Status Code**

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error.           |
| 401 Unauthorized          | Authentication failed.   |
| 403 Forbidden             | No operation permission. |
| 404 Not Found             | No resources found.      |
| 500 Internal Server Error | Internal service error.  |
| 503 Service Unavailable   | Service unavailable.     |

**Error Code**

See [Error Code](#).

**5.1.3.8 Generating a Schema Structure**

**Function**

This API is used to generate a schema structure where labels are represented with vertices and the relationship between the labels are represented with edges, and store the schema structure in an OBS bucket.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/structure/build

**Table 5-119** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Response Parameters

**Table 5-120** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String  | ID of the vertex query job. This parameter is left blank when the request fails.  |
| jobType      | Integer | Type of an asynchronous job   |

## Example Request

Generate a schema where labels are represented with vertices and the relationship between the labels are represented with edges.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "jobId": "2e0c08e1-3fbb-4b33-8776-4809176068d7154236181",
  "jobType": 1
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request ",
  "errorCode": "GES.8813"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.3.9 Generating Data Assets**

**Function**

This API is used to generate data assets.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/data-assets

**Table 5-121** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

None

## Response Parameters

**Table 5-122** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of the vertex query job. This parameter is left blank if the request fails.   |

## Example Request

Generate data assets.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
{}
```

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorCode": "GES.8818",
  "errorMessage": "Data assets information is being generated"
}
```



## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.1.3.10 Obtaining Data Assets

#### Function

This API is used to obtain data assets.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/data-assets

**Table 5-123** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

## Response Parameters

**Table 5-124** Response body parameters

| Parameter          | Type    | Description  |
|--------------------|---------|--|
| generating         | Boolean | Whether data assets are being generated  |
| progress           | String  | Progress of generating data assets   |
| last_generate_time | String  | Last time when a data asset is generated   |
| data_assets        | Object  | Number of vertices and edges under different labels  |
| errorCode          | String  | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| errorMessage       | String  | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |

**Table 5-125** data\_assets parameter description

| Parameter | Type   | Description                               |
|-----------|--------|---|
| vertex    | Object | Number of vertices under different labels |
| edge      | Object | Number of edges under different labels    |

### Example Request

Obtain data assets.

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
```

### Example Response

**Status code: 200**

There are several scenarios for example response for successful requests, including:

- A data asset has never been generated.

```
Http Status Code: 200
{ }
```

- A data asset has never been generated and is being generated.

```
Http Status Code: 200
{
  "progress": "10.05%",
  "generating": true
}
```

- A data asset has been generated and a latest one is being generated.

```
Http Status Code: 200
{
  "last_generate_time": "2022-1-3 12:34:12",
  "data_assets": {
    "vertex": {
      "label1": 3,
      "label2": 14
    },
    "edge": {
      "label3": 123,
      "label4": 435
    }
  },
  "progress": "10.05%",
  "generating": true
}
```

#### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorCode": "GES.8818",
  "errorMessage": "graph [demo] is not found"
}
```

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 5.1.4 Index Operation APIs

### 5.1.4.1 Creating an Index

#### Function

This API is used to create indexes based on the specified information such as `indexName` and `IndexType`. Currently, composite indexes and full-text indexes are supported.

- Composite indexes include global vertex indexes (`GlobalCompositeVertexIndex`) and global edge indexes (`GlobalCompositeEdgeIndex`). Composite indexes can be used to create indexes on labels and properties. Indexes can accelerate the query speed.
- Full-text indexes (`FullTextIndex`) can implement functions such as full-text search and fuzzy search. If you search data immediately after an update, you may get the old data. You are advised to query the data 1 second after the update. For details about how to use full-text indexes, see [Querying Vertices Based on Filter Criteria](#) and [Querying Edges Based on Filter Criteria](#). You can also use full-text indexes in Cypher statements. For details, see [Cypher Operation APIs](#).

 **NOTE**

- Currently, the full-text indexing feature is supported only in the ECS/BMS+MRS deployment mode.
- After an index is created, wait for 30 seconds for index synchronization. After the synchronization is complete, Cypher queries can be accelerated using the index.

## Index Feature

| Feature           | Fuzzy Search | Speed | Flexibility                        |
|-------------------|--------------|-------|------------------------------------|
| Composite indexes | No           | Fast  | Fixed composite property keys only |

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices

**Table 5-126** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Example Request

Create a composite index. The index name is **ageIndex** and the index type is global vertex index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "ageIndex",
  "indexType": "GlobalCompositeVertexIndex",

  "hasLabel": "true",
  "indexProperty": ["age"]
}
```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Request Parameters

**Table 5-127** Request body parameters

| Parameter     | Mandatory   | Type    | Description  |
|---------------|---|---------|--|
| indexName     | Yes   | String  | Index name. Only letters, digits, hyphens (-), and underscores (_) are allowed. Other characters are not allowed. The index name can contain a maximum of 63 characters. |
| indexType     | Yes   | String  | Index type. The value is case-sensitive. <b>GlobalCompositeVertexIndex</b> is a global composite vertex index.   |
| hasLabel      | No  | Boolean | Whether labels exist. The default value is <b>false</b> . <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>                                      |
| indexProperty | No (If <b>hasLabel</b> is <b>false</b> or <b>null</b> , this parameter is mandatory.) | String  | Index property list.<br>The property types that can be used to create indexes include integer, float, double, long, enum, char array, string, and date.                  |

 **NOTE**

If a property is of the string or char array type, the value must be no more than 40 bytes. The excess part will be deleted.

## Response Parameters

**Table 5-128** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type    | Description   |
|-----------|---------|---|
| errorCode | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>                               |
| jobId     | String  | ID of an asynchronous job<br><b>NOTE</b> <ul style="list-style-type: none"> <li>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a>.</li> </ul> |
| jobType   | Integer | Type of an asynchronous job   |
| result    | String  | If the execution is successful, the value of <b>result</b> is <b>success</b> .  |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 8
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8603"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.4.2 Deleting an Index

#### Function

This API is used to delete an index based on the specified indexName.

#### URI

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices/{index\_name}

**Table 5-129** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| index_name | Yes       | String | Index name   |

#### Response Parameters

**Table 5-130** Parameter description

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                 |

| Parameter | Type    | Description                 |
|-----------|---------|-----------------------------|
| jobType   | Integer | Type of an asynchronous job |

## Example Request

Delete an index by name. The index name is **ageIndex**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/ageIndex
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232",
  "jobType": 9
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8604 "
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).



### 5.1.4.3 Querying Indexes

#### Function

This API is used to query all indexes created on a graph.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices

**Table 5-131** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Response Parameters

**Table 5-132** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Index data you want to query  |
| result       | String | Query results. If the query is successful, <b>success</b> is displayed.   |
| indices      | List   | Indexes of the query results  |
| indexType    | String | Index types of the query results  |
| indexName    | String | Index names of the query results  |

| Parameter     | Type | Description                           |
|---------------|------|---------------------------------------|
| indexProperty | List | Index properties of the query results |

## Example Request

Query all indexes created on a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "result": "success",
    "indices": [
      {
        "indexType": "GlobalCompositeVertexIndex",
        "indexName": "ageIdx",
        "indexProperty": [
          "age"
        ],
        "hasLabel": "true"
      }
    ]
  }
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8605"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |

| Return Value              | Description            |
|---------------------------|------------------------|
| 404 Not Found             | No resources found.    |
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

## 5.1.5 Gremlin Operation APIs

### 5.1.5.1 Executing a Gremlin Query

#### Function

This API is used to return the query result of a Gremlin statement.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-gremlin-query

**Table 5-133** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-134** Request body parameter

| Parameter | Mandatory | Type   | Description                      |
|-----------|-----------|--------|----------------------------------|
| command   | Yes       | String | Query command (Gremlin language) |

## Response Parameters

**Table 5-135** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. This parameter is left blank when the request fails.   |

## Example Request

Perform the Gremlin query operation. If the query command is **g.V().limit(100)**, all vertices are queried, but the number of returned vertices is limited to 100.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query
{
  "command":"g.V().limit(100)"
}
```

 **NOTE**

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- The size of the request body cannot exceed 64 MB.

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "runtime": 0.775425022,
    "vertices": [
      {
        "id": "Vivian",
        "label": "user",
        "properties": {
          "Occupation": [
            "artist"
          ]
        }
      }
    ]
  }
}
```

```

    "Name": [
      "Vivian"
    ],
    "Zip-code": [
      "98133"
    ],
    "Gender": [
      "F"
    ],
    "Age": [
      "25-34"
    ]
  }
},
.....
]
}
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "org.apache.tinkerpop.gremlin.driver.exception.ResponseException: No such property: g1
for class: Script4",
  "errorCode": "GES.8503"
}

```

**Status Code**

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

**Error Code**

See [Error Code](#).

**5.1.6 Algorithm APIs**

**5.1.6.1 Executing an Algorithm**

**Function**

This API is used to run specified algorithms based on entered parameters.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-algorithm

**Table 5-136** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Example Request

Execute a specified algorithm. The algorithm name is **pagerank**, the weight coefficient is **0.85**, the convergence precision is **0.00001**, the maximum number of iterations is **1000**, and traversal is performed along edge directions.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName":"pagerank",
  "parameters":{
    "alpha":0.85,
    "convergence":0.00001,
    "max_iterations":1000,
    "directed":true
  }
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

## Response Parameters

**Table 5-137** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |
| jobType      | Integer | Job type. This parameter is left blank when the request fails.  |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 4200

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}
```

## Status Codes

| Return Value    | Description    |
|-----------------|----------------|
| 400 Bad Request | Request error. |

| Return Value              | Description               |
|---------------------------|---------------------------|
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.6.2 Algorithm API Parameter References

#### 5.1.6.2.1 Common Algorithm Parameters

#### Request Example

```
{  
  "algorithmName": "XXX",  
  "parameters": {  
    ...  
  }  
}
```



## Request Parameters

**Table 5-138** Request body parameters

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| algorithmName | Yes       | String | <p>Algorithm name.</p> <p>Available values are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• pagerank</li> <li>• personalrank</li> <li>• kcore</li> <li>• k_hop</li> <li>• shortest_path</li> <li>• all_shortest_paths</li> <li>• filtered_shortest_path</li> <li>• sssp</li> <li>• shortest_path_of_vertex_sets</li> <li>• n_paths</li> <li>• closeness</li> <li>• label_propagation</li> <li>• louvain</li> <li>• link_prediction</li> <li>• node2vec</li> <li>• realtime_recommendation</li> <li>• common_neighbors</li> <li>• connected_component</li> <li>• degree_correlation</li> <li>• triangle_count</li> <li>• cluster_coefficient</li> <li>• common_neighbors_of_vertex_sets</li> <li>• all_shortest_paths_of_vertex_sets</li> <li>• filtered_circle_detection</li> <li>• filtered_all_pairs_shortest_paths</li> <li>• filtered_all_shortest_paths</li> <li>• filtered_n_paths</li> </ul> |
| parameters    | Yes       | Object | Algorithm parameters. For details, see the parameter description of each algorithm.  |

**Table 5-139** New Body parameters of version 2.1.7

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| executionMode | No        | String  | <ul style="list-style-type: none"> <li>• <b>sync</b>: synchronous</li> <li>• <b>async</b>: asynchronous</li> </ul> <p>The default value is <b>async</b>.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• k_hop</li> <li>• shortest_path</li> <li>• all_shortest_paths</li> <li>• filtered_shortest_path</li> <li>• shortest_path_of_vertex_sets</li> <li>• n_paths</li> <li>• realtime_recommendation</li> </ul>  |
| offset        | No        | Integer | <p>Synchronization result offset. The default value is <b>0</b>.</p> <p><b>NOTE</b><br/>This parameter is available when <b>executionMode</b> is <b>sync</b>.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• k_hop</li> <li>• shortest_path</li> <li>• all_shortest_paths</li> <li>• shortest_path_of_vertex_sets</li> <li>• n_paths</li> <li>• realtime_recommendation</li> <li>• filtered_all_pairs_shortest_paths</li> <li>• filtered_all_shortest_paths</li> </ul> |

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| limit     | No        | Integer | <p>Maximum number of returned synchronization results. The maximum value is <b>100000</b>. The default value is <b>100000</b>.</p> <p><b>NOTE</b><br/>This parameter is available when <b>executionMode</b> is <b>sync</b>.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• k_hop</li> <li>• shortest_path</li> <li>• all_shortest_paths</li> <li>• shortest_path_of_vertex_sets</li> <li>• n_paths</li> <li>• realtime_recommendation</li> <li>• filtered_all_pairs_shortest_paths</li> <li>• filtered_all_shortest_paths</li> </ul> |

**Table 5-140** New Body parameters of version 2.2.4

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| vertex_filter | No        | Object | <p>Filter criteria for the vertices on a path. Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• filtered_shortest_path</li> <li>• filtered_all_pairs_shortest_paths</li> <li>• filtered_all_shortest_paths</li> </ul> <p>For details about the format, see <a href="#">Table 5-304</a> in "Filtered-query API".</p>              |
| edge_filter   | No        | Object | <p>Filter criteria for the edges (relationships) on a path. Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• filtered_shortest_path</li> <li>• filtered_all_pairs_shortest_paths</li> <li>• filtered_all_shortest_paths</li> </ul> <p>For details about the format, see <a href="#">Table 5-304</a> in "Filtered-query API".</p> |

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| filters   | No        | Object | Filter criteria. Each element in the array corresponds to a filter. This parameter applies only to filtered circle detection. For details about the format, see <a href="#">filters element formats</a> .<br>Supported algorithms: <ul style="list-style-type: none"> <li>filtered_n_paths</li> </ul> |

## Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the `job_id` returned by the algorithm to obtain the algorithm execution result.

### Status code: 200

Example response (successful request)

```
{
  "data": {
    "outputs": {
      $response_data //Result of each algorithm. The results vary with the algorithm.
      "runtime": 1.365867,
      "data_return_size": 3,
      "data_offset": 0,
      "data_total_size": 100
    }
  },
  "status": "complete"
}
```

### NOTE

`response_data` indicates the result of each algorithm. The results vary with algorithms.

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Running algorithm [XXXX] error: YYYYYYYYYY!",
  "errorCode": "GES.8301"
}
```

## Response Parameters

**Table 5-141** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.                           |
| errorCode    | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.                              |
| status       | String | Returned job status for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . This parameter is left blank when the query fails. |
| data         | Object | Algorithm execution result. This parameter is left blank when the query fails.   |

### 5.1.6.2.2 PageRank

**Table 5-142** parameters parameter description

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| alpha          | No        | Double  | Weight coefficient (also called damping coefficient). The value range is (0,1), and the default value is <b>0.85</b> .  |
| convergence    | No        | Double  | Convergence The value range is (0,1), and the default value is <b>0.00001</b> .   |
| max_iterations | No        | Integer | Maximum iterations An integer ranging from 1 to 2147483647. For frontend calls, the range is [1,2000]. The default value is <b>1000</b> .   |
| num_thread     | No        | Integer | Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to <b>1</b> . If the value is greater than 40, it is automatically set to <b>40</b> . The default value is <b>4</b> . |

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| directed  | No        | Boolean | Whether to consider the edge direction. The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> . |

 **NOTE**

Iterations and convergence

The algorithm is terminated when either the maximum number of iterations is reached or the convergence precision is met.

1. Generally, a smaller convergence precision and larger number of iterations lead to a better effect of the algorithm.
2. To meet a certain convergence precision, you should set the number of iterations as large as possible.
3. A larger number of iterations means a longer algorithm running time. To ensure that the algorithm runs at a certain number of iterations (that is, in a fixed duration), you should set the convergence precision as small as possible.

**Table 5-143** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| pagerank  | List | PageRank value of each vertex. The format is as follows:<br>[ <b>{vertexId:rankValue}</b> ,...],<br>where<br><b>vertexId</b> is of the string type.<br><b>rankValue</b> is of the double type. |

### 5.1.6.2.3 PersonalRank

**Table 5-144** parameters parameter description

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| source      | Yes       | String | Node ID  |
| alpha       | No        | Double | Weight coefficient (also called damping coefficient). The value range is (0,1), and the default value is <b>0.85</b> . |
| convergence | No        | Double | Convergence<br>The value range is between 0 and 1. The default value is <b>0.00001</b> .                               |

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| max_iterations | No        | Integer | Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1,2000]. The default value is <b>1000</b> .  |
| num_thread     | No        | Integer | Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to <b>1</b> . If the value is greater than 40, it is automatically set to <b>40</b> . The default value is <b>4</b> . |
| directed       | No        | Boolean | Whether to consider the edge direction. The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> .   |

 NOTE

For details about algorithm iterations and convergence, see [Iterations and Convergence of PageRank](#).

**Table 5-145** response\_data parameter description

| Parameter    | Type   | Description  |
|--------------|--------|--|
| source       | String | -  |
| personalrank | List   | PersonalRank value of each vertex. The format is as follows:<br>[ <b>{vertexId:rankValue},...</b> ],<br>where<br><b>vertexId</b> is of the string type.<br><b>rankValue</b> is of the double type. |

### 5.1.6.2.4 K-core

**Table 5-146** parameters parameter description

| Parameter | Mandatory | Description  | Type    | Value Range                | Default Value |
|-----------|-----------|--|---------|----------------------------|---------------|
| k         | Yes       | Number of cores<br>The algorithm returns vertices whose number of cores is greater than or equal to k. | Integer | Greater than or equal to 0 | -             |

**Table 5-147** response\_data parameter description

| Parameter | Type                      | Description   |
|-----------|---------------------------|---|
| coreness  | List<Map<String,Integer>> | Coreness value ( $coreness \geq k$ ) of each vertex. The format is as follows:<br>[ <b>{vertexId:corenessValue},...</b> ],<br>where<br><b>vertexId</b> is of the string type.<br><b>corenessValue</b> is of the integer type. |

### 5.1.6.2.5 K-Hop

**Table 5-148** parameters parameter description

| Parameter  | Mandatory | Type    | Description   |
|------------|-----------|---------|---|
| k          | Yes       | Integer | Number of hops. The value ranges from 1 to 100.   |
| num_thread | No        | Integer | Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to <b>1</b> . If the value is greater than 40, it is automatically set to <b>40</b> . The default value is <b>4</b> . |
| source     | Yes       | String  | Vertex ID   |



| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| mode       | No        | String  | Direction. The options are: <ul style="list-style-type: none"> <li>• <b>OUT</b>: Hop from the outgoing edges</li> <li>• <b>IN</b>: Hop from the incoming edges</li> <li>• <b>ALL</b>: Hop from edges in both directions</li> </ul> The default value is <b>OUT</b> . |
| statistics | No        | Boolean | Whether to only return the neighbor count statistics results, with a value of <b>true</b> or <b>false</b> . The default value is <b>false</b> .  |

**Table 5-149** response\_data parameter description

| Parameter       | Type    | Description  |
|-----------------|---------|--|
| vertices        | List    | ID of the vertex within k hops. The format is as follows:<br>[vertexId,...],<br>where <b>vertexId</b> is of the string type. |
| source          | String  | Source vertex ID   |
| k               | Integer | Number of hops   |
| k_hop_neighbors | Integer | Number of vertices within k hops (excluding the source vertex)   |

### 5.1.6.2.6 Common Neighbors

**Table 5-150** parameters parameter description

| Parameter | Mandatory | Description      | Type   | Value Range | Default Value |
|-----------|-----------|------------------|--------|-------------|---------------|
| source    | Yes       | Source vertex ID | String | -           | -             |
| target    | Yes       | Target vertex ID | String | -           | -             |

**Table 5-151** response\_data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| vertices         | List    | Common neighbor vertices. The format is as follows:<br>[vertexId,...],<br>where<br><b>vertexId</b> is of the string type |
| common_neighbors | Integer | Number of common neighbor vertices   |
| source           | String  | Source vertex ID   |
| target           | String  | Target vertex ID   |

### 5.1.6.2.7 Common Neighbors of Vertex Sets

**Table 5-152** parameters parameter description

| Parameter       | Mandatory | Description          | Type   | Value Range   | Default Value |
|-----------------|-----------|----------------------|--------|---|---------------|
| sources (2.2.6) | Yes       | Source vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |
| targets (2.2.6) | Yes       | Target vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Mike,Amy</b> .<br>The maximum ID number is 100000.    | -             |

| Parameter           | Mandatory | Description                            | Type    | Value Range   | Default Value |
|---------------------|-----------|--|---------|---|---------------|
| restricted (2.2.13) | No        | Whether other constraints are included | Boolean | <p><b>true</b> or <b>false</b></p> <ul style="list-style-type: none"> <li><b>false</b>: There is no additional constraint. The found common neighbors are the intersection of the neighborhoods corresponding to the source vertex set and target vertex set.</li> <li><b>true</b>: There are additional constraints. The found common neighbors are not only the intersection of the neighborhoods corresponding to the source vertex set and target vertex set, but each vertex in the common neighbor set has at least two neighboring vertices in the source vertex set and target vertex set.</li> </ul> | true          |

**Table 5-153** response\_data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| vertices         | List    | Common neighbor vertices. The format is as follows:<br>[vertexId,...],<br>where <b>vertexId</b> is of the string type. |
| common_neighbors | Integer | Number of common neighbors   |

### 5.1.6.2.8 Link Prediction

**Table 5-154** parameters parameter description

| Parameter | Mandatory | Description      | Type   | Value Range | Default Value |
|-----------|-----------|------------------|--------|-------------|---------------|
| source    | Yes       | Source vertex ID | String | -           | -             |
| target    | Yes       | Target vertex ID | String | -           | -             |

**Table 5-155** response\_data parameter description

| Parameter       | Type   | Description            |
|-----------------|--------|------------------------|
| source          | String | Source vertex ID       |
| target          | String | Target vertex ID       |
| link_prediction | Double | Link prediction result |

### 5.1.6.2.9 Shortest Path

**Table 5-156** parameters parameter description

| Parameter | Mandatory | Description                | Type   | Value Range | Default Value |
|-----------|-----------|----------------------------|--------|-------------|---------------|
| source    | Yes       | Source vertex ID of a path | String | -           | -             |
| target    | Yes       | Target vertex ID of a path | String | -           | -             |

| Parameter  | Mandatory | Description                            | Type    | Value Range  | Default Value |
|------------|-----------|--|---------|--|---------------|
| weight     | No        | Weight of an edge                      | String  | Empty or character string <ul style="list-style-type: none"> <li>Empty: The default weight and distance of edges are <b>1</b>.</li> <li>Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is <b>1</b> by default.</li> </ul> <b>NOTE</b><br>The weight of an edge must be greater than <b>0</b> . | -             |
| directed   | No        | Whether to consider the edge direction | Boolean | The value can be <b>true</b> or <b>false</b> .   | false         |
| timeWindow | No        | Time window used for time filtering    | Object  | For details, see <a href="#">Table 5-157</a> . <b>NOTE</b><br><b>timeWindow</b> does not support the shortest path with weight. That is, parameters <b>timeWindow</b> and <b>weight</b> cannot be both specified.  | -             |

**Table 5-157** timeWindow parameter description

| Parameter  | Mandatory | Description                                       | Type   | Value Range  | Default Value |
|------------|-----------|---|--------|--|---------------|
| filterName | Yes       | Name of the time property used for time filtering | String | Character string: The property on the corresponding vertex/edge is used as the time.                         | -             |
| filterType | No        | Filtering by vertex or edge                       | String | <b>V</b> : filtering by vertex<br><b>E</b> : filtering by edge<br><b>BOTH</b> : filtering by vertex and edge | BOTH          |
| startTime  | No        | Start time  | String | Date character string or timestamp   | -             |

| Parameter | Mandatory | Description | Type   | Value Range                        | Default Value |
|-----------|-----------|-------------|--------|------------------------------------|---------------|
| endTime   | No        | End time    | String | Date character string or timestamp | -             |

**Table 5-158** response\_data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| path      | List   | Shortest path. The format is as follows:<br>[vertexId,...]<br>where<br><b>vertexId</b> is of the string type. |
| source    | String | Source vertex ID  |
| target    | String | Target vertex ID  |

### 5.1.6.2.10 All Shortest Paths

**Table 5-159** parameters parameter description

| Parameter | Mandatory | Description                            | Type    | Value Range                 | Default Value |
|-----------|-----------|--|---------|-----------------------------|---------------|
| source    | Yes       | Source vertex ID of a path             | String  | -                           | -             |
| target    | Yes       | Target vertex ID of a path             | String  | -                           | -             |
| directed  | No        | Whether to consider the edge direction | Boolean | <b>true</b> or <b>false</b> | false         |

**Table 5-160** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| paths     | List | All shortest paths between the source vertex and target vertex. The format is as follows:<br>[[path1],[path2]] |

| Parameter    | Type    | Description      |
|--------------|---------|------------------|
| paths_number | Integer | Number of paths  |
| source       | String  | Source vertex ID |
| target       | String  | Target vertex ID |

### 5.1.6.2.11 Filtered Shortest Path

#### Request

- Parameter description

**Table 5-161** parameters parameter description

| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| source     | Yes       | String  | Source vertex ID of a path   |
| target     | Yes       | String  | Target vertex ID of a path   |
| directed   | No        | Boolean | Whether to consider the edge direction. The default value is <b>false</b> .  |
| num_thread | No        | Integer | Number of concurrent threads. The value ranges from <b>1</b> to <b>40</b> . If the value is less than <b>1</b> , it is automatically set to <b>1</b> . If the value is greater than <b>40</b> , it is automatically set to <b>40</b> . The default value is <b>4</b> . |

- Request example
  - Synchronization

```

{
  "executionMode": "sync",
  "algorithmName": "filtered_shortest_path",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "IN",
      "rightvalue": {
        "value": [
          "xxx",
          "rate"
        ]
      }
    }
  },
  "vertex_filter": {
    "property_filter": {

```

```

    "leftvalue": {
      "property_name": "title"
    },
    "predicate": "PREFIX",
    "rightvalue": {
      "value": "tr_"
    }
  }
},
"parameters": {
  "source": "tr_1",
  "target": "tr_117",
  "directed": true
}
}

```

- Asynchronization

```

{
  "executionMode": "async",
  "algorithmName": "filtered_shortest_path",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "IN",
      "rightvalue": {
        "value": [
          "xxx",
          "rate"
        ]
      }
    }
  },
  "vertex_filter": {
    "property_filter": {
      "leftvalue": {
        "property_name": "title"
      },
      "predicate": "PREFIX",
      "rightvalue": {
        "value": "tr_"
      }
    }
  },
  "parameters": {
    "source": "tr_1",
    "target": "tr_117",
    "directed": true
  }
}

```

## Response

- Synchronous **data** parameter description

**Table 5-162** response\_data parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| path      | Yes       | List   | Vertex result set. If the last layer of <b>filters</b> is vertex filtering, the <b>data</b> contains vertices. |
| source    | Yes       | String | Source vertex ID   |



| Parameter | Mandatory | Type   | Description            |
|-----------|-----------|--------|------------------------|
| target    | Yes       | String | Target vertex ID       |
| runtime   | Yes       | Double | Algorithm running time |

- Response example

- Synchronous response example (successful request)

```
{
  "data": {
    "outputs": {
      "path": [
        "tr_1",
        "tr_5",
        "tr_26",
        "tr_117"
      ],
      "runtime": 0.735766,
      "source": "tr_1",
      "target": "tr_117"
    }
  }
}
```

- Synchronous response example (failed request)

```
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8402"
}
```

- Asynchronous response parameters

**Table 5-163** response\_data parameter description

| Parameter    | Mandatory | Type    | Description  |
|--------------|-----------|---------|--|
| errorMessage | No        | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | No        | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |
| jobId        | No        | String  | ID of the algorithm execution job. This parameter is left blank when the request fails.  |
| jobType      | No        | Integer | Job type. This parameter is left blank when the request fails.   |

- Example response

- Asynchronous response example (successful request)

```
{
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",
  "jobType": 2
}
```

- Asynchronous response example (failed request)

```
{
  "errorMessage": "graph [test_117d] is not found",
  "errorCode": "GES.8402"
}
```

### 5.1.6.2.12 SSSP

**Table 5-164** parameters parameter description

| Parameter | Mandatory | Description                            | Type    | Value Range                 | Default Value |
|-----------|-----------|--|---------|-----------------------------|---------------|
| source    | Yes       | Vertex ID                              | String  | -                           | -             |
| directed  | No        | Whether to consider the edge direction | Boolean | <b>true</b> or <b>false</b> | true          |

**Table 5-165** response\_data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| distance  | List   | Path length of each vertex in the graph from the source vertex. The format is as follows:<br>[ <b>{vertexId:distanceValue}</b> ,...],<br>where<br><b>vertexId</b> is of the string type.<br><b>distanceValue</b> is of the double type. |
| source    | String | Source vertex ID  |

### 5.1.6.2.13 Shortest Path of Vertex Set

**Table 5-166** parameters parameter description

| Parameter  | Mandatory | Description                            | Type    | Value Range   | Default Value |
|------------|-----------|--|---------|---|---------------|
| sources    | Yes       | Source vertex ID set                   | String  | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |
| targets    | Yes       | Target vertex ID set                   | String  | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |
| directed   | No        | Whether to consider the edge direction | Boolean | <b>true</b> or <b>false</b>   | false         |
| timeWindow | No        | Time window used for time filtering    | Object  | For details, see <a href="#">Table 5-167</a> .  | -             |

**Table 5-167** timeWindow parameter description

| Parameter  | Mandatory | Description                                       | Type   | Value Range  | Default Value |
|------------|-----------|---|--------|--|---------------|
| filterName | No        | Name of the time property used for time filtering | String | Character string: The property on the corresponding vertex/edge is used as the time.                         | -             |
| filterType | No        | Filtering by vertex or edge                       | String | <b>V</b> : filtering by vertex<br><b>E</b> : filtering by edge<br><b>BOTH</b> : filtering by vertex and edge | <b>BOTH</b>   |
| startTime  | No        | Start time  | String | Date character string or timestamp   | -             |

| Parameter | Mandatory | Description | Type   | Value Range                        | Default Value |
|-----------|-----------|-------------|--------|------------------------------------|---------------|
| endTime   | No        | End time    | String | Date character string or timestamp | -             |

**Table 5-168** response\_data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| path      | List   | Shortest path. The format is as follows:<br>[vertexId,...]<br>where<br><b>vertexId</b> is of the string type. |
| source    | String | Source vertex ID  |
| target    | String | Target vertex ID  |

### 5.1.6.2.14 n-Paths

**Table 5-169** parameters parameter description

| Parameter | Mandatory | Description                            | Type    | Value Range                 | Default Value |
|-----------|-----------|--|---------|-----------------------------|---------------|
| source    | Yes       | Source vertex ID of a path             | String  | -                           | -             |
| target    | Yes       | Target vertex ID of a path             | String  | -                           | -             |
| directed  | No        | Whether to consider the edge direction | Boolean | <b>true</b> or <b>false</b> | false         |
| n         | No        | Number of paths                        | Integer | 1 to 100                    | 10            |
| k         | No        | Number of hops                         | Integer | 1 to 10                     | 5             |

**Table 5-170** response\_data parameter description

| Parameter    | Type    | Description  |
|--------------|---------|--|
| paths        | List    | Paths between the source vertex and target vertex. The format is as follows: [[path1],[path2]] |
| paths_number | Integer | Number of paths  |
| source       | String  | Source vertex ID   |
| target       | String  | Target vertex ID   |

### 5.1.6.2.15 Filtered n-Paths

#### Introduction

The filtered n-Paths algorithm is used to find no more than n k-hop loop-free paths between the source and target vertices. The start vertex (source), end vertex (target), number of hops (k), number of paths (n), and filter criteria (filters) are the parameters for the algorithm.

- Algorithm name: filtered\_n\_paths
- filtered\_n\_paths

#### Applicable Scope

Any network

#### Request Parameters

**Table 5-171** Body format

| Field         | Mandatory | Type        | Description   |
|---------------|-----------|-------------|---|
| algorithmName | Yes       | String      | The value is <b>filtered_n_paths</b> .  |
| parameters    | Yes       | JSON format | For details about the format, see <a href="#">Table 5-172</a> .   |
| filters       | Yes       | JSON Array  | Filter criteria. Each element in the array corresponds to a filter. For details about the format, see <a href="#">Table 5-173</a> . |

**Table 5-172** parameters parameter description

| Parameter | Mandatory | Description     | Type   | Value Range       | Default Value |
|-----------|-----------|-----------------|--------|-------------------|---------------|
| source    | Yes       | Source vertex   | String | Internal vertices | None          |
| target    | Yes       | Target vertex   | String | Internal vertices | None          |
| k         | Yes       | Number of hops  | Int    | [2,6]             | 2             |
| n         | Yes       | Number of paths | Int    | [1,1000]          | 1             |

**Table 5-173** filters element format

| Parameter     | Mandatory | Type | Value Range | Default Value | Description                                 |
|---------------|-----------|------|-------------|---------------|---|
| edge_filter   | No        | json | N/A         | N/A           | Filter criteria for full-graph edge query   |
| vertex_filter | No        | json | None        | None          | Filter criteria for full-graph vertex query |

**Table 5-174** response\_data parameter description

| Field        | Mandatory | Type      | Description  |
|--------------|-----------|-----------|--|
| path_length  | Yes       | int       | Path length  |
| paths_number | Yes       | int       | Number of paths  |
| paths        | Yes       | JSONArray | Path set. Example value:<br>["111","119","58","96","82","57","56"] |
| source       | Yes       | String    | Source vertex  |
| target       | Yes       | String    | Target vertex  |

## Example Request

```
POST http://IP:PORT/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_n_paths",
  "filters": [
```

```
{
  "edge_filter":
  {
    "property_filter":
    {
      "leftvalue":
      {
        "label_name": "labelName"
      },
      "predicate": "=",
      "rightvalue":
      {
        "value": "default"
      }
    }
  }
},
"parameters":
{
  "k": 6,
  "n": 100,
  "source": "111",
  "target": "56"
}
}
```

## Response

```
{
  "jobId": "b14f6380-f115-46ab-990e-9a76a984ebd2154236181",
  "jobType": 2
}
```

## Example for Querying a Job

GET http://IP:PORT/ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs/{jobId}/status

## Query Result

```
{
  "data": {
    "outputs": {
      "data_return_size": 3,
      "paths": [
        [
          "111",
          "119",
          "58",
          "96",
          "82",
          "57",
          "56"
        ],
        [
          "111",
          "119",
          "58",
          "61",
          "76",
          "57",
          "56"
        ],
        [
          "111",
          "119",
          "58",
          "79",
          "76",
          "56"
        ]
      ]
    }
  }
}
```

```

        "57",
        "56"
    ]
  ],
  "runtime": 0.000308,
  "source": "111",
  "path_length": 6,
  "data_offset": 0,
  "paths_number": 3,
  "data_total_size": 3,
  "target": "56"
}
},
"status": "success"
}

```

### 5.1.6.2.16 Filtered All Pairs Shortest Paths

Table 5-175 parameters parameter description

| Parameter | Mandatory | Description   | Type    | Value Range   | Default Value |
|-----------|-----------|---|---------|---|---------------|
| sources   | Yes       | Set of start vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,). | String  | The number of source vertices cannot exceed 10,000. | -             |
| targets   | Yes       | Set of end vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,).   | String  | The number of target vertices cannot exceed 10,000. | -             |
| directed  | No        | Whether the edges are directed  | Boolean | The value can be <b>true</b> or <b>false</b> .      | false         |
| cutoff    | No        | Maximum length  | Integer | 1-100   | 6             |



| Parameter  | Mandatory | Description             | Type    | Value Range   | Default Value  |
|------------|-----------|-------------------------|---------|---|----------------|
| path_limit | No        | Maximum number of paths | Integer | <ul style="list-style-type: none"> <li>For synchronous tasks:<br/>The value ranges from 1 to 100000. The default value is <b>100000</b>.</li> <li>For asynchronous tasks:<br/>The value ranges from 1 to 1000000. The default value is <b>1000000</b>.</li> </ul> | 100000/1000000 |

 **NOTE**

- Synchronous tasks: Number of source vertices x Number of target vertices x Maximum path length (**cutoff**) <= 1000000, Maximum number of paths (**path\_num**) x Maximum path length (**cutoff**) <= 1000000.
- This algorithm checks memory capacity. When the memory is insufficient, the error "memory is not enough" is reported.

**Table 5-176** response\_data parameter description

| Parameter    | Type    | Description  |
|--------------|---------|--|
| batch_paths  | List    | Batch paths. Format:<br>[paths_element,...]<br>where<br><b>Paths_element</b> indicates the path from a source to a target. The format is as follows:<br>{<br>"paths": [<br>[<br>"Alice",<br>"Janet",<br>"Sue",<br>"Serena",<br>"Bonnie"<br>]<br>],<br>"source": "Alice",<br>"target": "Bonnie"<br>}, |
| paths_number | Integer | Number of paths  |

### 5.1.6.2.17 All Shortest Paths of Vertex Sets

**Table 5-177** parameters parameter description

| Parameter | Mandatory | Description          | Type   | Value Range   | Default Value |
|-----------|-----------|----------------------|--------|---|---------------|
| sources   | Yes       | Source vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |

| Parameter | Mandatory | Description                            | Type    | Value Range   | Default Value |
|-----------|-----------|--|---------|---|---------------|
| targets   | Yes       | Target vertex ID set                   | String  | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |
| directed  | No        | Whether to consider the edge direction | Boolean | <b>true</b> or <b>false</b> . It is a Boolean value.  | false         |

**Table 5-178** response\_data parameter description

| Parameter | Type   | Description  |
|-----------|--------|--|
| paths     | List   | All shortest paths between the source vertex and target vertex. The format is as follows:<br>[[path1],[path2]] |
| source    | String | Source ID of a path  |
| target    | String | Target ID of a path  |

### 5.1.6.2.18 Filtered All Shortest Paths

#### Parameters

**Table 5-179** parameters parameter description

| Parameter | Mandatory | Description                    | Type    | Value Range                                    | Default Value |
|-----------|-----------|--------------------------------|---------|--|---------------|
| source    | Yes       | Source vertex ID               | String  | -<br>-   | -             |
| target    | Yes       | Target vertex ID               | String  | -  | -             |
| directed  | No        | Whether the edges are directed | Boolean | The value can be <b>true</b> or <b>false</b> . | false         |

**Table 5-180** response\_data parameter description

| Parameter    | Type    | Description   |
|--------------|---------|---|
| paths        | List    | Paths between the source and target vertices. The format is as follows:<br>[[path1],[path2]]<br>where<br>For the format of each path, see <a href="#">Shortest Path</a> . |
| paths_number | Integer | Number of paths   |
| source       | String  | Source vertex ID  |
| target       | String  | Target vertex ID  |

### Example Request

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_all_shortest_paths",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "=",
      "rightvalue": {
        "value": "friends"
      }
    }
  },
  "parameters": {
    "source": "Alice",
    "target": "Jay",
    "directed": true
  }
}
```

### Response

```
{
  "data": {
    "outputs": {
      "data_return_size": 8,
      "paths": [
        [
          "Alice",
          "Janet",
          "Yvette",
          "Willy",
          "Jay"
        ],
        ...
        [
          "Alice",
          "Jacob",
          "Jimmy",
          "Cary",
          "Jay"
        ]
      ]
    }
  }
}
```

```

    "runtime": 0.005276,
    "source": "Alice",
    "data_offset": 0,
    "paths_number": 8,
    "data_total_size": 8,
    "target": "Jay"
  }
}
}

```

**Table 5-181** response\_data parameter description

| Parameter    | Type    | Description   |
|--------------|---------|---|
| paths        | List    | Paths between the source and target vertices. The format is as follows:<br>[[path1],[path2]]<br>where<br>For the format of each path, see <a href="#">Shortest Path</a> . |
| paths_number | Integer | Number of paths   |
| source       | String  | Source vertex ID  |
| target       | String  | Target vertex ID  |

### 5.1.6.2.19 Connected Component

 NOTE

This algorithm can run without specifying its **parameters**.

**Table 5-182** response\_data parameter description

| Parameter    | Type    | Description  |
|--------------|---------|--|
| Max_WCC_size | Integer | Maximum number of vertices in the largest connected component  |
| Max_WCC_id   | String  | ID of the largest connected component  |
| community    | List    | Connected component set (community) corresponding to each vertex, in <b>[{vertexId:communityId},...]</b> format.<br>where<br><b>vertexId</b> is of the string type.<br><b>communityId</b> is of the string type. |

### 5.1.6.2.20 Label Propagation

**Table 5-183** parameters parameter description

| Parameter      | Mandatory | Description   | Type    | Value Range   | Default Value |
|----------------|-----------|---|---------|---|---------------|
| convergence    | No        | Convergence   | Double  | A real number between 0 and 1   | 0.00001       |
| max_iterations | No        | Maximum iterations  | Integer | An integer ranging from 1 to 2147483647. For frontend calls, the range is [1,2000].   | 1000          |
| initial        | No        | Name of the property used as the initialization label on a vertex | String  | <p>Empty or character string</p> <ul style="list-style-type: none"> <li>Empty: Each vertex is allocated with a unique initialization label. This method is applicable to scenarios where no vertex label information exists.</li> <li>Character string: The value of the property field corresponding to each vertex is used as the initialization label (the type is string, and the initialization label field is left blank for a vertex with unknown labels). This method is applicable to scenarios where some vertex labels are marked to predict unknown vertex labels.</li> </ul> <p><b>NOTE</b><br/>If the value of <b>initial</b> is a character string, the number of vertices with initialization labels must be greater than 0 and less than the total number of vertices.</p> | -             |

 **NOTE**

For details about algorithm iterations and convergence, see [Iterations and Convergence of PageRank](#).

**Table 5-184** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| community | List | Community corresponding to each vertex. The format is as follows:<br>[ <b>{vertexId:communityId},...</b> ]<br>where<br><b>vertexId</b> is of the string type.<br><b>communityId</b> is of the string type. |

### 5.1.6.2.21 Louvain

**Table 5-185** parameters parameter description

| Parameter      | Mandatory | Description        | Type    | Value Range   | Default Value |
|----------------|-----------|--------------------|---------|---|---------------|
| convergence    | No        | Convergence        | Double  | A real number between 0 and 1   | 0.00001       |
| max_iterations | No        | Maximum iterations | Integer | An integer ranging from 1 to 2147483647. For frontend calls, the range is [1,2000].   | 100           |
| weight         | No        | Weight of an edge  | String  | Empty or null character string <ul style="list-style-type: none"> <li>• Empty: The default weight and distance of edges are <b>1</b>.</li> <li>• Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is <b>1</b> by default.</li> </ul> <b>NOTE</b><br>The weight of an edge must be greater than <b>0</b> . | weight        |

 NOTE

For details about algorithm iterations and convergence, see [Iterations and Convergence of PageRank](#).

**Table 5-186** response\_data parameter description

| Parameter     | Type    | Description  |
|---------------|---------|--|
| modularity    | Double  | Modularity   |
| community_num | Integer | Number of communities  |
| community     | List    | Community corresponding to each vertex. The format is as follows:<br>[ <b>{vertexId:communityId},...</b> ]<br>where<br><b>vertexId</b> is of the string type.<br><b>communityId</b> is of the string type. |

### 5.1.6.2.22 Node2vec

**Table 5-187** parameters parameter description

| Parameter  | Mandatory | Description                                 | Type    | Value Range                      | Default Value |
|------------|-----------|---|---------|----------------------------------|---------------|
| P          | No        | Rollback parameter                          | Double  | Greater than 0                   | 1             |
| Q          | No        | Forward parameter                           | Double  | Greater than 0                   | 1             |
| dim        | No        | Mapping dimension                           | Integer | An integer ranging from 1 to 200 | 50            |
| walkLength | No        | Random walk length                          | Integer | An integer ranging from 1 to 100 | 40            |
| walkNumber | No        | Number of random walk steps of each vertex. | Integer | An integer ranging from 1 to 100 | 10            |
| iterations | No        | Number of iterations                        | Integer | An integer ranging from 1 to 100 | 10            |



**Table 5-188** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| embedding | List | Vector representation of each vertex mapped to the Euclidean space. The format is as follows:<br>[ <b>{vertexId:vectorValue}</b> ]<br>where<br><b>vertexId</b> is of the string type.<br><b>vectorValue</b> : is a euclidean vector, for example, [-0.485, -0.679, 0.356]. |

### 5.1.6.2.23 Real-time Recommendation

**Table 5-189** parameters parameter description

| Parameter | Mandatory | Description   | Type    | Value Range  | Default Value |
|-----------|-----------|---|---------|--|---------------|
| sources   | Yes       | Vertex ID. Multiple vertices are supported.                 | String  | The number of source vertices cannot exceed the upper limit (30 by default, which can be changed using <b>source_limit</b> ) . Use commas (,) to separate the IDs. | -             |
| alpha     | No        | Weight coefficient. A larger value indicates a longer step. | Double  | A real number between 0 and 1  | 0.85          |
| N         | No        | Total number of walk steps                                  | Integer | 1 to 200000  | 10000         |

| Parameter    | Mandatory | Description  | Type    | Value Range   | Default Value |
|--------------|-----------|--|---------|---------------|---------------|
| nv           | No        | Parameter indicating that the walk process ends ahead of schedule: minimum number of access times of a potential recommended vertex<br><b>NOTE</b><br>If a vertex is accessed during random walk and the number of access times reaches <b>nv</b> , the vertex will be recorded as the potential recommended vertex.   | Integer | 1 to 10       | 5             |
| np           | No        | Parameter indicating that the walk process ends ahead of schedule: number of potential recommended vertices<br><b>NOTE</b><br>If the number of potential recommended vertices of a source vertex reaches <b>np</b> , the random walk for the source vertex ends ahead of schedule.   | Integer | 1 to 2000     | 1000          |
| label        | No        | Expected type of the vertex to be output.<br><b>NOTE</b> <ul style="list-style-type: none"> <li>Expected type of the vertex to be output. If the value is null, the original calculation result of the algorithm is output without considering the vertex type.</li> <li>If the value is not null, vertices with the <b>label</b> are filtered from the calculation result.</li> </ul> | String  | Vertex label  | -             |
| directed     | No        | Whether to consider the edge direction   | Boolean | true or false | true          |
| source_limit | No        | Maximum number of source vertices  | Int     | 1~100000      | 30            |

| Parameter  | Mandatory | Description   | Type | Value Range   | Default Value |
|------------|-----------|---|------|---------------|---------------|
| restricted | No        | Whether to accept invalid source vertices<br><b>restricted=true</b> : If a vertex that does not exist in the graph is passed to <b>sources</b> , an error is reported.<br><b>restricted=false</b> : A vertex that does not exist in the graph can be passed to <b>sources</b> . However, if all source vertices do not exist, an error is reported. | Bool | true or false | true          |

**Table 5-190** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| score     | List | Score of each vertex, which reflects the recommendation degree. A larger value indicates a higher recommendation degree. The format is as follows:<br>[ <b>vertexId</b> : <b>scoreValue</b> ,...]<br>where<br><b>vertexId</b> is of the string type.<br><b>scoreValue</b> is of the double type. |
| sources   | List | ID of the source vertex  |

### 5.1.6.2.24 Degree Correlation

**Table 5-191** response\_data parameter description

| Parameter          | Type   | Description        |
|--------------------|--------|--------------------|
| degree_correlation | Double | Degree correlation |

### 5.1.6.2.25 Triangle Count

**Table 5-192** parameters parameter description

| Parameter  | Mandatory | Description   | Type    | Value Range  |
|------------|-----------|---|---------|--|
| statistics | No        | Whether to export only the total statistical result. <ul style="list-style-type: none"> <li><b>true</b>: Export only the statistical result.</li> <li><b>false</b>: Export the number of triangles corresponding to each vertex.</li> </ul> | Boolean | <b>true</b> or <b>false</b> . The default value is <b>true</b> . |

**Table 5-193** response\_data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| triangle_count   | Integer | Number of triangles  |
| vertex_triangles | List    | Number of triangles on each vertex. The format is as follows:<br>[ <b>{vertexId : vertexTriangleCount},...</b> ],<br>where<br><b>vertexId</b> is of the string type.<br><b>vertexTriangleCount</b> is of the integer type. |

### 5.1.6.2.26 Cluster Coefficient

**Table 5-194** response\_data parameter description

| Parameter           | Type    | Description  |
|---------------------|---------|--|
| cluster_coefficient | Double  | Cluster coefficient  |
| statistics          | Boolean | Whether to only return the global average clustering coefficient. The default value is <b>true</b> . |

### 5.1.6.2.27 Closeness Centrality

**Table 5-195** parameters parameter description

| Parameter | Mandatory | Description                       | Type   | Value Range | Default Value |
|-----------|-----------|-----------------------------------|--------|-------------|---------------|
| source    | Yes       | ID of the vertex to be calculated | String | -           | -             |

**Table 5-196** response\_data parameter description

| Parameter | Type   | Description                 |
|-----------|--------|-----------------------------|
| closeness | Double | Closeness centrality degree |
| source    | String | Vertex ID to be calculated  |

### 5.1.6.2.28 Filtered Circle Detection

#### Request example

```

Post http://{}/ges/v1.0/1/graphs/movie/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_circle_detection",
  "parameters": {
    "n": 10,
    "statistics": true,
    "output_format": "edgeld"
  },
  "filters": [
    {
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "transfer"
          }
        }
      }
    }
  ],
  "times": 5
}

```

## Parameters

**Table 5-197** parameters parameter description

| Parameter     | Mandatory | Description   | Type    | Value Range  | Default Value  |
|---------------|-----------|---|---------|--|--|
| sources       | No        | Set of source vertex IDs to be queried  | String  | -  | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana.</b> |
| n             | No        | Upper limit of the number of enumerated circles that meet the filter criteria | Integer | [1,100000]   | 100  |
| statistics    | No        | Whether to export the number of circles that meet the filter criteria         | Boolean | <b>true</b> or <b>false</b>                            | false  |
| batch_number  | No        | Number of source vertices for batch processing                                | Integer | [1,1000]   | 10   |
| output_format | No        | Output format   | String  | <b>vertexId</b> , <b>edgeId</b> , or <b>edgeObject</b> | edgeObject   |
| filters       | Yes       | Filter criteria. Each element in the array corresponds to a filter.           | Object  | -  | -  |

**Table 5-198** filters element formats

| Parameter     | Mandatory | Description   | Type    | Value Range             | Default Value |
|---------------|-----------|---|---------|-------------------------|---------------|
| operator      | No        | Direction of the query to be performed at the current layer   | String  | <b>out, in, or both</b> | out           |
| edge_filter   | No        | Filter criteria for the current layer. For details, see <a href="#">Table 5-304</a> in the <b>Filtered-query API</b> .            | Object  | -                       | -             |
| vertex_filter | No        | Filter criteria of vertices at the current layer. For details, see <a href="#">Table 5-304</a> in the <b>Filtered-query API</b> . | Object  | -                       | -             |
| times         | No        | Number of layers queried using the same filter criteria   | Integer | [1,10]                  | 1             |

 **NOTE**

- Filter criteria at the first layer are used to filter source vertices. Therefore, only the **vertex\_filter** parameter is valid.
- Filter criteria at the last layer are used to filter source vertices.
- The circle length ranges from 3 to 10. Therefore, the number of filtering layers is 4 to 11.

**Table 5-199** response\_data parameter description

| Parameter     | Mandatory | Type    | Description  |
|---------------|-----------|---------|--|
| circles       | Yes       | List    | Set of circles found. The format is <b>[[circle1], [circle2], ...]</b> . The <b>circle</b> format is as follows: <ul style="list-style-type: none"> <li>If <b>output_format</b> is <b>edgeObject</b>, the format is <b>[{"source": sourceId, "target": targetId, "index": edgeIndex}, ...]</b>, where <b>sourceId</b>, <b>targetId</b>, and <b>edgeIndex</b> are of the string type.</li> <li>If <b>output_format</b> is <b>edgeId</b>, the format is <b>[sourceId-targetId-edgeIndex,...]</b>, where <b>sourceId-targetId-edgeIndex</b> is of the string type.</li> <li>If <b>output_format</b> is <b>vertexId</b>, the format is <b>[vertexId, ...]</b>, where <b>vertexId</b> is of the string type.</li> </ul> |
| runtime       | Yes       | Double  | Algorithm running time   |
| n             | Yes       | Integer | Maximum number of enumerated circles   |
| circle_number | No        | Integer | When <b>statistics</b> is set to <b>true</b> , the number of circles that meet filter criteria is displayed.   |

### 5.1.6.2.29 Subgraph Matching

**Table 5-200** parameters parameter description

| Parameter | Mandatory | Description  | Type   | Value Range  |
|-----------|-----------|--|--------|--|
| edges     | Yes       | Edge set of the subgraph to be matched. The vertex ID must be of the <code>size_t</code> type. | String | The value is in standard CSV format. The start and end vertices of an edge are separated by a comma (,), and edges are separated by a newline character (\n). For example, <b>1,2\n2,3</b> . |



| Parameter    | Mandatory | Description   | Type    | Value Range  |
|--------------|-----------|---|---------|--|
| vertices     | Yes       | Label of each vertex on the subgraph to be matched.                     | String  | The value is in standard CSV format. Vertices and their labels are separated by commas (,), and labels are separated by newline characters (\n). For example, <b>1,BP\n2,FBP\n3,CP</b> . |
| directed     | No        | Whether to consider the direction of the graph                          | Boolean | The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> .  |
| n            | No        | Maximum number of subgraphs to be searched for                          | Integer | The value range is [1,100000]. The default value is <b>100</b> .   |
| batch_number | No        | Number of queries processed in batches each time                        | Integer | The value range is [1,1000000]. The default value is <b>10000</b> .  |
| statistics   | No        | Whether to display the number of all subgraphs that meet the conditions | Boolean | The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> .   |

**Table 5-201** response\_data parameter description

| Parameter       | Mandatory | Type    | Description  |
|-----------------|-----------|---------|--|
| subgraphs       | Yes       | List    | Subgraphs with the same pattern of the <b>pattern_graph</b> . The value is in the [[subgraph1],[subgraph2], ...] format. Each subgraph is in the [vertex1,vertex2, ...] format, where <b>vertex</b> is of the string type. The vertices of each subgraph correspond to those of <b>pattern_graph</b> . |
| pattern_graph   | Yes       | List    | Graph pattern. The value is in the [vertex1,vertex2, ...] format, where vertex is of the string type.  |
| subgraph_number | No        | Integer | Number of matched graphs. When <b>statistics</b> is set to <b>true</b> , the total number of graphs that meet query conditions is displayed.   |

### 5.1.6.2.30 Topicrank

**Table 5-202** parameters parameter description

| Parameter      | Mandatory | Description   | Type    | Value Range   | Default Value |
|----------------|-----------|---|---------|---|---------------|
| sources        | Yes       | Vertex ID. You can specify multiple node IDs in CSV format and separate them with commas (,). | String  | Currently, a maximum of 100000 IDs are allowed.                                     | -             |
| actived_p      | No        | Initial weight of the sources vertex.   | Double  | The value ranges from 0 to 100000.  | 1             |
| default_p      | No        | Initial weight of a non-source vertex   | Double  | The value ranges from 0 to 100000.  | 1             |
| filtered       | No        | Whether to filter results   | Boolean | The value can be <b>true</b> or <b>false</b> .                                      | false         |
| only_neighbors | No        | Whether to display only the neighboring vertices of the sources                               | Boolean | The value can be <b>true</b> or <b>false</b> .                                      | false         |
| alpha          | No        | Weight coefficient  | String  | The value is a real number between 0 and 1.   | 0.85          |
| convergence    | No        | Convergence   | String  | The value is a real number between 0 and 1.   | 0.00001       |
| max_iterations | No        | Maximum iterations  | Integer | An integer ranging from 1 to 2147483647. For frontend calls, the range is [1,2000]. | 1000          |
| directed       | No        | Whether the edges are directed  | Boolean | The value can be <b>true</b> or <b>false</b> .                                      | true          |

| Parameter  | Mandatory | Description       | Type    | Value Range                    | Default Value |
|------------|-----------|-------------------|---------|--------------------------------|---------------|
| num_thread | No        | Number of threads | Integer | The value ranges from 1 to 40. | 4             |

**Table 5-203** response\_data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| topicrank | List | TopicRank value of each vertex. The format is as follows: [{vertexId:rankValue},...], where <b>vertexId</b> is of the string type. <b>rankValue</b> is of the double type. |

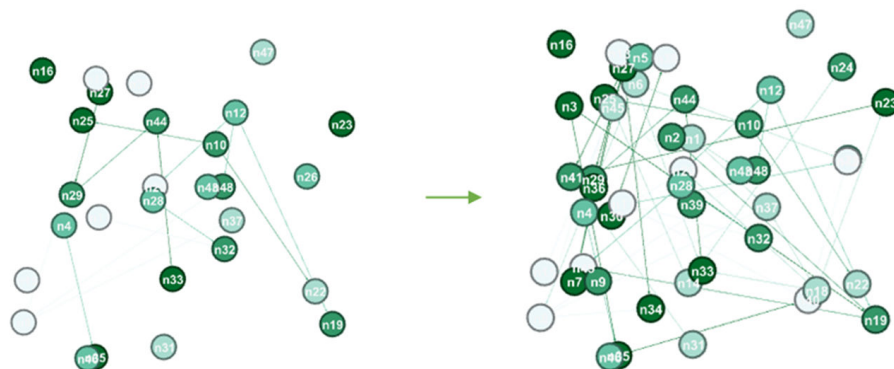
## 5.1.7 Temporal Graph APIs

### 5.1.7.1 Community Evolution (temporal\_graph)

#### Function

The community evolution algorithm generates a temporal graph that shows structure changes of a community over time.

**Figure 5-1** Principle



#### URL

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/dynamicgraphs/action?  
action\_id=execute-analysis

**Table 5-204** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-205** Request body parameters

| Parameter     | Mandatory | Type   | Description          |
|---------------|-----------|--------|----------------------|
| algorithmName | Yes       | String | Algorithm name       |
| dynamicRange  | Yes       | Object | Temporal parameters  |
| parameters    | Yes       | Object | Algorithm parameters |

**Table 5-206** dynamicRange parameters

| Parameter  | Mandatory | Type            | Description  |
|------------|-----------|-----------------|--|
| start      | Yes       | Date or integer | Start time of the temporal analysis. The start time must be earlier than the end time. |
| end        | Yes       | Date or integer | End time of the temporal analysis  |
| time_props | Yes       | Object          | Time properties for the temporal analysis  |

**Table 5-207** time\_props parameters

| Parameter | Mandatory | Type   | Description                     |
|-----------|-----------|--------|---------------------------------|
| stime     | Yes       | String | Property name of the start time |
| etime     | Yes       | String | Property name of the end time   |

**Table 5-208** parameters

| Parameter       | Mandatory | Type    | Description   |
|-----------------|-----------|---------|---|
| source          | Yes       | String  | Vertices in the community network. You can specify a maximum of 100,000 vertices.     |
| temporal_vertex | No        | Boolean | Whether to perform community evolution on a node. The default value is <b>false</b> . |

## Response Parameters

**Table 5-209** Parameters in a response

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| errorMessage | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.                |
| errorCode    | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.                   |
| status       | No        | String | Returned job status for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . If the query fails, the field is empty. |
| data         | No        | JSON   | Algorithm execution result. If the query fails, the field is empty.   |

## Example Request

Observe the community evolution of some nodes. The algorithm name is **temporal\_graph**, the start time of dynamic analysis is **\${startTime}**, and the end time is **\${endTime}**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName":"temporal_graph",
  "dynamicRange":{
    "start":"${startTime}",
    "end":"${endTime}",
    "time_props":{"stime":"${property(start_time)}","etime":"${property(start_time)}"}
  },
  "parameters":{
```

```
"sources":[],  
}  
}
```

## Example Response

### Status code: 200

Example response for a successful request

```
{  
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",  
  "jobType": 0  
}
```

### Status code: 400

Example response for a failed request

```
{  
  "errorMessage": "${errorMessage}",  
  "errorCode": "GES.8301"  
}
```

## Status Code

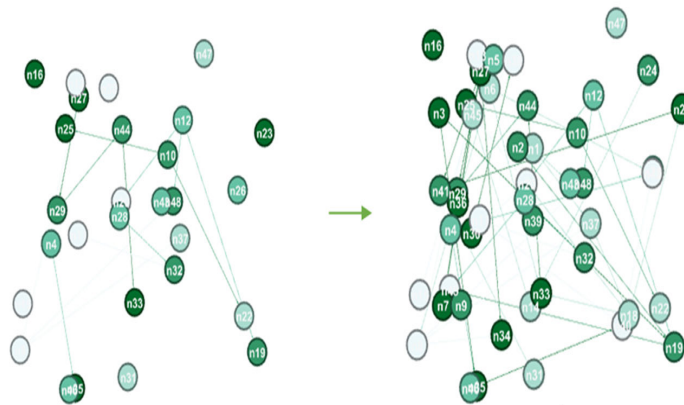
| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error            |
| 401 Unauthorized          | Authorization failed     |
| 403 Forbidden             | No operation permissions |
| 404 Not Found             | No resources found       |
| 500 Internal Server Error | Internal server error    |
| 503 Service Unavailable   | Service unavailable      |

### 5.1.7.2 Temporal BFS (temporal\_bfs)

#### Function

This algorithm searches for associated vertices based on temporal message passing and temporal BFS algorithms, and outputs the visit time of each vertex and the distance from the vertex to the source start vertex.

Figure 5-2 Principle



## URL

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/dynamicgraphs/action?  
action\_id=execute-analysis

Table 5-210 URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

Table 5-211 Request body parameters

| Parameter     | Mandatory | Type   | Description          |
|---------------|-----------|--------|----------------------|
| algorithmName | Yes       | String | Algorithm name       |
| dynamicRange  | Yes       | String | Temporal parameters  |
| parameters    | Yes       | String | Algorithm parameters |

Table 5-212 dynamicRange parameters

| Parameter | Mandatory | Type            | Description                         |
|-----------|-----------|-----------------|-------------------------------------|
| start     | Yes       | Date or integer | Start time of the temporal analysis |

| Parameter  | Mandatory | Type            | Description                               |
|------------|-----------|-----------------|---|
| end        | Yes       | Date or integer | End time of the temporal analysis         |
| time_props | Yes       | Object          | Time properties for the temporal analysis |

**Table 5-213** time\_props parameters

| Parameter | Mandatory | Type   | Description                     |
|-----------|-----------|--------|---------------------------------|
| stime     | Yes       | String | Property name of the start time |
| etime     | Yes       | String | Property name of the end time   |

**Table 5-214** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| source    | Yes       | String  | Start vertex ID  |
| k         | No        | Integer | Distance from the target vertices to the start vertex. The value range is [1, 100] and the default value is 3. |
| directed  | No        | Boolean | Search direction. The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> .            |

## Response Parameters

**Table 5-215** Parameters in a response

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| errorMessage | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |



| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| status    | No        | String | Returned job status for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . If the query fails, the field is empty. |
| data      | No        | Json   | Algorithm execution result. If the query fails, the field is empty.   |

## Example Request

Specify a source vertex ID to search for associated vertices. The algorithm name is **temporal\_bfs**, the start time of dynamic analysis is **`\${startTime}**, and the end time is **`\${endTime}**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName":"temporal_bfs",
  "dynamicRange":{
    "start":"${startTime}",
    "end":"${endTime}",
    "time_props":{"stime":"${property(start_time)}","etime":"${property(start_time)}"}
  },
  "parameters":{
    "source":""
  }
}
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",
  "jobType": 0
}
```

**Status code: 400**

Example response for a failed request

```
{
  "errorMessage":"${errorMessage}",
  "errorCode":"GES.8301"
}
```

## Status Code

| Return Value     | Description              |
|------------------|--------------------------|
| 400 Bad Request  | Request error            |
| 401 Unauthorized | Authorization failed     |
| 403 Forbidden    | No operation permissions |

| Return Value              | Description           |
|---------------------------|-----------------------|
| 404 Not Found             | No resources found    |
| 500 Internal Server Error | Internal server error |
| 503 Service Unavailable   | Service unavailable   |

### 5.1.7.3 Temporal Paths

#### Function

This API is used to execute the temporal paths algorithm based on input parameters.

Note: Only one temporal path that meets the conditions is returned between two vertices.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/dynamicgraphs/action?  
action\_id=execute-analysis

**Table 5-216** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-217** Request body parameters

| Parameter     | Mandatory | Type                                | Description   |
|---------------|-----------|-------------------------------------|---|
| algorithmName | Yes       | String                              | Algorithm name  |
| parameters    | Yes       | <a href="#">parameters</a> Object   | Algorithm parameters. For details, see the parameter description of each algorithm. |
| dynamicRange  | Yes       | <a href="#">dynamicRange</a> Object | Temporal parameters   |

**Table 5-218** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| source    | Yes       | String  | Source vertex ID   |
| targets   | Yes       | String  | Target vertex ID set. The value is in CSV format. IDs are separated by commas (,), for example, <b>Alice,Nana</b> . The quantity cannot be greater than 100000. The default value is <b>1000</b> .   |
| directed  | No        | Boolean | Whether to consider the edge direction. The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> .   |
| k         | No        | Integer | Maximum depth. The value ranges from 1 to 100. The default value is <b>3</b> .   |
| strategy  | No        | String  | Algorithm policy. The value can be <b>shortest</b> , <b>foremost</b> , or <b>fastest</b> .<br>(Note: <b>fastest</b> is not supported currently.)<br>The default value is <b>shortest</b> . <ul style="list-style-type: none"> <li>• <b>shortest</b>: Runs the <b>shortest temporal paths</b> algorithm to return the temporal path with the shortest distance.</li> <li>• <b>foremost</b>: Runs the <b>foremost temporal paths</b> algorithm to return the temporal path that reaches the target node as early as possible.</li> <li>• <b>fastest</b>: Runs the <b>fastest temporal paths</b> algorithm to return the temporal path that takes the shortest time.</li> </ul> |

**Table 5-219** dynamicRange

| Parameter  | Mandatory | Type                              | Description                           |
|------------|-----------|-----------------------------------|---------------------------------------|
| start      | Yes       | Date/ Integer                     | Start time for temporal analysis      |
| end        | Yes       | Date/ Integer                     | End time for temporal analysis        |
| time_props | Yes       | <a href="#">time_props</a> Object | Time properties for temporal analysis |

**Table 5-220** time\_props

| Parameter | Mandatory | Type   | Description                     |
|-----------|-----------|--------|---------------------------------|
| stime     | Yes       | String | Name of the start time property |
| etime     | Yes       | String | Name of the end time property   |

## Response Parameters

**Table 5-221** Response parameters

| Parameter    | Mandatory | Type    | Description   |
|--------------|-----------|---------|---|
| errorMessage | No        | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.  |
| errorCode    | No        | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.   |
| jobId        | No        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |
| jobType      | No        | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

Specify a source vertex ID to search for associated vertices. The algorithm name is **temporal\_paths**, the start time of dynamic analysis is **1646092800**, the end time is **1646170716**, and the source vertex ID is **Person00014**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName":"temporal_paths",
  "dynamicRange":{"
    "start":1646092800,
    "end":1646170716,
    "time_props":
```

```

    {"stime":"startTime","etime":"endtime"}
  },
  "parameters":{
    "source":" Person00014",
    "targets":"Person00055,Person00058,Person00052,Person00061,Person00060,Place00032,Place00016,Place00026,Place00015,Place00043",
    "strategy":"shortest",
    "directed":true
  }
}

```

## Example Response

### Status code: 200

Example response for a successful request

```

{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}

```

### Status code: 400

Example response for a failed request

```

{
  "errorMessage":"graph [demo] is not found",
  "errorCode":"GES.8402"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## 5.1.8 Path APIs

### 5.1.8.1 Querying Path Details

#### Function

This API is used to query the path details. All possible paths will be listed.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/paths/action?action\_id=query-detail

**Table 5-222** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Example Request

Query path details. The paths to be queried are **Ray**, **Lethal Weapon**, and **Alice**. Do not set the query direction.

```
post http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail
{
  "paths":[
    [
      "Ray",
      "Lethal Weapon",
      "Alice"
    ]
  ],
  "directed":false
}
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Request Parameters

**Table 5-223** Request body parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| paths     | Yes       | List    | Set of paths to be queried   |
| directed  | No        | Boolean | Whether the querying path is directional or non-directional:<br><b>true:</b> directional<br><b>false:</b> non-directional<br>default=false |

## Response Parameters

**Table 5-224** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | Object | Query results.<br>This parameter is left blank when the query fails.  |

**Table 5-225** data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| outputs   | Object | Query results containing the paths  |
| paths     | List   | Collection of paths that contain detailed vertex and edge information, in JSONArray format<br><b>NOTE</b><br>In the returned paths: <ul style="list-style-type: none"> <li>If the vertex does not exist, the corresponding position is {}.</li> <li>If there is no edge between vertices, the corresponding position is {"edges": []}.</li> </ul> |

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "paths": [
        [
          {
```

```

    "id": "Ray",
    "label": "user",
    "properties": {
      "Name": ["Ray"],
      "Gender": ["M"],
      "Age": ["18-24"],
      "Occupation": ["college/grad student"],
      "Zip-code": ["90241"]
    }
  },
  {
    "edges": [
      {
        "source": "Ray",
        "target": "Lethal Weapon",
        "index": "1",
        "label": "rate",
        "properties": {
          "Score": [2],
          "Datetime": ["2000-11-22 19:16:16"]
        }
      }
    ]
  }
],
{
  "id": "Alice",
  "label": "user",
  "properties": {
    "Name": ["Alice"],
    "Gender": ["F"],
    "Age": ["25-34"],
    "Occupation": ["academic/educator"],
    "Zip-code": ["79928"]
  }
}
]
}
}
}
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8107"
}

```

**Status Code**

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |



## Error Code

See [Error Code](#).

### 5.1.8.2 Querying Tree Details

#### Function

After you input a vertex, a tree starting from the vertex is output, including nodes and edges in the tree, and information about reachable paths. Parameters can be the direction (out, in, and out and in), maximum number of hops, and edge properties to be filtered.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=query-tree

**Table 5-226** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

```
Post /ges/v1.0/{project_id}/graphs/test/action?action_id=query-tree
{
  "executionMode": "sync",
  "rootId": "0",
  "maxDepth": 20,
  "maxVertices": 5000,
  "maxEdges": 10000,
  "maxDegree": 1000,
  "direction": "both",
  "labelFilters": [
    "edge",
    "default"
  ],
  "withVertex": true,
  "withEdge": true
}
```

## Request Parameters

**Table 5-227** Request body parameters

| Parameter     | Mandatory | Type          | Description  |
|---------------|-----------|---------------|--|
| executionMode | No        | String        | <ul style="list-style-type: none"> <li>• <b>sync</b>: synchronous</li> <li>• <b>async</b>: asynchronous</li> </ul> The default value is <b>async</b> , indicating asynchronous response.                 |
| rootId        | Yes       | String        | ID of the root node  |
| maxDepth      | Yes       | Integer       | Tree depth, that is, the maximum length of a path. The value ranges from <b>1</b> to <b>20</b> .   |
| direction     | Yes       | String        | Possible values:<br><b>in</b> : Query incoming edges.<br><b>out</b> : Query outgoing edges.<br><b>both</b> : Query edges in both directions.   |
| labelFilters  | No        | Array of Json | Whether to filter paths by label. The default value is <code>[]</code> .   |
| maxVertices   | Yes       | Integer       | Maximum number of vertices that can be obtained. The value ranges from <b>0</b> to <b>200,000</b> (excluding <b>0</b> ).   |
| maxEdges      | Yes       | Integer       | Maximum number of edges that can be obtained. The value ranges from <b>0</b> to <b>200,000</b> (excluding <b>0</b> ).  |
| maxDegree     | No        | Long          | This parameter is used to access a super node. When the number of unidirectionally connected edges of an accessed node is greater than <b>maxDegree</b> , the next hop of the node will not be accessed. |
| withVertex    | No        | Boolean       | Whether to output the vertex data. The default value is <b>true</b> .  |
| withEdge      | No        | Boolean       | Whether to output the edge data. The default value is <b>true</b> .  |

 **NOTE**

In asynchronous mode, vertex and edge data is output at the same time. Pagination is not supported.

## Response Parameters

**Table 5-228** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | Object | Query results. For details, see <a href="#">Table 5-229</a> and <a href="#">Table 5-230</a> .<br>This parameter is left blank when the query fails.   |

**Table 5-229** Synchronous data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| vertices  | List | Vertex result set. If no corresponding vertices are found, the value of <b>vertices</b> is empty. |
| edges     | List | Edge result set. If no corresponding edges are found, the value of <b>edges</b> is empty.         |

**Table 5-230** Asynchronous data parameter description

| Parameter | Type    | Description   |
|-----------|---------|---|
| jobId     | String  | ID of the algorithm execution job. This parameter is left blank when the request fails. |
| jobType   | Integer | Job type. This parameter is left blank when the request fails.                          |

## Example Response

- Synchronous call  
**Status code: 200**  
Example response (successful request)

```
Http Status Code: 200
{
```

```

"data": {
  "edges": [
    {
      "index": "0",
      "label": "edge",
      "properties": {
        "_type": [ "ACTED_IN" ],
        "role": [ "Neo" ]
      },
      "source": "546",
      "target": "0"
    },
    ...
  ],
  "vertices": [
    {
      "id": "546",
      "label": "node",
      "properties": {
        "_labels": [ "Actor" ],
        "d": [ "" ],
        "name": [ "Keanu Reeves" ],
        "personId": [ "keanu" ],
        "title": [ "" ],
        "year": [ "" ]
      }
    },
    ...
  ]
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [test] is not found",
  "errorCode": "GES.8808"
}

```

- Asynchronous call

**Status code: 200**

Example response (successful request)

```

{
  "jobId": "aca63a57-55f9-4c6e-932f-bcd6b87f7ab4",
  "jobType": 1
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "executionMode is not correct, it should be sync or async",
  "errorCode": "GES.8808"
}

```

## Status Codes

| Return Value     | Description           |
|------------------|-----------------------|
| 400 Bad Request  | Request error.        |
| 401 Unauthorized | Authorization failed. |

| Return Value              | Description               |
|---------------------------|---------------------------|
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.8.3 Repeat-query API

#### Function

This API is used to cyclically perform traversal query based on certain exit conditions.

For example, the Gremlin statement is as follows:

```
g.V('a','b').repeat(out('c')).times(2).emit().path()
g.V('label_1').repeat(out('label_2')).emit()
```

#### URL

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=repeat-query

**Table 5-231** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

### NOTE

The number of elements in each traversal cannot exceed 100 million.

**Table 5-232** Request body parameters

| Parameter description | Mandatory | Type          | Description   |
|-----------------------|-----------|---------------|---|
| executionMode         | No        | String        | <b>sync</b> : synchronous; <b>async</b> : asynchronous. The default value is <b>async</b> , indicating asynchronous mode.   |
| vertices              | Yes       | Array of Json | List of IDs of source vertices to be queried  |
| edges                 | No        | Array of Json | List of edges to be queried. Either this parameter or <b>vertices</b> is selected. For details, see <a href="#">edges element formats</a> .   |
| repeat                | Yes       | Array of Json | Filter criteria for repeat queries. Each element in the array corresponds to a filter. For details about the format, see <a href="#">repeat element formats</a> .   |
| until                 | No        | Array of Json | <b>while/do looping</b> mode. Stops traversing the filter condition list. Each element in the array corresponds to the condition to be terminated at each layer in the repeat. The filter type must be the same as that of each layer in the repeat. For example, the first layer is <code>vertex_filter</code> . For details about the format, see <a href="#">repeat element formats</a> .  |
| times                 | No        | Integer       | Maximum number of steps. The default value is <b>5</b> and the maximum value is <b>20</b> .   |
| query_type            | No        | String        | Possible values are <b>Default</b> , <b>SimpleSubgraph</b> , <b>SimpleEdges</b> , and <b>Path</b> . <ul style="list-style-type: none"> <li>• <b>Default</b> returns the query content. This is the default mode.</li> <li>• <b>SimpleSubgraph</b> returns the entire submap, including only the ID and label information of vertices and edges.</li> <li>• <b>SimpleEdges</b> returns all edges, including only the ID and label information of the edges.</li> <li>• <b>Path</b> returns the path information, that is, the set of paths.</li> </ul> |
| emit                  | No        | Boolean       | Whether to return all elements. The default value is <b>true</b> . This parameter is available only when <b>query_type</b> is set to <b>Default</b> .   |

**Table 5-233 edges** element formats

| Parameter | Mandatory | Type   | Description                             |
|-----------|-----------|--------|---|
| source    | Yes       | String | Source vertex ID                        |
| target    | Yes       | String | Target vertex ID                        |
| index     | Yes       | String | Indexes of edges in the source edge set |

**Table 5-234 repeat** element formats

| Parameter     | Mandatory | Type        | Description   |
|---------------|-----------|-------------|---|
| operator      | Yes       | String      | Query type. s are as follows: inV: ingress; outV: egress. bothV: ingress point.   |
| vertex_filter | No        | Json String | It takes effect on the vertex of the next hop. For details, see <a href="#">Table property_filter element format</a> .      |
| edge_filter   | No        | Json String | Indicates the edge between point-to-point vertices. For details, see <a href="#">Table property_filter element format</a> . |

**Table 5-235 repeat** element formats

| Parameter     | Mandatory | Type        | Description  |
|---------------|-----------|-------------|--|
| vertex_filter | No        | Json String | This parameter is optional when <b>operator</b> is set to <b>inV</b> , <b>outV</b> , or <b>bothV</b> . For details about the format, see <a href="#">property_filter element formats</a> . |

**Table 5-236 property\_filter** element formats

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| leftvalue | No        | String | Left value of the string type. For details, see <a href="#">leftvalue element formats</a> . |

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| properties | Yes       | String | <p>Filter type. Possible values are = (equal to), != (not equal to), &lt; (less than), &lt;= (less than or equal to), &gt; (greater than), &gt;= (greater than or equal to), &amp; (and),   (or), <b>HAS/HASNOT</b> (whether a property exists), and <b>CONTAIN/NOTCONTAIN</b> (whether the property value contains the right value).</p> <p><b>SUBSET</b>: The right value is a subset of the attribute value. <b>IN/NOTIN</b>: Check whether the left value and right value overlap.</p> <p><b>PREFIX</b>: The right value is the prefix of the left value. <b>FUZZY</b>: Fuzzy match. <b>REGEX</b>: Regular expression match.</p> <p><b>SUBSTRING</b>: The right value is the substring of the left value. <b>CISUBSTRING</b>: The substring is case-insensitive.</p> |
| rightvalue | Yes       | String | <p>Right value. For details about the format, see <a href="#">rightvalue element formats</a>.</p>  |

**Table 5-237 leftvalue element formats**

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| label_name      | No        | String | <p>If <b>label</b> is used as the filter criterion, set this parameter to <b>labelName</b> and <b>value</b> of <b>rightvalue</b> to the label name.</p>         |
| property_name   | No        | String | <p>If <b>property</b> is used as the filter criterion, set this parameter to the property name and <b>value</b> of <b>rightvalue</b> to the property value.</p> |
| ID              | No        | String | <p>If the vertex ID is filtered, this parameter is optional.</p>  |
| property_filter | No        | String | <p>If <b>predicate</b> is set to <b>&amp;</b> or <b> </b>, <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b>.</p>                  |
| degree          | No        | String | <p>Direction of vertex degree filtering statistics. This parameter is optional. The value can be <b>both</b>, <b>in</b>, or <b>out</b>.</p>                     |



**Table 5-238** rightvalue element formats

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| value           | Yes       | String | If <b>label</b> is used as the filter criterion, the value is the label name. If <b>property</b> is used as the filter criterion, the value is the property name. |
| property_filter | No        | String | If <b>predicate</b> is set to <b>&amp;</b> or <b> </b> , <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b> .                         |

**Table 5-239** predicate application scenarios

| predicate                                   | label_name | id  | property_name | Nested Filtering |
|---|------------|-----|---------------|------------------|
| <b>&amp;</b>                                | No         | No  | No            | Yes              |
| <b> </b>                                    | No         | No  | No            | Yes              |
| HAS/HASNOT                                  | No         | No  | Yes           | No               |
| <b>=/!</b><br><b>=/&lt;/&lt;=&gt;/&gt;=</b> | Yes        | Yes | Yes           | No               |

## Response Parameters

**Table 5-240** Response body parameters

| Parameter description | Type   | Description   |
|-----------------------|--------|---|
| errorMessage          | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode             | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data                  | Object | Query results This parameter is left blank when the query fails.  |

**Table 5-241** data parameter description

| Parameter description | Type   | Description      |
|-----------------------|--------|------------------|
| source                | String | Source vertex ID |
| target                | String | Target vertex ID |
| index                 | String | Edge index       |
| label                 | String | Edge label       |

## Example Request

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=repeat-query
{
  "executionMode": "async",
  "repeat": [
    {
      "operator": "outV",
      "vertex_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "rate"
          }
        }
      }
    }
  ],
  "until": [
    {
      "vertex_filter": {
        "property_filter": {
          "leftvalue": {
            "property_name": "movieid"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "1"
          }
        }
      }
    }
  ],
  "emit": false,
  "vertices": [
    "tr_10","tr_11"
  ]
}
```

### NOTE

The preceding request is equivalent to the following Gremlin statement:  
`g.V('tr_10','tr_11').repeat(out().hasLabel('rate')).until(has('movieid','1')).dedup()`

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "data":{
    "edges":[
      {
        "index":"1",
        "source":"tr_1",
        "label":"rate",
        "properties":{
          "Rating":[
            0
          ],
          "Datetime":[
            ""
          ]
        }
      },
      {
        "index":"199998",
        "source":"tr_1",
        "label":"rate",
        "properties":{
          "Rating":[
            0
          ],
          "Datetime":[
            ""
          ]
        }
      },
      {
        "index":"200000",
        "source":"tr_1",
        "label":"rate",
        "properties":{
          "Rating":[
            0
          ],
          "Datetime":[
            ""
          ]
        }
      }
    ]
  }
}
```

**Status code: 400**

Example response for a failed request

```
HttpStatusCode:400
{
  "errorMessage":"graph [tesdt_117] is not found",
  "errorCode":"GES.8806"
}
```

**Status Codes**

| Response Code             | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.1.9 Graph Statistics APIs

### 5.1.9.1 Querying General Information About a Graph

#### Function

This API is used to query the general information about a graph, such as the numbers of vertices and edges.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/summary?  
label\_details={labelDetails}

**Table 5-242** URI parameters

| Parameter    | Mandatory | Type    | Description   |
|--------------|-----------|---------|---|
| project_id   | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name   | Yes       | String  | Graph name  |
| labelDetails | No        | Boolean | Whether to return the number of vertices and edges under each label. The default value is <b>false</b> . If this parameter is set to <b>true</b> , the numbers of vertices and edges under each label are returned. |

#### Request Parameters

None

## Response Parameters

**Table 5-243** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. This parameter is left blank when the request fails.   |

**Table 5-244** data parameter description

| Parameter    | Type    | Description   |
|--------------|---------|---|
| vertexNum    | Integer | Number of vertices in a graph   |
| edgeNum      | Integer | Number of edges in a graph  |
| labelDetails | Object  | Numbers of vertices and edges under each label. To properly display this parameter, create vertex and edge indexes based on <a href="#">Table 5-245</a> . |

**Table 5-245** Description of each element in **labelDetails** when the execution is successful

| Parameter     | Type   | Description   |
|---------------|--------|---|
| labelInVertex | Object | Number of vertices under each label. If the number of vertices under a label is 0, the label is not displayed.<br><br>To include this parameter in the response, create an index by referring to <a href="#">Creating an Index</a> . During index creation, set <b>indexType</b> to <b>GlobalComposite-VertexIndex</b> , set <b>hasLabel</b> to <b>true</b> , and leave <b>indexProperty</b> blank. |

| Parameter    | Type   | Description  |
|--------------|--------|--|
| labelInEdge  | Object | Number of edges under different labels. If the number of edges under a label is 0, the label is not displayed.<br><br>To include this parameter in the response, create an index by referring to <a href="#">Creating an Index</a> . During index creation, set <b>indexType</b> to <b>GlobalCompositeEdgeIndex</b> , set <b>hasLabel</b> to <b>true</b> , and leave <b>indexProperty</b> blank. |
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>  |

## Example Request

Query general information about a graph, such as the numbers of vertices and edges. The value **true** indicates that the numbers of vertices and edges of different labels are returned.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details=true
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

- Response example 1 of a successful request (The numbers of vertices and edges under each label are returned.)

```
Http Status Code: 200
{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "labelInVertex": {
        "movie": 46,
        "user": 100
      },
      "labelInEdge": {
        "default": 450,
        "rate": 1209
      }
    }
  }
}
```

```

    },
    "edgeNum": 1659
  }
}

```

- Response example 2 of a successful request (The numbers of vertices and edges under each label fail to be returned.)

Http Status Code: 200

```

{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "errorMessage": "Label index in vertices is not found.Label index in edges is not found.",
      "errorCode": "GES.8017"
    },
    "edgeNum": 1659
  }
}

```

- Example response 3 of a successful request (Only the number of vertices under each label is returned.)

Http Status Code: 200

```

{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "errorMessage": "Label index in edges is not found.",
      "labelInVertex": {
        "movie": 46,
        "user": 100
      },
      "errorCode": "GES.8017"
    },
    "edgeNum": 1659
  }
}

```

### Status code: 400

#### Example response (failed request)

Http Status Code: 400

```

{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8001"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.9.2 Querying the Graph Version

#### Function

This API is used to query the graph version.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/version

**Table 5-246** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

#### Response Parameters

**Table 5-247** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |



| Parameter | Type   | Description   |
|-----------|--------|---|
| version   | String | Query results. This parameter is left blank when the request fails. |

## Example Request

Query the graph version.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/version
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "version": "2.0.0"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

## Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

## 5.1.10 Graph Operation APIs

### 5.1.10.1 Importing a Graph

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=import-graph

**Table 5-248** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-249** Request body parameters

| Parameter       | Mandatory | Type   | Description  |
|-----------------|-----------|--------|--|
| edgesetPath     | No        | String | Edge file directory or name  |
| edgesetFormat   | No        | String | Format of the edge data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.  |
| vertexsetPath   | No        | String | Vertex file directory or name  |
| vertexsetFormat | No        | String | Format of the vertex data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.  |
| schemaPath      | No        | String | OBS path of the metadata file of the new data  |
| logDir          | No        | String | Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes. |
| parallelEdge    | No        | Object | Repetitive edge processing   |

| Parameter   | Mandatory | Type      | Description  |
|-------------|-----------|-----------|--|
| action      | No        | String    | <p>Processing mode of repetitive edges. The value can be <b>allow</b>, <b>ignore</b>, or <b>override</b>. The default value is <b>allow</b>.</p> <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul>   |
| ignoreLabel | No        | Boolean   | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: Indicates that the repetitive edge definition contains the label. That is, the &lt;source vertex, target vertex, label&gt; indicates an edge.</li> </ul> |
| delimiter   | No        | Character | <p>Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).</p>  |
| trimQuote   | No        | Character | <p>Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.</p>   |

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| offline       | No        | Boolean | Whether offline import is selected. The value is <b>true</b> or <b>false</b> , and the default value is <b>false</b> . <ul style="list-style-type: none"> <li>• <b>true</b>: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import.</li> <li>• <b>false</b>: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.</li> </ul> |
| obsParameters | Yes       | Object  | OBS parameters  |

**Table 5-250** obsParameters parameters

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | String | User AK     |
| secretKey | Yes       | String | User SK     |

 **NOTE**

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

## Response Parameters

**Table 5-251** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId     | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                |

## Example Request

Import a graph. The edge file directory is **testbucket/demo\_movie/edges/** and the edge data set format is CSV; the vertex file directory is **testbucket/demo\_movie/vertices/** and the vertex data set format is CSV; the OBS path of the metadata file of the new data is **testbucket/demo\_movie/incremental\_data\_schema.xml** and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true
  },
  "delimiter": ",",
  "trimQuote": "\"",
  "offline": true,
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorCode": "GES.8013",
  "errorMessage": "graph [movie2] is not found"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.10.2 Exporting a Graph

#### NOTE

If you choose to export CSV files to your local host, the files are opened using the spreadsheet software by default. You are advised to open the files in a text editor. If the data contains special characters such as plus signs (+), minus signs (-), equal signs (=), and at signs (@), the data will be parsed into formulas by the software. To ensure system security, pay attention to the following when opening such files:

1. Do not select **Enable Dynamic Data Exchange Server Launch (not recommended)**.
2. Do not select **Enable** or **Yes** if a dialog box indicating a security issue is displayed.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=export-graph

**Table 5-252** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-253** Request body parameters

| Parameter       | Mandatory | Type   | Description                           |
|-----------------|-----------|--------|---------------------------------------|
| graphExportPath | Yes       | String | OBS path to which a graph is exported |
| edgeSetName     | Yes       | String | Name of the exported edge data set    |
| vertexSetName   | Yes       | String | Name of the exported vertex data set  |
| schemaName      | Yes       | String | Name of the exported metadata file    |
| obsParameters   | Yes       | String | OBS parameters                        |
| accessKey       | Yes       | String | User AK                               |
| secretKey       | Yes       | String | User SK                               |

 **NOTE**

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

## Response Parameters

**Table 5-254** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                |

## Example Request

- Example request**  
 POST `http://{{SERVER_URL}}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph`

```
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge.csv",
  "vertexSetName": "set_vertex.csv",
  "schemaName": "set_schema.xml",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorCode": "GES.8011",
  "errorMessage": "graph [movie2] is not found"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.10.3 Clearing a Graph

#### URI

POST `/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph`



**Table 5-255** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-256** Request body parameter

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| clearMetadata | No        | Boolean | Whether to clear schema data. The default value is <b>false</b> . |

## Response Parameters

**Table 5-257** Response body parameters

| Parameter    | Mandatory | Description   |
|--------------|-----------|---|
| errorMessage | No        | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | No        | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | No        | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                |

## Example Request

Clear a graph by deleting its schema data.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph
{
  "clearMetadata": true
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorCode": "GES.8012",
  "errorMessage": "graph [movie2] is not found"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.1.11 Subgraph Operation APIs

### 5.1.11.1 Querying a Subgraph

#### Function

This API is used to query the subgraphs formed by the entered vertices and edges between the vertices.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/subgraphs/action?  
action\_id=query

**Table 5-258** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-259** Request body parameter

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| vertices  | Yes       | String | Vertex ID array of the subgraph<br><b>NOTE</b><br>The maximum number of vertices that can be entered is 100,000. If the number of vertices exceeds this limit, an error is reported. |

#### Response Parameters

**Table 5-260** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| errorCode | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>  |
| data      | Object | The <b>data</b> field is contained when the query is successful, and the <b>data</b> field contains the subgraph query result. <p><b>NOTE</b><br/>The maximum number of subgraph edges that can be returned is 100,000. If the number of edges exceeds this limit, an error is reported.</p> |

## Example Request

Query the subgraphs formed by the entered vertices and edges between the vertices. The subgraph vertex IDs are **Ray**, **Ella**, and **Lethal Weapon**.

```
{
  "vertices":[
    "Ray",
    "Ella",
    "Lethal Weapon"
  ]
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data":{
    "vertices":[
      {
        "id":"Ray",
        "label":"user",
        "properties":{
          "Name":["Ray"],
          "Gender":["M"],
          "Age":["18-24"],
          "Occupation":["college/grad student"],
          "Zip-code":["90241"]
        }
      },
      {
        "id":"Ella",
        "label":"user",
        "properties":{
          "Occupation":["other or not specified"],
          "Name":["Era"],
          "Zip-code":["94402"],
          "Gender":["F"],
          "Age":["25-34"]
        }
      }
    ]
  }
}
```

```

    }
  ],
  "edges":[
    {
      "source":"Ray",
      "target":"Lethal Weapon",
      "index":"1",
      "label":"rate",
      "properties":{
        "Score":[2],
        "Datetime":["2000-11-22 19:16:16"]
      }
    },
    {
      "index":"0",
      "source":"Ella",
      "label":"rate",
      "properties":{
        "Score":[5],
        "Datetime":["2000-11-23 02:30:29"]
      },
      "target":"Lethal Weapon"
    },
    {
      "index":"5",
      "source":"Ella",
      "label":"friends",
      "properties":{},
      "target":"Ray"
    }
  ]
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

## 5.1.11.2 Executing an Algorithm on a Subgraph

### Introduction

This API is used to adjust the subgraph creation type based on the input and executes an algorithm on the generated subgraph.

### URL

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/subgraphs/action?  
action\_id=execute-algorithm

### Parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

### Request Parameters

**Table 5-261** Request body parameters

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| algorithmName   | Yes       | String | Algorithm name<br>Available values are as follows: <ul style="list-style-type: none"> <li>connected_component</li> <li>kcore</li> </ul>   |
| parameters      | Yes       | JSON   | Algorithm parameters <ul style="list-style-type: none"> <li><a href="#">connected_component</a></li> <li><a href="#">Kcore</a></li> </ul> |
| subgraphCreator | Yes       | Object | Subgraph parameters For details, see <a href="#">subgraphCreator parameters</a> .   |

**Table 5-262 subgraphCreator parameters**

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| name       | No        | String | Type of the subgraph creator. Currently, only <b>filtered</b> is available. |
| parameters | Yes       | JSON   | The parameter format varies according to the name of the subgraph creator.  |

**Table 5-263 Parameters when name=filtered**

| Parameter     | Mandatory | Type   | Description               |
|---------------|-----------|--------|---------------------------|
| vertex_filter | No        | String | Vertex filtering criteria |
| edge_filter   | No        | String | Edge filtering criteria   |

## Response Parameters

**Table 5-264 Response body parameters**

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |
| jobType      | Integer | Task type. This parameter is left blank if the request fails.   |

## Example Request

Execute a specific algorithm on a generated subgraph. The algorithm name is **connected\_component**, and the subgraph generator type is **filtered**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?
action_id=execute-algorithm
{
  "algorithmName":"connected_component",
  "subgraphCreator": {
    "name": "filtered",
    "parameters": {
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "PHYSICAL_LINK"
          }
        }
      }
    }
  },
  "parameters":{
    "num_thread":4
  }
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage":"graph [demo] is not found",
  "errorCode":"GES.8402"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |



| Return Value            | Description          |
|-------------------------|----------------------|
| 503 Service Unavailable | Service unavailable. |

## Error Codes

See [Error Codes](#).

## 5.1.12 Job Management APIs

### 5.1.12.1 Querying Job Status on the Service Plane

#### Function

This API is used to query the execution status of a job. After asynchronous APIs such as those for querying vertices and edges or executing algorithms are used, job IDs are returned. You can use the job ID to query the execution status of a job.

#### URI

GET `/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=offset&limit=limit`

**Table 5-265** URI parameters

| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| project_id | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| job_id     | Yes       | String  | Job ID   |
| offset     | No        | Integer | Offset of a query. The default value is <b>0</b> .   |
| limit      | No        | Integer | Maximum number of records that can be queried. The default value is <b>100000</b> .                      |

## Response Parameters

**Table 5-266** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>  |
| status       | String | Returned job status after the query is successful. Possible values: <ul style="list-style-type: none"> <li><b>pending</b></li> <li><b>running</b></li> <li><b>success</b></li> <li><b>failed</b></li> </ul> This parameter is left blank when the query fails. |
| data         | Object | Algorithm execution result. This parameter is left blank when the query fails.   |

**Table 5-267** data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| vertices         | List    | Vertex-associated algorithm result                         |
| edges            | List    | Edge-associated algorithm result                           |
| outputs          | Object  | Other results  |
| data_return_size | Integer | Number of records returned after a query                   |
| data_offset      | Integer | Result offset of a query                                   |
| data_total_size  | Integer | Total amount of result data generated by asynchronous jobs |

## Example Request

Query the execution status of a job. The query offset is **0**, and the maximum number of returned results is **2**.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=0&limit=2
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "data_return_size": 2,
      "vertices": [
        {
          "id": "Sarah",
          "label": "user",
          "properties": {
            "Occupation": [
              "other or not specified"
            ],
            "Name": [
              "Sarah"
            ],
            "Zip-code": [
              "55105"
            ],
            "Gender": [
              "F"
            ],
            "Age": [
              "18-24"
            ]
          }
        },
        {
          "id": "Sidney",
          "label": "user",
          "properties": {
            "Occupation": [
              "writer"
            ],
            "Name": [
              "Sidney"
            ],
            "Zip-code": [
              "85296"
            ],
            "Gender": [
              "M"
            ],
            "Age": [
              "18-24"
            ]
          }
        }
      ]
    }
  }
}
```

```

    "data_offset": 0,
    "data_total_size": 19
  }
},
"status": "success"
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.12.2 Canceling a Job**

**Function**

This API is used to cancel a job that has been submitted.

 **NOTE**

Only jobs returned by the following APIs can be canceled: exporting a graph, importing a graph, querying vertices that meet filter criteria, querying edges that meet filter criteria, edge filtering query, [Filtered-query V2](#), executing an algorithm, and adding an index.

**URI**

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs/{job\_id}

**Table 5-268** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| job_id     | Yes       | String | Job ID   |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

For details, see the URI parameters.

## Response Parameters

**Table 5-269** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

Cancel a job that has been submitted.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{ }
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "can not find job to cancel, id is 9440a7ebXXXXXXXXXXXXXXXXXXXX2d079a67001679122",
  "errorCode": "GES.8303"
}
```

### Status Code

| Return Value              | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

### Error Code

See [Error Code](#).

### 5.1.12.3 Exporting Job Execution Results to Files

#### Function

This API is used to export the execution result (**result**) of an asynchronous job (**jobId**) to a file.

- The following algorithms are supported:
  - PageRank, PersonalRank, and Pixie
  - Louvain, Label Propagation, and Connected Component
  - K-Core
  - SSSP, Shortest Path (including Time Window Shortest Path), Shortest Path of Vertex Sets, All Shortest Paths, and n Paths
  - Triangle Count, Cluster Coefficient, Degree Correlation, and Closeness
  - Link Prediction
  - Betweenness, edge\_betweenness, and od\_betweenness

- The following queries are supported:
  - [Cypher Operation APIs](#)
  - [Querying Vertices That Meet Filter Criteria](#)
  - [Querying Edges That Meet Filter Criteria](#)

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs/{job\_id}/action?  
action\_id=export-result

**Table 5-270** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| job_id     | Yes       | String | ID of the job corresponding to the response  |

## Request Parameters

**Table 5-271** Request body parameters

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| exportPath    | Yes       | String  | Export path   |
| fileName      | No        | String  | Name of the exported file   |
| obsParameters | Yes       | String  | OBS authentication parameters. For details, see <a href="#">Table 5-272</a> .   |
| paginate      | No        | Object  | Pagination parameter. By default, pagination is disabled for the asynchronous task export API. For details, see <a href="#">Table 5-273</a> .   |
| erase         | No        | Boolean | Specifies whether to delete the original job after the export. The value can be true or false. The default value is true, indicating that the job is deleted and resources are released by default. |

**Table 5-272** obsParameters parameter description

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | string | AK          |
| secretKey | Yes       | string | SK          |

 **NOTE**

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

**Table 5-273** Elements in paginate

| Parameter       | Mandatory | Type | Description   |
|-----------------|-----------|------|---|
| enable          | No        | Bool | Whether to enable pagination. The default value is <b>false</b> . To enable pagination, set this parameter to <b>true</b> . |
| rowCountPerFile | No        | Int  | Maximum number of rows in each file when graphs are exported by page. The default value is <b>1000000</b> .                 |
| numThread       | No        | Int  | Number of concurrent threads when graphs are exported by page. The default value is <b>8</b> .                              |
| maxSizePerFile  | No        | Int  | Maximum size of each file when execution results are exported by page, in bytes.  |

- paginate parameter description
  - a. When pagination is enabled, **fileName** in the request body indicates the directory name, and the directory is used to store pagination files. When pagination is disabled, **fileName** indicates the file name. Before the export, ensure that the path the file name points to is empty so that the existing data on OBS will not be overwritten during the export.
  - b. If the value of **numThread** is greater than the number of vCPUs used by the GES graph instance, the parameter is set to the number of vCPUs.
  - c. The **rowCountPerFile** value affects the number of actually used threads. That is, when the ratio of the result set size to **rowCountPerFile** is less than **numThread**, the ratio is used as the value of **numThread**.
  - d. If the request is canceled by the user, the data uploaded to OBS will not be deleted. For details about the API for canceling jobs, see [Canceling a Job](#).



- When the pagination function is enabled, the file name is named as follows:  
If **enable** in the **paginate** parameter is set to **true**, **fileName** indicates a directory. Files in the directory are named using the combination of the thread ID and file number, and are separated by a period (.). For example, for 3.2 million data records, the examples of the first and last file names in different configurations are as follows:

|                                    |            |             |           |           |           |
|------------------------------------|------------|-------------|-----------|-----------|-----------|
| rowCountPerFile                    | 100,000    | 100,000     | 1 million | 1 million | 5 million |
| numThread                          | 2          | 48          | 2         | 5         | 2         |
| Threads Actually Used              | 2          | 32          | 2         | 4         | 1         |
| Files Generated by a Single Thread | 16         | 1           | 2         | 1         | 1         |
| First File Name                    | 00.000.csv | 00.000.csv  | 00.00.csv | 00.00.csv | 00.00.csv |
| Last File Name                     | 01.015.csv | 031.000.csv | 01.01.csv | 03.00.csv | 00.00.csv |

To export the data of a ten-billion-vertex graph on multiple nodes at the same time, use the host ID as the prefix of the file name.

## Response Parameters

**Table 5-274** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job.<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Querying Job Status on the Service Plane (1.0.0)</a> .     |

## Example Request

- Example request 1: Export the execution results of an asynchronous job to an OBS file. The export path is **demo\_movie/**. The name of the exported file is **louvain**. After the export, the original job is deleted.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result
{
  "exportPath": "demo_movie/",
  "fileName": "louvain",
  "erase": true,
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

- Example request 2: Export the execution result of an asynchronous job to an OBS file. The export path is **demo\_movie/**. The name of the exported file is **louvain**. By default, pagination is disabled. When data is exported by page, the maximum number of rows in each file is 100,000.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result
{
  "exportPath": "demo_movie/",
  "fileName": "louvain",
  "paginate": {
    "enable": true,
    "numThread": 2,
    "rowCountPerFile": 100000
  },
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

### NOTE

Currently, Cypher statements can only be used to export common value types, such as attribute values, numbers, and strings, but not composite value types (such as lists and maps), vertices, or edges. For example:

- The results of the following statements can be exported:  
match (n) return id(n) limit 10  
match (n) return n.age, n.occupation  
match (n)-[r]->(m) return n.Rating limit 10  
unwind [1,2,3] as p return p
- The exported CSV file contains null values or blank lines because the results of the following statements contain objects or compound values:  
return [1,2,3], {a:1}  
match (n) return n limit 10  
match (n)-[r]->(m) return r limit 10

## Example Response

**Status code: 200**

Example response (successful request)

```
HttpStatusCode: 200
{
```

```
"jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

**Status code: 400**

Example response (failed request)

```
HttpStatusCode: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8011"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**Exporting the Result in CSV File**

- The following is an example of the algorithm execution result, for example, content of **Louvain.csv**:

```
# modularity: 0.4269691347613425,
#community_num: 4,
#runtime: 0.003784,
#data_total_size: 34
#community:
1,1
2,1
...
```

- The following is an example Cypher query result:

- Example 1

Query statement:

```
match (n:user)-[r]->(m:movie) return id(n),n.Name, n.Occupation, n.Age,r.Score,m.Title
```

Result:

```
#data_total_size:1209
#data_return_size:1209
#data_offset:0
#records:
Vivian, artist, 25-34, 5, Lethal Weapon
Vivian, Artist, 25-34, 4, Raising Arizona
Mercedes, K-12 student, Under 18, 3, Lethal Weapon
```

```
Mercedes, K-12 student, Under 18, 3, The Rock
...
```

- Example 2

Query statement:

```
match (n)-->(m) where id(n)='Vivian' return labels(m),count(*)
```

Result:

```
#data_total_size:2
#data_return_size:2
#data_offset:0
#records:
user,5
movie,2
```

### 5.1.12.4 Querying the Job List

#### Function

After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned.

#### URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?
limit={limit}&offset={offset}
```

Table 5-275 URI parameters

| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| project_id | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| offset     | No        | Integer | Offset of a query. The default value is <b>0</b> .   |
| job_id     | Yes       | String  | ID of the job corresponding to the response  |
| limit      | No        | Integer | Maximum number of records that can be queried. The default value is <b>100000</b> .                      |

#### Request Parameters

For details, see the URI parameters.

## Response Parameters

**Table 5-276** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Query result. If the query is successful, the value is <b>success</b> . If the query fails, the value is <b>failed</b> .  |
| jobs         | Object | Job status list stored in the system. If execution succeeds, this parameter is contained in the response. <a href="#">Table 5-277</a> describes the structure of a single jobs field.                         |

**Table 5-277** Job status structure

| Parameter | Type   | Description   |
|-----------|--------|---|
| jobId     | String | Job name.   |
| request   | Object | Request content, including the command, URL, and body.                              |
| status    | String | Job status. The value can be <b>pending</b> , <b>running</b> , or <b>complete</b> . |

### Example Request

Query the job list and return the job ID and status of each job.

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}
```

### Example Response

**Status code: 200**

Example response for a successful request

```
{
  "jobs": [
```

```

{
  "jobId": "527e0c2e-57ee-4c23-8cf7-baff8049c6b9001234245",
  "rawRequest": "{\"graphName\":\"ges_yiwan_jimengke\", \"offline\":false, \"edgesetPath\":\"/user/ges/data/ranking_edge.csv\", \"vertexsetPath\":\"/user/ges/data/movies_vertex_new.csv\", \"delimiter\": \"\\\", \"trimQuote\": \"\\\"\", \"vertexsetFormat\": \"csv\", \"parallelEdge\": { \"ignoreLabel\": true, \"action\": \"allow\" }, \"edgesetFormat\": \"csv\" }\",
  "taskType": "ImportGraph",
  "canStop": false,
  "progress": "100",
  "files": [
    {
      "edgeFiles": [
        {
          "fileName": "/user/ges/data/ranking_edge.csv",
          "totalLines": 1659,
          "startTime": 1707201116130344,
          "successfulLines": 1659,
          "endTime": 1707201116149714,
          "status": "success",
          "failedLines": 0
        }
      ]
    },
    {
      "vertexFiles": [
        {
          "fileName": "/user/ges/data/movies_vertex_new.csv",
          "totalLines": 146,
          "startTime": 1707201115842346,
          "successfulLines": 146,
          "endTime": 1707201115876634,
          "status": "success",
          "failedLines": 0
        }
      ]
    }
  ],
  "startTime": "2024-02-06 14:31:55",
  "endTime": "2024-02-06 14:31:56",
  "status": "success"
},
{
  "jobId": "9d43e7a1-bfc5-4990-b494-01c31098523e001234245",
  "rawRequest": "{\"graphName\":\"ges_yiwan_jimengke\", \"offline\":false, \"edgesetPath\": \"\\\", \"vertexsetPath\": \"\\\", \"delimiter\": \"\\\", \"schemaPath\": \"/user/ges/data/movie_schema.xml\", \"trimQuote\": \"\\\"\", \"vertexsetFormat\": \"csv\", \"parallelEdge\": { \"ignoreLabel\": true, \"action\": \"allow\" }, \"edgesetFormat\": \"csv\" }\",
  "taskType": "ImportGraph",
  "canStop": false,
  "progress": "100",
  "files": [
    {
      "schemaFiles": [
        {
          "totalLabels": 4,
          "fileName": "/user/ges/data/movie_schema.xml",
          "failedLabels": 0,
          "startTime": 1707201113260909,
          "successfulLabels": 4,
          "endTime": 1707201113369960,
          "status": "success"
        }
      ]
    }
  ],
  "startTime": "2024-02-06 14:31:45",
  "endTime": "2024-02-06 14:31:53",
  "status": "success"
}

```

```
  ],  
  "jobCount": 2  
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "graph : movidde not exist",  
  "errorCode": "GES.8000",  
  "result": "failed"  
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.13 Custom Operations APIs**

**5.1.13.1 Performing Custom Operations**

**Function**

This API is used to perform a specified custom operation.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-operation

**Table 5-278** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-279** Request body parameters

| Parameter | Mandatory | Type   | Description                                  |
|-----------|-----------|--------|--|
| api       | Yes       | String | Available APIs used by the custom operation. |
| command   | Yes       | String | Command executed by the custom operation.    |

## Response Parameters

**Table 5-280** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | json   | Execution result of the custom operation. This parameter is left blank if the request fails.  |

## Example Request

Perform a specified custom operation. The API used for the custom operation is **gremlin**, and the command is `{"command": "g.V('1')"`.



```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-operation
{
  "api": "gremlin",
  "command": "{\"command\": \"g.V('1')\"}"
}
```

## Example Response

**Status code: 200**

Example response (successful request)

```
{
  "data":{
    "vertices":[
      {
        "id":"1",
        "label":"movie",
        "properties":{
          "genres":[
            "Comedy"
          ],
          "movieid":[
            1
          ],
          "title":[
            "Airplane! (1980)"
          ]
        }
      }
    ],
    "runtime":0.126476598
  }
}
```

**Status code: 400**

Example response (failed request)

```
Internal Server Error
{
  "errorCode":"GES.8814",
  "errorMessage":"Unsupported API."
}
```

## Status Code

| Return Value              | Description              |
|---------------------------|--------------------------|
| 400 Bad Request           | Request error.           |
| 401 Unauthorized          | Authentication failed.   |
| 403 Forbidden             | No operation permission. |
| 404 Not Found             | No resources found.      |
| 500 Internal Server Error | Internal service error.  |
| 503 Service Unavailable   | Service unavailable.     |

## Error Code

See [Error Code](#).

## 5.1.14 Cypher Operation APIs

### 5.1.14.1 Executing Cypher Queries

#### Function

Cypher is a widely used declarative graph database query language. It can be used to query data in GES and returns results. Graph statistics are used in Cypher implementation. Currently, the label-based vertex and edge indexes are used during Cypher query and compilation. To use Cypher normally, create indexes by referring to [Cypher Prerequisites](#).

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-cypher-query

**Table 5-281** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

Execute a Cypher query. The Cypher statement is **match (n) return n limit 1**. The returned results are in the format that each element corresponds to a field in the row.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "parameters": {},
    "resultDataContents": ["row"],
    "includeStats": false
  }]
}
```

## Request Parameters

**Table 5-282** Request body parameter

| Parameter  | Mandatory | Type | Description   |
|------------|-----------|------|---|
| statements | Yes       | List | Statement group that contains one or more statements. The <a href="#">statements parameters</a> table describes the format of each element. |

**Table 5-283** statements parameters

| Parameter          | Mandatory | Type           | Description  |
|--------------------|-----------|----------------|--|
| statement          | Yes       | String         | Cypher statement   |
| parameters         | Yes       | Object         | Cypher statement parameters, which are used for parameterized queries. By default, this field is left blank.<br>For details, see <a href="#">parameterized queries</a> .   |
| resultDataContents | No        | String or List | Format of the returned result. You can set one or more formats. Available values are <b>row</b> , <b>graph</b> , and <b>raw</b> (added in version 2.2.27).   |
| includeStats       | No        | Boolean        | Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.   |
| runtime            | No        | String         | Executor type. The value can be <b>map</b> or <b>slotted</b> . The default value is <b>map</b> .<br><b>NOTE</b> <ol style="list-style-type: none"> <li>1. The slotted executor is supported since version 2.3.14.</li> <li>2. Compared with the map executor, the slotted executor completes more statement data flow analysis in the plan generation phase of statements. In most cases, it executes faster while requiring less memory.</li> </ol> |

| Parameter              | Mandatory | Type   | Description  |
|------------------------|-----------|--------|--|
| executionMode (2.2.23) | No        | String | Execution mode. Set this parameter to <b>sync</b> for synchronous execution and to <b>async</b> for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For details about how to obtain the query result in asynchronous mode, see <a href="#">Querying Job Status on the Service Plane</a> . |
| limit (2.2.23)         | No        | Int    | Maximum number of results of the asynchronous query. This parameter is valid only when <b>executionMode</b> is <b>sync</b> . The default value is <b>100000</b> .  |

 NOTE

- You can add the **explain** and **profile** prefixes before the statement to display the query plan.
  - **explain** displays only the query plan but does not execute the statement.
  - **profile** displays the query plan and executes the statement.
- In asynchronous mode (**executionMode** is **async**), cypher query results can be exported to CSV files (GES 2.3.4 or later supports this function). For details, see [Exporting Job Execution Results to Files](#). Currently, the following values can be returned:
  1. Vertex and edge single-value properties, vertex and edge IDs, and group counts.
  2. The current version does not support exporting object types. Objects are converted to null values in the CSV file.

## Response Parameters

**Table 5-284** Response body parameters

| Parameter | Type | Description  |
|-----------|------|--|
| results   | List | Each element of the list is the return result of a Cypher statement.               |
| errors    | List | Each element in the list contains the code and message information in string form. |

**Table 5-285** Elements of the results parameter

| Parameter | Type | Description              |
|-----------|------|--------------------------|
| columns   | List | Name of a returned field |

| Parameter       | Type    | Description  |
|-----------------|---------|--|
| data            | List    | Returned data value. Each element indicates a record.  |
| stats           | Object  | Addition, deletion, and modification statistics  |
| plan            | Object  | If the Cypher statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed. The profile feature is supported since version 2.3.12. |
| jobId(2.3.10)   | String  | Asynchronous job ID if the request is executed asynchronously  |
| jobType(2.3.10) | Integer | Type of the asynchronous job if the request is executed asynchronously   |

**Table 5-286** Elements of the data parameter

| Parameter   | Type   | Description  |
|-------------|--------|--|
| row         | List   | Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when <b>resultDataContents</b> is empty or contains <b>row</b> . |
| meta        | List   | Type of each field in a row. This parameter is displayed only when <b>resultDataContents</b> is empty or contains <b>row</b> .   |
| graph       | Object | Information returned in graph format. This parameter is displayed only when <b>resultDataContents</b> contains <b>graph</b> .  |
| raw(2.2.27) | List   | Information returned in raw format. This parameter is displayed only when <b>resultDataContents</b> contains <b>raw</b> .  |

**Table 5-287** stats elements in a response

| Parameter        | Type    | Description                               |
|------------------|---------|---|
| contains_updates | Boolean | Whether data is modified during the query |
| edges_created    | Integer | Number of created edges                   |
| edges_deleted    | Int     | Number of deleted edges                   |

| Parameter        | Type    | Description                             |
|------------------|---------|---|
| labels_set       | Integer | Number of labels that have been set     |
| properties_set   | Integer | Number of properties that have been set |
| vertices_created | Integer | Number of created vertices              |
| vertices_deleted | Integer | Number of deleted vertices              |

## Example Response

### Status code: 200

Example response for a successful request (synchronous call)

Http Status Code: 200

```
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ],
  "stats": {
    "contains_updates": false,
    "edges_created": 0,
    "edges_deleted": 0,
    "labels_set": 0,
    "properties_set": 0,
    "vertices_created": 0,
    "vertices_deleted": 0
  }
},
"errors": []
}
```

### Status code: 200

Example response for a successful request (asynchronous call)

```
Http Status Code: 200
{
  "results": [
    {
      "columns": [
        "jobId",
        "jobType"
      ],
      "jobId": "b64a5846-e306-4f87-b0f1-d595ee2a9910",
      "jobType": 1,
      "data": [
        {
          "row": [
            "b64a5846-e306-4f87-b0f1-d595ee2a9910",
            1
          ],
          "meta": [
            null,
            null
          ]
        }
      ]
    }
  ],
  "errors": []
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "results": [],
  "errors": [
    {
      "code": "GES.8904",
      "message": "Label index in vertices is not found."
    }
  ]
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

### 5.1.14.2 Cypher Prerequisites

The current Cypher query compilation process uses the label-based vertex and edge indexes. To use Cypher normally, use the [index creation API](#) to create indexes.

- Example command for adding a vertex label index. The index name is **cypher\_vertex\_index**, and the index type is global vertex index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "cypher_vertex_index",
  "indexType": "GlobalCompositeVertexIndex",
  "hasLabel": "true",
  "indexProperty": []
}
```

- Example command for adding an edge label index. The index name is **cypher\_edge\_index**, and the index type is global edge index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "cypher_edge_index",
  "indexType": "GlobalCompositeEdgeIndex",
  "hasLabel": "true",
  "indexProperty": []
}
```

 **NOTE**

1. You do not need to create the indexes if the graph is of the 100-million-edge, 1-billion-edge, or 10-billion-edge types. (The GES version must be 2.3.6 or later.)
2. If you need to create indexes, you must create both two indexes (vertex label index and edge label index) at the same time to use Cypher for query.
3. If your graph already has a vertex index or an edge index whose **hasLabel** is **true** and **indexProperty** is empty, you do not need to create the vertex index or edge index again.
4. The API for creating an index is asynchronous. To check whether the index is successfully created, use the [API for querying job status](#).
5. If fine-grained permissions are used, a member account must have the schema permission and the read permission on all labels (including the default label **\_\_DEFAULT\_\_**) to create indexes. Otherwise, after an index is created, you need to use an account with the required permissions to send **call db.schema()** to update the compiler cache of Cypher queries.

### 5.1.14.3 Basic Operations and Compatibility

#### Basic Operations

| Operation                                  | Cypher Statement  |
|--|---|
| Querying vertices                          | match (n) return n  |
| Querying edges                             | match (n)-[r]->(m) return n, r, m                         |
| Querying paths                             | match (n:user)-[r]->(m:movie)-->(s:series) return n,r,m,s |
| Querying vertices based on filter criteria | match(n:user) where n.userid>=5 return n                  |



| Operation                | Cypher Statement   |
|--------------------------|--|
| Grouping and aggregating | match(n:movie) return n.genres, count(*)                                 |
| Deduplicating            | match(n:movie) return distinct n.genres                                  |
| Sorting                  | match(n:movie) return n order by n.movieid                               |
| Creating a vertex        | create (n:user{userid:1}) return n                                       |
| Creating an edge         | match (n:user{userid:15}),(m:movie{movieid:10}) create (n)-[r:rate]->(m) |
| Deleting a vertex        | match (n:user{userid:1}) delete n  |
| Modifying labels         | match (n:user{userid:1}) set n:movie return n                            |
| Modifying properties     | match (n:user{userid:1}) set n.userid=2 return n                         |

## Compatibility to Cypher

### 1. Cypher clauses

Cypher implements a couple of clauses. You can combine clauses to implement different query semantics, including vertex and edge filtering, multi-hop query, sorting and deduplication, and grouping and aggregation. Currently, GES supports the Cypher clauses listed in the following table.

**Table 5-288** Supported Cypher clauses

| Clause         | Support             | Example  |
|----------------|---------------------|--|
| match          | Partially supported | match (n:movie) return n   |
| optional match | Partially supported | optional match (n)-->(m) where id(n)='1' return m                              |
| return         | Supported           | return [1,2,3] as p  |
| with           | Supported           | match (n) with labels(n) as label, count(*) as count where count > 10 return * |
| where          | Supported           | match (n:movie) where n.movieid > 10 return n                                  |
| order by       | Supported           | match (n:movie) return n order by n.genres                                     |
| skip           | Supported           | match (n:movie) return n order by n.genres skip 5                              |
| limit          | Supported           | match (n:movie) return n order by n.genres skip 5 limit 10                     |

| Clause          | Support   | Example  |
|-----------------|-----------|--|
| create          | Supported | create (n:user{ _ID_: 'Jack' }) return n   |
| delete          | Supported | match (n:movie)<-[r]-(m:user) delete r   |
| set             | Supported | match (n:user{userid:0}) set n.gender='M'<br>return n  |
| call procedures | Supported | call db.schema()   |
| unwind          | Supported | unwind [1, 2, 3] as p return p   |
| union           | Supported | match (n:movie) return id(n) union match (n:user) return id(n)<br><b>NOTE</b><br>Union is available for graphs smaller than 10 billion edges only. |

 **NOTE**

1. Currently, merge and foreach operations are not supported. Cypher statements cannot add or delete indexes.
  2. GES metadata is not schema-free, and the vertex and edge label properties are strictly restricted. Therefore, the remove operation is not supported.
  3. The order by clause does not support sorting of the list type. When Cardinality of the property value is not single, the sorting result is unknown.
- Available items for the match clause

| Item           | Description   | Example Clauses   | Earliest Version Required |
|----------------|---|---|---------------------------|
| Vertex pattern | Patterns for matching vertex with specified labels, properties, and IDs.  | match (n:movie{title:'hello'})<br>match (n) where id(n)='xx'  | 2.2.16                    |
| Edge pattern   | Patterns for matching directional and non-directional edges with specified labels and properties. Specified IDs of both start and end vertices are supported. | match (n)-[r] -> (m)<br>match (n)-[r]- (m)<br>match (n)-[r:rate{Rating:1}] - (m)<br>match (n)-[r]- (m) where id(n)='x'and id(m)='y' | 2.2.16                    |

|                              |  |                            |        |
|------------------------------|--|----------------------------|--------|
| Path                         | Anonymous paths  | match (n)-[r]->(m)-->(s)   | 2.2.16 |
|                              | Named paths  | match p=(n)-[r]->(m)-->(s) | 2.2.19 |
| Multiple patterns            | You can enter multiple patterns after <b>match</b> and separate them with commas (.).<br>match (n)-[r]->(m), (m)-->(s)   |                            | 2.2.16 |
| Multi-match                  | You can enter multiple match clauses. You can use <b>with</b> to connect multiple clauses.<br>match (n)-[r]->(m) with m match (m)-->(s)  |                            | 2.2.16 |
| Variable-length path pattern | Patterns for matching variable-length paths starting with a specified vertex.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' return p<br>match p=(n{title:'name'})-[r*1..3]->(m) return p |                            | 2.2.19 |
|                              | Traversal conditions for matching variable-length paths.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' and all (x in nodes(p) where x.prop='value1') return p                            |                            | 2.2.28 |
|                              | Both start vertex and end vertex of a variable-length path can be specified.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='y' return p  |                            | 2.3.9  |
|                              | Deduplication by end vertex is not supported:<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='yy' return distinct m   |                            | None   |

## 2. Parameterized queries

Cypher supports parameterized queries. Numeric and string values in a query statement are extracted and converted to parameters for faster compilation, improving the query speed.

There are some examples of parameterized queries:

### - Example 1

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user) where n.occupation = $occupation return n",
    "parameters": {
      "occupation" : "artist"
    },
    "resultDataContents": ["row"]
  }]
}
```

### - Example 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user {`Zip-code`:'98133'}) set n = $props return n",
    "parameters": {
      "props": {
        "gender": "M",
        "age": "56+"
      }
    },
    "resultDataContents": ["row"]
  }
}
```

 **NOTE**

There are some scenarios where parameterized queries are not supported. The following syntax is not valid:

1. Using **\$param** to search by property key and value. For example, **match (n) where n.\$param = 'something'**
2. Using **\$code** for vertex and edge labels. For example, **match (n:user) set n:\$code**
3. Supported data types

Currently, GES supports 10 data types: char, char\_array, float, double, Boolean, long, Integer, date, enum, and string. Both Boolean and numeric types are supported in the Cypher syntax. The following table lists the mapping between other types and Cypher data types.

**Table 5-289** Mapping between data types of GES and Cypher

| GES        | Cypher   | Description   |
|------------|----------|---|
| char       | String   | -   |
| char_array | String   | -   |
| string     | String   | -   |
| enum       | String   | The Cypher syntax does not provide enum-related syntax. During Cypher query, an enum is converted to a string. When Cypher is used to set properties, values that are not in the enumeration list fail to be set. |
| date       | Temporal | Currently, Cypher dates can be converted into GES dates, but Cypher date functions cannot be used for inputting a date.   |

**Table 5-290** Special types supported by Cypher

| Type | Supported | Example                     |
|------|-----------|-----------------------------|
| Node | Yes       | match (n) return n limit 10 |

| Type              | Supported | Example   |
|-------------------|-----------|---|
| Relations<br>hip  | Yes       | match (n)-[r]->(m) return r limit 10                |
| List              | Yes       | return [1,2,3] as li                                |
| Map               | Yes       | match (n)-->(m) return {start:id(n), end:id(m)}     |
| Path              | Yes       | match p=(n1)-[:friends*1..2]-(n2) return p limit 10 |
| Point,<br>Spatial | No        | -   |

 **NOTE**

For the special types listed above, only the List type can be used to match multi-value properties in GES. Other types cannot be used in a set statement for setting the value of a property.

4. Vertex ID compatibility

- Cypher does not provide the syntax for setting the ID when a vertex is added. In GES, however, an ID of the string type is required to uniquely identify a vertex. To use the Cypher syntax in GES, add `_ID_` to specify the ID of a vertex in the create statement. For example, the `create(n{_ID_:'123456'})` statement creates a vertex whose ID is 123456.
- If the ID is not specified, a random ID is generated for the vertex.

 **NOTE**

The `_ID_` identifier is supported only in the create statement. The match and set clauses do not support the `_ID_` identifier. In the match clause, you can use the `id()` function to obtain the vertex ID.

### 5.1.14.4 Supported Expressions, Functions, and Procedures

#### Expression

Cypher queries support multiple expressions and can be used in combination to form various filter criteria. Currently, the following expressions are supported.

| Operation Type     | Expression | Example   |
|--------------------|------------|---|
| Logical operations | and        | match (n:user) where n.age='Under 18' and n.gender='F' return n |
|                    | or         | match(n:user) where n.`Zip-code`='22181' or n.userid=6 return n |
|                    | not        | match(n:movie) where not n.genres contains 'Drama' return n     |

| Operation Type                | Expression   | Example  |
|-------------------------------|--|--|
| Null value judgment           | is null  | match (n) where n.userid is null return n  |
|                               | is not null  | match (n) where n.userid is not null return n  |
| Comparison calculation        | >, >=, <, <=, =, <>                                      | match(n:user) where n.userid >= 5 return n   |
| Arithmetic operators (2.3.10) | +, -, *, /, %, ^   | return (1+3)%3   |
| String comparisons            | starts with  | match(n:movie) where n.genres starts with 'Comedy' return n  |
|                               | ends with  | match(n:movie) where n.genres ends with 'Drama' return n   |
|                               | contains   | match(n:movie) where n.genres contains 'Drama' return n  |
| List-related operation        | in   | match(n:student) where 'math' in n.courses return n  |
|                               | []   | match(n:user) return n['userid']<br>with [1, 2, 3, 4] as list return list[0]<br>with [1, 2, 3, 4] as list return list[0..1]<br>match p=(n)-->(m) return [x in nodes(p) where x.gender='F' id(x)]       |
| Date expressions (2.3.10)     | .year, .month, .day, .hour, .minute, .second, .dayOfWeek | Year, month, and day of a specific date: with '2000-12-27 23:44:41' as strVal with datetime(strVal) as d2 return d2.year, d2.month, d2.day, d2.hour, d2.minute, d2.second, d2.dayOfWeek, d2.ordinalDay |

 **NOTE**

The where clause in Cypher queries does not support regular expressions.

## Function

Cypher supports the following functions for grouping, aggregation, and vertex and edge operations:

1. Aggregate  
Aggregate functions **count** and **collect** are supported.

| Function | Earliest Version Supported | Description                          | Example   |
|----------|----------------------------|--------------------------------------|---|
| count    | 2.2.17                     | Returns the total number of results. | match (n) return count(*)<br>match (n) return count(n.userid) |
| collect  | 2.2.17                     | Collects results into a list.        | match (n:movie) return n.genres, collect(n) as movieList      |
| sum      | 2.3.3                      | Returns the sum of values.           | unwind [1, 2.0, 3] as p<br>return sum(p)                      |
| avg      | 2.3.3                      | Returns the average of values.       | unwind [1, 2.0, 3] as p<br>return avg(p)                      |
| min      | 2.3.3                      | Returns the minimum value.           | unwind [1, 2.0, 3] as p<br>return min(p)                      |
| max      | 2.3.3                      | Returns the maximum value.           | unwind [1, 2.0, 3] as p<br>return max(p)                      |

2. Regular functions

Based on the types of input parameters, regular functions are classified into vertex and edge functions, path functions, list functions, and value functions.

**Table 5-291** Vertex and edge functions

| Function | Earliest Version Supported | Description                       | Example   |
|----------|----------------------------|-----------------------------------|---|
| id       | 2.2.16                     | Obtains the ID of a vertex.       | match (n) return id(n)                            |
| labels   | 2.2.16                     | Obtains labels of a vertex.       | match (n) return labels(n)                        |
| type     | 2.2.16                     | Obtains the label of an edge.     | match(n)-[r]->(m) return type(r)                  |
| degree   | 2.2.26                     | Obtains the degree of a vertex.   | match (n) where id='Vivian'<br>return degree(n)   |
| inDegree | 2.2.26                     | Obtains the indegree of a vertex. | match (n) where id='Vivian'<br>return inDegree(n) |

| Function  | Earliest Version Supported | Description                          | Example  |
|-----------|----------------------------|--------------------------------------|--|
| outDegree | 2.2.26                     | Obtains the outdegree of a vertex.   | match (n) where id='Vivian'<br>return outDegree(n) |
| startNode | 2.3.10                     | Obtains the start vertex of an edge. | match (n)-[r]->(m) return<br>startNode(r)          |
| endNode   | 2.3.10                     | Obtains the end vertex of an edge.   | match (n)-[r]->(m) return<br>endNode(r)            |

**Table 5-292** Path functions

| Function      | Earliest Version Supported | Description                             | Example  |
|---------------|----------------------------|---|--|
| nodes         | 2.2.19                     | Obtains the list of vertices on a path. | match p=(n)-[:friends*1..2]->(m) return nodes(p)         |
| relationships | 2.2.19                     | Obtains the list of edges on a path.    | match p=(n)-[:friends*1..2]->(m) return relationships(p) |
| length        | 2.2.19                     | Obtains the path length.                | match p=(n)-[:friends*1..2]->(m) return length(p)        |

**Table 5-293** List functions

| Function | Earliest Version Supported | Description                          | Example                                     |
|----------|----------------------------|--------------------------------------|---|
| head     | 2.3.10                     | Obtains the first element of a list. | with [1,2,3,4] as list return<br>head(list) |
| last     | 2.3.10                     | Obtains the last element of a list.  | with [1,2,3,4] as list return<br>last(list) |
| size     | 2.3.10                     | Obtains the list length.             | with [1,2,3,4] as list return<br>size(list) |



| Function | Earliest Version Supported | Description       | Example                            |
|----------|----------------------------|-------------------|------------------------------------|
| range    | 2.3.10                     | Generates a list. | return range(1,5),<br>range(1,5,2) |

**Table 5-294** Value functions

| Function    | Earliest Version Supported | Description                               | Example  |
|-------------|----------------------------|---|--|
| toString    | 2.2.21                     | Converts a value to a string.             | match (n) where toString(labels(n)) contains 'movi' return n |
| toUpper     | 2.2.26                     | Converts a string into uppercase letters. | match (n:movie) return toUpper(n.title)                      |
| toLowerCase | 2.2.26                     | Converts a string into lowercase letters. | match (n:movie) return toLower(n.title)                      |
| toInteger   | 2.2.29                     | Converts a string to an int number.       | with '123' as p return toInteger(p)                          |
| toLong      | 2.2.29                     | Converts a string to a long number.       | with '123' as p return toLong(p)                             |
| toFloat     | 2.2.29                     | Converts a string to a float number.      | with '123.4' as p return toFloat(p)                          |
| toDouble    | 2.2.29                     | Converts a string to a double number.     | with '123.4' as p return toDouble(p)                         |
| toBoolean   | 2.2.29                     | Converts a string to a bool value.        | with 'true' as p return toBoolean(p)                         |
| size        | 2.2.29                     | Obtains the string length.                | with 'GES' as p return size(p)                               |
| subString   | 2.3.10                     | Truncates a part of a string.             | return subString('abc', 1),<br>subString('abcde', 1,2)       |

|          |        |  |  |
|----------|--------|--|--|
| coalesce | 2.3.10 | Obtains the first non-null value of the parameters.          | return coalesce(null, '123')                   |
| trim     | 2.3.11 | Removes whitespace characters on both sides of a string.     | return trim(' hello ')                         |
| lTrim    | 2.3.11 | Removes whitespace characters on the left side of a string.  | return lTrim(' hello')                         |
| rTrim    | 2.3.11 | Removes whitespace characters on the right side of a string. | return rTrim('hello ')                         |
| reverse  | 2.3.11 | Returns a string with the characters in reverse order.       | return reverse('hello')                        |
| left     | 2.3.11 | Obtains several characters from the left side of a string.   | with 'hello' as p return left(p, 3)            |
| right    | 2.3.11 | Obtains several characters from the right side of a string.  | with 'hello' as p return right(p, 3)           |
| replace  | 2.3.11 | Replaces a string.   | with 'hello' as p return replace(p, 'll', 'o') |
| split    | 2.3.11 | Splits a string.   | with 'hello' as p return split(p, 'e')         |

**Table 5-295** Mathematical functions

| Function | Earliest Version Supported | Description                                    | Example                       |
|----------|----------------------------|--|-------------------------------|
| floor    | 2.3.10                     | Rounds a number down to the nearest integer.   | return floor(4.1)             |
| ceil     | 2.3.10                     | Rounds a number up to the nearest integer.     | return ceil(4.1)              |
| round    | 2.3.14                     | Round  | return round(3.4), round(3.5) |
| abs      | 2.3.14                     | Absolute value function                        | return abs(-3),abs(-3.5)      |
| sin      | 2.3.14                     | Sine function                                  | return sin(pi()/2)            |
| cos      | 2.3.14                     | Cosine function                                | return cos(0),cos(pi()/2)     |
| tan      | 2.3.14                     | Tangent function                               | return tan(pi()/4)            |
| acos     | 2.3.14                     | Inverse cosine function                        | return acos(1)                |
| asin     | 2.3.14                     | Inverse sine function                          | return asin(0)                |
| atan     | 2.3.14                     | Inverse tangent function                       | return atan(1)                |
| cot      | 2.3.14                     | Cotangent function                             | return cot(pi()/4)            |
| radians  | 2.3.14                     | Converts degree to radian.                     | return radians(180)           |
| degrees  | 2.3.14                     | Converts radian to degree.                     | return degrees(pi())          |
| pi       | 2.3.14                     | Returns the approximate value of Pi ( $\pi$ ). | return pi()                   |

**Table 5-296** Date and time functions

| Function       | Earliest Version Supported | Description   | Example                                 |
|----------------|----------------------------|---|---|
| datetime(val)  | 2.3.10                     | Returns the time based on the timestamp.                        | return datetime(1688696395)             |
| datetime()     | 2.3.14                     | Obtains the current time (valid only for read statements).      | return datetime()                       |
| timestamp(val) | 2.3.10                     | Returns the timestamp based on the time string.                 | return timestamp('2023-07-07 02:20:42') |
| timestamp()    | 2.3.14                     | Obtains the current timestamp (valid only for read statements). | return timestamp()                      |
| localDatetime  | 2.3.14                     | Converts a time or timestamp to a local time string.            | return localDatetime(timestamp())       |

**Table 5-297** Predicate functions

| Function | Earliest Version Supported | Description   | Example                |
|----------|----------------------------|---|------------------------|
| all      | 2.2.19                     | If all elements meet the expression, <b>true</b> is returned. | all (x in p where x>1) |
| any      | 2.2.19                     | If any element meets the expression, <b>true</b> is returned. | any (x in p where x>1) |

| Function | Earliest Version Supported | Description  | Example                   |
|----------|----------------------------|--|---------------------------|
| none     | 2.2.19                     | If all elements cannot meet the expression, <b>true</b> is returned. | none (x in p where x>1)   |
| single   | 2.2.19                     | If only one element meets the expression, <b>true</b> is returned.   | single (x in p where x>1) |

**Table 5-298** Algorithm expressions

| Function         | Earliest Version Supported | Description                                      | Example   |
|------------------|----------------------------|--|---|
| shortestPath     | 2.3.2                      | Returns the shortest path between two vertices.  | The following statement returns the shortest path between the given vertices <b>n</b> and <b>m</b> . The direction is m to n, and the edge label is <b>rate</b> :<br>with n,m, shortestPath((n)-[:rate*]-(m)) as p return p |
| allShortestPaths | 2.3.2                      | Returns all shortest paths between two vertices. | The following statement returns all shortest paths between the given vertices <b>n</b> and <b>m</b> :<br>with n,m, allShortestPaths((n)-[*]-(m)) as p return p  |

 **NOTE**

Degree functions, path operation functions, and algorithm expressions are not available if the graph size is ten-billion-edge.

## Procedure

Currently, GES supports the following procedures.

| Procedure  | Statement                                   |
|--|---|
| Obtaining graph pattern information                                      | call db.schema()                            |
| Obtaining vertex labels  | call db.labels()                            |
| Obtaining edge labels  | call db.relationshipTypes()                 |
| Querying the Cypher statements that are being executed                   | call dbms.listQueries()                     |
| Terminating a Cypher statement based on <b>queryId</b>                   | call dbms.killQuery('queryId')              |
| Querying indexes   | call db.indexes()                           |
| Full-text indexing for querying vertices that meet the search conditions | call db.index.fulltext.queryNodes()         |
| Full-text indexing for querying edges that meet the conditions           | call db.index.fulltext.queryRelationships() |

 **NOTE**

- Full-text indexes support six types of queries: prefix, wildcard, regexp, fuzzy, match, and combine. To use full-text indexes, you need to call the API for creating a full-text index.
- Function and procedure names are case sensitive and must be in lower camel case.
- Example of a full-text index query request  
 POST http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-cypher-query
 

```

      {
        "statements": [
          {
            "statement": "call db.index.fulltext.queryNodes('combine', {title:'1977'}) yield node, score
            return node, score skip 1 limit 10",
            "resultDataContents": [
              "row"
            ],
            "parameters": {}
          }
        ]
      }
      
```
- Parallel edge processing policy  
 When using Cypher to add edges, you can add duplicate edges. Duplicate edges are two edges with the same source vertex and target vertex.
- How to add an edge without a label  
 When you use a Cypher statement to add an edge, set the label of the edge to the default value **\_\_DEFAULT\_\_**. For example, **create ()-[r:\_\_DEFAULT\_\_]->() return r**.

## Querying the Schema Structure Using Cypher

- Function

You can call the **db.schema ()** function using Cypher to query the structure of a generated schema (obtained from OBS).

- Query statement
  - Name: Schema structure query
  - Statement: **call db.schema()**
  - Note:

If you did not call the API for generating the schema structure, the returned schema file contains all labels.

If you have called the API for generating the schema structure, this API returns the labels as the vertices and the relationships between the labels as edges.

## 5.1.15 Filtered-query API

### Function

This API filters the k-hop process layer by layer, and lists the k hop vertices or edges that meet the filtering criteria.

### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=filtered-query

**Table 5-299** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

### Request Parameters

 **NOTE**

If **executionMode** is **sync**, the number of returned vertices cannot exceed 100,000.

**Table 5-300** Request body parameters

| Parameter     | Mandatory | Type          | Description   |
|---------------|-----------|---------------|---|
| executionMode | No        | String        | <ul style="list-style-type: none"> <li>• <b>sync</b>: synchronous</li> <li>• <b>async</b>: asynchronous</li> </ul> The default value is <b>sync</b> , indicating synchronous response.  |
| vertices      | Yes       | Array of Json | List of IDs of source vertices to be queried  |
| query_type    | No        | String        | Possible values are <b>Default</b> , <b>AllVertices</b> , <b>SimpleEdges</b> , <b>Path</b> . <ul style="list-style-type: none"> <li>• <b>Default</b> indicates the default mode, that is, the <math>k</math> hop is returned.</li> <li>• <b>AllVertices</b> returns details about all vertices within <math>k</math> hops.</li> <li>• <b>SimpleEdges</b> returns all edges within <math>k</math> hops, contain only the ID and label information of the edges.</li> <li>• <b>Path</b> returns the path information, that is, the set of paths.</li> </ul> |
| by            | No        | Array of Json | Specified output field. This parameter is valid only when <b>query_type</b> is set to <b>Default</b> or <b>AllVertices</b> . Currently, only one layer is supported. If no field is specified, all content is output by default.  |
| edges         | No        | Array of Json | List of edges to be queried. Either this parameter or <b>vertices</b> is selected. For details, see <a href="#">Table 5-301</a> .   |
| filters       | Yes       | Array of Json | Filter criteria. Each element in the array corresponds to a filter. For details about the formats, see <a href="#">Table 5-302</a> .  |
| full_path     | No        | Boolean       | Whether to return a complete path. The default value is <b>false</b> . <ul style="list-style-type: none"> <li>• If the value is <b>true</b>, the paths from the source vertex to all leaf vertices are returned.</li> <li>• If the value is <b>false</b>, the paths from the source vertex to the leaf vertices at layer <math>k</math> are returned.</li> </ul>  |



| Parameter          | Mandatory | Type    | Description   |
|--------------------|-----------|---------|---|
| visualized         | No        | Boolean | Whether to enable visualization. The default value is <b>false</b> . In asynchronous mode: <ul style="list-style-type: none"> <li>When <b>visualized</b> is <b>false</b>, the job query result is returned on multiple pages.</li> <li>When <b>visualized</b> is <b>true</b>, the job query result is returned on one page.</li> </ul>      |
| restricted(2.2.28) | No        | Boolean | Whether to verify the input. The default value is <b>true</b> . <ul style="list-style-type: none"> <li><b>true</b>: If <b>vertices</b> contains vertices that do not exist, the query exits and an error is reported.</li> <li><b>false</b>: The system filters out vertices that do not exist and then performs the query task.</li> </ul> |

**Table 5-301** edges element formats

| Parameter | Mandatory | Type   | Description                             |
|-----------|-----------|--------|---|
| source    | Yes       | String | Source vertex ID                        |
| target    | Yes       | String | Target vertex ID                        |
| index     | No        | String | Indexes of edges in the source edge set |

**Table 5-302 Filters element formats**

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| operator      | Yes       | String | <p>Query type. Possible values:</p> <ul style="list-style-type: none"> <li>• <b>inV</b>: incoming vertex</li> <li>• <b>outV</b>: outgoing vertex</li> <li>• <b>bothV</b>: incoming and outgoing vertices</li> <li>• <b>vertex</b>: all vertices. Filtering is available only at the first layer. If vertices are input in the beginning, the first-layer output is the input vertices. If no vertices are input in the beginning, all vertices are output at the first layer.</li> <li>• <b>in</b>: incoming edge</li> <li>• <b>out</b>: outgoing edge</li> <li>• <b>both</b>: incoming and outgoing edges</li> <li>• <b>edge</b>: all edges. Filtering is available only at the first layer. The usage is similar to that of vertices</li> </ul> <p>The query result of the previous layer is the input of the next layer.</p> <ul style="list-style-type: none"> <li>• If the result of the previous layer is a vertex, the corresponding operations can be <b>inV</b>, <b>outV</b>, <b>bothV</b>, <b>in</b>, <b>out</b>, and <b>both</b>.</li> <li>• If the result of the previous layer is an edge, the corresponding operation can be <b>inV</b>, <b>outV</b>, and <b>bothV</b>.</li> </ul> |
| vertex_filter | No        | String | This parameter is optional when <b>operator</b> is set to <b>inV</b> , <b>outV</b> , or <b>bothV</b> . For details about the formats, see <a href="#">Table 5-304</a> .  |
| edge_filter   | No        | String | This parameter is optional when <b>operator</b> is set to <b>in</b> , <b>out</b> , or <b>both</b> . For details about the formats, see <a href="#">Table 5-304</a> .   |

**Table 5-303 by element formats**

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| id        | No        | Boolean | Whether to output the ID. The default value is <b>false</b> .    |
| label     | No        | Boolean | Whether to output the label. The default value is <b>false</b> . |

| Parameter          | Mandatory | Type            | Description  |
|--------------------|-----------|-----------------|--|
| properties         | No        | Boolean         | Whether to output properties. The default value is <b>false</b> .  |
| selectedProperties | No        | Array of String | When <b>properties</b> is set to <b>true</b> , you can select the properties to be output. If this parameter is left blank, all properties are output. By default, this parameter is left blank. |

**Table 5-304** property\_filter element formats

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| leftvalue | Yes       | String | Left value. For details about the formats, see <a href="#">Table 5-305</a> . |

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| predicate  | Yes       | String | <p>Filtering type. The supported operations include:</p> <p>Relational operators:</p> <ul style="list-style-type: none"> <li>• =: equal to</li> <li>• !=: not equal to</li> <li>• &lt;: less than</li> <li>• ≤: Less than or equal to</li> <li>• &gt;: greater than</li> <li>• ≥: greater than or equal to</li> </ul> <p>Logical operations:</p> <ul style="list-style-type: none"> <li>• &amp;: and</li> <li>•  : or</li> </ul> <p>Set operations:</p> <ul style="list-style-type: none"> <li>• IN/NOTIN: whether the left value and right value have an intersection</li> <li>• CONTAIN/NOTCONTAIN: whether the property value contains the right value</li> <li>• SUBSET: The right value is a subset of the property value.</li> </ul> <p>Match operators:</p> <ul style="list-style-type: none"> <li>• PREFIX: The right value is the prefix of the left value.</li> <li>• NOTPREFIX: The right value is not the prefix of the left value.</li> <li>• SUFFIX: The right value is the suffix of the left value.</li> <li>• NOTSUFFIX: The right value is not the suffix of the left value.</li> <li>• SUBSTRING: The right value is a sub-string of the left value.</li> <li>• NOTSUBSTRING: The right value is not a sub-string of the left value.</li> <li>• FUZZY: fuzzy match</li> <li>• REGEX: regex match</li> <li>• CISUBSTRING: sub-string that ignores cases</li> </ul> <p>HAS/HASNOT: whether this property exists. Only property filtering is supported. That is, the left value can only be <b>property_name</b>.</p> |
| rightvalue | Yes       | String | <p>Right value. For details about the formats, see <a href="#">Table 5-306</a>.</p>   |

**Table 5-305 leftvalue** element formats

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| label_name      | No        | String | If <b>label</b> is used as the filter criterion, <b>label_name</b> can be selected and the value is <b>labelName</b> . Set the <b>value</b> field of <b>rightvalue</b> to the label name.           |
| property_name   | No        | String | If <b>property</b> is used as the filter criterion, <b>property_name</b> can be selected and the value is the property name. Set the <b>value</b> field of <b>rightvalue</b> to the property value. |
| id              | No        | String | If the vertex ID is filtered, this parameter is optional.   |
| property_filter | No        | String | If <b>predicate</b> is set to <b>&amp;</b> or <b> </b> , <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b> .   |

**Table 5-306 rightvalue** element formats

| Parameter       | Mandatory | Type   | Description  |
|-----------------|-----------|--------|--|
| value           | Yes       | String | <ul style="list-style-type: none"> <li>If <b>label</b> is used as the filter criterion, the value is the label name.</li> <li>If <b>property</b> is used as the filter criterion, the value is the property name.</li> </ul> |
| property_filter | No        | String | If <b>predicate</b> is set to <b>&amp;</b> or <b> </b> , <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b> .  |

**Table 5-307 predicate** application scenarios

| predicate          | label_name | id | property_name | Nested Filtering |
|--------------------|------------|----|---------------|------------------|
| &                  | No         | No | No            | Yes              |
|                    | No         | No | No            | Yes              |
| HAS/HASNOT         | No         | No | Yes           | No               |
| CONTAIN/NOTCONTAIN | No         | No | Yes           | No               |

| predicate          | label_name | id  | property_name | Nested Filtering   |
|--------------------|------------|-----|---------------|--|
| SUBSET             | No         | No  | Yes           | Yes (Only the right value set is supported. If the right value is single, no filtering function is available.) |
| IN/NOTIN           | Yes        | Yes | Yes           | Yes (Only the right value set is supported. If the right value is single, no match is available.)              |
| PREFIX             | Yes        | Yes | Yes           | No   |
| FUZZY              | Yes        | Yes | Yes           | No   |
| REGEX              | Yes        | Yes | Yes           | No   |
| SUBSTRING          | Yes        | Yes | Yes           | No   |
| CISUBSTRING        | Yes        | Yes | Yes           | No   |
| =/!<br>=/<</<=>/>= | Yes        | Yes | Yes           | No   |

 **NOTE**

- The left value set is supported. The left value in the body is a string.
- The right value set is supported. If you select **No**, only the first character string in the set is matched even if the right value set is supported.
- Boolean value matching. When the right value is **true**, the value is identified as true for matching. Otherwise, the value is identified as false for matching.

## Response Parameters

- Synchronous response

**Table 5-308** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. This parameter is left blank when the query fails.   |

**Table 5-309** data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| vertices  | List | Vertex result set. If the last layer of <b>filters</b> is vertex filtering, the <b>data</b> contains vertices.  |
| edges     | List | Edge result set. If the last layer of <b>filters</b> is edge filtering, the <b>data</b> contains edges.   |
| paths     | List | Path set. This parameter is available only when <b>with_path</b> is set to <b>true</b> . For details about the formats, see <a href="#">Table 5-310</a> . |

**Table 5-310** path parameter description

| Parameter | Type   | Description      |
|-----------|--------|------------------|
| source    | String | Source vertex ID |
| target    | String | Target vertex ID |
| index     | String | Edge index       |
| label     | String | Edge label       |

- Asynchronous response

**Table 5-311** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String  | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank when the request fails.   |
| jobType      | Integer | Job type. This parameter is left blank when the request fails.  |

### Example Request

- (Synchronous mode) List the kth-hop vertices or edges that meet filter criteria. The execution mode is synchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=filtered-query

```
{
  "executionMode": "sync",
  "visualized": "false",
  "filters": [
    {
      "operator": "outV"
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "rate"
          }
        }
      }
    }
  ],
  "full_path": false,
  "vertices": [
    "tr_10"
  ]
}
```

- (Asynchronous mode) List the kth-hop vertices or edges that meet filter criteria. The execution mode is asynchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.



```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query
{
  "executionMode": "async",
  "visualized": "false",
  "filters": [
    {
      "operator": "outV"
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "rate"
          }
        }
      }
    }
  ],
  "full_path": false,
  "vertices": [
    "tr_10"
  ]
}
```

- property\_filter is nested. List the kth-hop vertices or edges that meet filter criteria. The execution mode is synchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.

```
{
  "executionMode": "sync",
  "filters": [
    {
      "operator": "outV",
      "vertex_filter": {
        "property_filter": {
          "leftvalue": {
            "property_filter": {
              "leftvalue": {
                "property_name": "genres"
              },
              "predicate": "PREFIX",
              "rightvalue": {
                "value": "A|"
              }
            }
          }
        }
      },
      "predicate": "&",
      "rightvalue": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "movie"
          }
        }
      }
    }
  ],
  "vertices": [
    "tr_3"
  ]
}
```

```
]
}
```

## Example Response

- Synchronous response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "edges": [
      {
        "index": "1",
        "source": "tr_1",
        "label": "rate",
        "properties": {
          "Rating": [
            0
          ],
          "Datetime": [
            ""
          ]
        },
        "target": "tr_3"
      },
      .....
      {
        "index": "199998",
        "source": "tr_1",
        "label": "rate",
        "properties": {
          "Rating": [
            0
          ],
          "Datetime": [
            ""
          ]
        },
        "target": "tr_200000"
      }
    ]
  }
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8806"
}
```

- Asynchronous response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "6622f13c-4b88-45f5-89a9-eea096647c4a",
  "jobType": 1
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "executionMode is not correct, it should be sync or async",
  "errorCode": "GES.8806"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.1.16 Filtered-query V2

#### Function

This is a new version of the Filtered Query API. This API supports both Filtered Query and Repeat Query functions.

#### NOTE

You can use this API to accelerate multi-hop filtered query and looped traversal query. For example, a Gremlin query like this:  
`g.V('node1').repeat(out('label_2')).emit()`

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=path-query

**Table 5-312** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

 NOTE

The number of elements in each traversal cannot exceed 100 million.

**Table 5-313** Request body parameters

| Parameter                  | Man<br>dato<br>ry | Type             | Defaul<br>t<br>Value | Description  |
|----------------------------|-------------------|------------------|----------------------|--|
| executionM<br>ode (2.2.22) | No                | String           | sync                 | Execution mode of the query task. The value can be <b>sync</b> (synchronously) or <b>async</b> (asynchronously).   |
| vertices                   | Yes               | Array of<br>Json | None                 | List of IDs of source vertices you want to query   |
| repeat                     | Yes               | Array of<br>Json | None                 | Filter criteria for repeat queries. Each element in the array corresponds to a filter. For details about the format, see <a href="#">repeat elements</a> .   |
| until<br>(2.2.22)          | No                | Array of<br>Json | None                 | Conditions to stop the traversal. The logic is similar to while/do loop. For details about the format, see <a href="#">until elements</a> . For APIs of the 2.2.22 version, <b>until</b> supports only one stop condition.   |
| times                      | No                | int              | 2                    | Maximum number of steps. The default value is <b>2</b> , and the maximum value is <b>20</b> .  |
| emit                       | No                | Boolean          | false                | Whether all elements will be returned. The default value is <b>false</b> . If <b>select</b> and <b>as</b> are configured, or <b>queryType</b> is set to <b>Tree</b> , this parameter determines whether vertices that are not on the final complete path will be returned. |
| limit                      | No                | int              | 10000                | Number of vertices, edges, or paths.   |
| queryType(2<br>.2.22)      | No                | String           | Defaul<br>t          | Type of the query. The value can be <b>Default</b> or <b>Tree</b> . <b>Default</b> means that the path query result will be returned. <b>Tree</b> means that the path information of the path query will be returned in the tree structure.                                |

| Parameter       | Mandatory | Type            | Default Value | Description   |
|-----------------|-----------|-----------------|---------------|---|
| select (2.2.21) | No        | Array of String | None          | <p>Fields you want to be displayed in the result. The values can be the fields set in the <b>as</b> parameter, or you can set the <b>by</b> parameter together with this parameter to control output content.</p> <p>If you set the <b>by</b> parameters together, you can set <b>select</b> to <b>v0</b>, <b>v1</b>, <b>v2</b>, ..., and <b>vtimes</b>:</p> <ul style="list-style-type: none"> <li>• <b>v0</b>: layer 0 of the k hops</li> <li>• <b>v1</b>: the first layer of the K hops</li> <li>• <b>v2</b>: the second layer of the K hops</li> </ul> <p>The path selected by this parameter is deduplicated by default.</p> |
| by(2.2.21)      | No        | Array of Json   | Output all    | <p>Content of output fields.</p> <ol style="list-style-type: none"> <li>1. If this parameter is not specified, all content is output by default.</li> <li>2. If the <b>select</b> parameter is configured, the number of <b>by</b> must be the same as the number of <b>select</b>. Each <b>by</b> can output only one element.</li> <li>3. If <b>select</b> is not configured, <b>by</b> takes effect on the final result set. For details, see <b>by elements</b>.</li> </ol>   |
| statistics      | No        | Boolean         | false         | Whether only the number of hit records will be returned. The default value is <b>false</b> .  |
| mode            | No        | Boolean         | None          | Traversal mode that will be set forcibly. The value can be <b>Dense</b> or <b>Sparse</b> . The default mode is automatically switched based on the graph structure.   |
| strategy        | No        | String          | ShortestPath  | Traversal policy. The value can be <b>ShortestPath</b> or <b>Walk</b> .   |

| Parameter  | Mandatory | Type    | Default Value | Description   |
|------------|-----------|---------|---------------|---|
| restricted | No        | Boolean | true          | <p>Whether the input is restricted. The default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: If <b>vertices</b> contains vertices that do not exist, the query exits and an error is reported.</li> <li>• <b>false</b>: The system filters out vertices that do not exist and then performs the query task.</li> </ul> |

**Table 5-314** by elements

| Parameter          | Mandatory | Type            | Description  |
|--------------------|-----------|-----------------|--|
| id                 | No        | Boolean         | Whether to output the ID. The default value is <b>false</b> .  |
| label              | No        | Boolean         | Whether to output the label. The default value is <b>false</b> .   |
| properties         | No        | Boolean         | Whether to output properties. The default value is <b>false</b> .  |
| selectedProperties | No        | Array of String | When <b>properties</b> is set to <b>true</b> , you can select the properties to be output. If this parameter is left blank, all properties are output. |

**Table 5-315** repeat elements

| Parameter     | Mandatory | Type        | Description   |
|---------------|-----------|-------------|---|
| operator      | Yes       | String      | Type of the query. The value can be <b>inV</b> (incoming vertex), <b>outV</b> (outgoing vertex), and <b>bothV</b> (incoming and outgoing vertices). |
| vertex_filter | No        | JSON String | Search conditions for the next hop. For details about the format, see <a href="#">property_filter elements</a> .                                    |
| edge_filter   | No        | JSON String | Edge search conditions. For details about the format, see <a href="#">property_filter elements</a> .  |
| as            | No        | JSON String | Alias of elements at this layer. The value can be used to select output fields.   |

**Table 5-316** until elements

| Parameter     | Mandatory | Type        | Description  |
|---------------|-----------|-------------|--|
| vertex_filter | No        | JSON String | This parameter is optional when <b>operator</b> in <b>repeat</b> is set to <b>inV</b> , <b>outV</b> , or <b>bothV</b> . For details about the format, see <a href="#">property_filter elements</a> . |

**Table 5-317** property\_filter elements

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| leftvalue | Yes       | String | Left value of a search condition. For details, see <a href="#">leftvalue elements</a> . |

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| predicate  | Yes       | String | <p>Filtering type. The options are:</p> <p>Relational operators:</p> <ul style="list-style-type: none"> <li>• =: equal to</li> <li>• !=: not equal to</li> <li>• &lt;: less than</li> <li>• ≤: Less than or equal to</li> <li>• &gt;: greater than</li> <li>• ≥: greater than or equal to</li> </ul> <p>Logical operations:</p> <ul style="list-style-type: none"> <li>• &amp;: and</li> <li>•  : or</li> </ul> <p>Set operations:</p> <ul style="list-style-type: none"> <li>• IN/NOTIN: whether the left value and right value have an intersection</li> <li>• CONTAIN/NOTCONTAIN: whether the property value contains the right value</li> <li>• SUBSET: The right value is a subset of the property value.</li> </ul> <p>Match operators:</p> <ul style="list-style-type: none"> <li>• PREFIX: The right value is the prefix of the left value.</li> <li>• NOTPREFIX: The right value is not the prefix of the left value.</li> <li>• SUFFIX: The right value is the suffix of the left value.</li> <li>• NOTSUFFIX: The right value is not the suffix of the left value.</li> <li>• SUBSTRING: The right value is a sub-string of the left value.</li> <li>• NOTSUBSTRING: The right value is not a sub-string of the left value.</li> <li>• FUZZY: fuzzy match</li> <li>• REGEX: regex match</li> <li>• CISUBSTRING: sub-string that ignores cases</li> </ul> <p>HAS/HASNOT: whether this property exists. Only property filtering is supported. That is, the left value can only be <b>property_name</b>.</p> |
| rightvalue | Yes       | String | <p>Right value. For details about the format, see <a href="#">rightvalue elements</a>.</p>   |



**Table 5-318 leftvalue** elements

| Parameter       | Mandatory | Type        | Description   |
|-----------------|-----------|-------------|---|
| label_name      | No        | String      | If <b>label</b> is used as the filter criterion, <b>label_name</b> can be selected and the value is <b>labelName</b> . Set the <b>value</b> field of <b>rightvalue</b> to the label name. |
| property_name   | No        | String      | If <b>property</b> is used as the filter criterion, set this parameter to the property name and set <b>value</b> of <b>rightvalue</b> to the property value.                              |
| id              | No        | String      | If the vertex ID is filtered, this parameter is optional.   |
| property_filter | No        | JSON String | If <b>predicate</b> is set to <b>&amp;</b> or <b> </b> , <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b> .   |
| degree          | No        | String      | Direction of vertex degree filtering statistics. This parameter is optional. The value can be <b>both</b> , <b>in</b> , or <b>out</b> .   |

**Table 5-319 rightvalue** elements

| Parameter       | Mandatory | Type        | Description   |
|-----------------|-----------|-------------|---|
| value           | Yes       | String      | If <b>label</b> is used as the search criterion, the value is the label name. If <b>property</b> is used as the search criterion, the value is the property name. |
| property_filter | No        | JSON String | If <b>predicate</b> is set to <b>&amp;</b> or <b> </b> , <b>property_filter</b> can be nested in <b>leftvalue</b> and <b>rightvalue</b> .                         |

**Table 5-320 predicate** supported left values

| predicate          | label_name | id  | property_name | Nested Filters |
|--------------------|------------|-----|---------------|----------------|
| &                  | No         | No  | No            | Yes            |
|                    | No         | No  | No            | Yes            |
| HAS/HASNOT         | No         | No  | Yes           | No             |
| =/!<br>=/<</<=>/>= | Yes        | Yes | Yes           | No             |

## Response Parameters

- Synchronous response

**Table 5-321** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | Query results. If the query fails, the field is empty.  |

**Table 5-322** data parameter description

| Parameter | Type | Description  |
|-----------|------|--|
| vertices  | List | Vertex result set. If the last layer of <b>filters</b> is vertex filtering, the <b>data</b> contains vertices. |
| edges     | List | Edge result set. If the last layer of <b>filters</b> is edge filtering, the <b>data</b> contains edges.        |

- Asynchronous response

**Table 5-323** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type    | Description   |
|-----------|---------|---|
| errorCode | String  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| job_id    | String  | ID of the algorithm execution job. This parameter is left blank when the request fails.   |
| jobType   | Integer | Task type. This parameter is left blank when the request fails.   |

## Example Request

- Example request 1: List the kth-hop vertices or edges that meet filter criteria. The query type is outgoing vertex, and the query is performed on the next-hop vertex.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query
{
  "repeat":[
    {
      "operator":"outV",
      "vertex_filter":{
        "property_filter":{
          "leftvalue":{
            "label_name":"labelName"
          },
          "predicate":"=",
          "rightvalue":{
            "value":"rate"
          }
        }
      }
    }
  ],
  "times":2,
  "vertices":[
    "1","2"
  ]
}
```

### NOTE

The preceding request is equivalent to this Gremlin statement:  
**g.V('1','2').repeat(out().hasLabel('rate').times(2).dedup().**

- Example request 2: List the kth-hop vertices or edges that meet filter criteria. The query type is outgoing vertex, and the query is performed on the next-hop vertex.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query
{
  "repeat":[
    {
      "operator":"outV",
      "vertex_filter":{
        "property_filter":{
          "leftvalue":{
            "label_name":"labelName"
          }
        }
      }
    }
  ]
}
```

```

    },
    "predicate":"=",
    "rightvalue":{
      "value":"rate"
    }
  }
}
],
"until":[
  {
    "vertex_filter":{
      "property_filter":{
        "leftvalue":{
          "property_name":"movieid"
        },
        "predicate":"=",
        "rightvalue":{
          "value":"1"
        }
      }
    }
  }
]
"vertices":[
  "v1","v2"
]
}

```

 **NOTE**

The preceding request is equivalent to this Gremlin statement:  
`g.V('v1','v2').repeat(out().hasLabel('rate')).until(has('movieid','1')).dedup()`

## Example Response

- Synchronous response

**Status code: 200**

Example response (successful request)

```

{
  "data":{
    "vertices":[
      {
        "id":"51",
        "label":"user",
        "properties":{
          "occupation":[
            "homemaker"
          ],
          "gender":[
            "F"
          ],
          "Zip-code":[
            "46911"
          ],
          "userid":[
            5
          ],
          "age":[
            "56+"
          ]
        }
      }
    ]
  }
}

```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8806"
}
```

- Asynchronous response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "6622f13c-4b88-45f5-89a9-eea096647c4a",
  "jobType": 1
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "executionMode is not correct, it should be sync or async",
  "errorCode": "GES.8806"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.1.17 Domain-Specific Language (DSL) Query APIs**

**5.1.17.1 Executing the DSL Algorithm**

**Function**

This API is used to provide flexible and controllable DSLs to help users design and run algorithms at low costs. For details about the DSL algorithm, see *DSL Syntax*.

After the DSL algorithm is executed, you need to use the "Dumping HyG Algorithm Results" API to dump the DSL execution results to OBS.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=algorithm-query

**Table 5-324** URI parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| project_id | Yes       | String | Project ID, which is used for resource isolation. For details about how to obtain the value, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name  |

## Request Parameters

**Table 5-325** Body parameter

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| commands  | Yes       | String | Command executed by the custom operation. For details about the syntax, see the syntax introduction. |

## Response Parameters

**Table 5-326** Response parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| data      | Object | Query results. This parameter is left blank when the query fails. |

## Example Request

```
POST/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=algorithm-query
{
  "commands":"Match<Vertex> v(['1']); v.repeat(bothV()).limit(2); return v;"
}
```

## Example Response

- Example response for a successful request

**Status code: 200**

OK

```
{
  "data":{
    "vertices":[
      {
        "id":"1",
        "label":"movie",
        "properties":{
          "genres":[
            "Comedy"
          ],
          "movieid":[
            1
          ],
          "title":[
            "Airplane! (1980)"
          ]
        }
      }
    ],
    "runtime":0.126476598
  }
}
```

- Example response for a failed request

```
{
  "errorCode":"GES.8814",
  "errorMessage":"Unsupported API."
}
```

## Return Value

- Normal  
200
- Abnormal

**Table 5-327** Description

| Return Value    | Description    |
|-----------------|----------------|
| 400 Bad Request | Request error. |

| Return Value              | Description               |
|---------------------------|---------------------------|
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

### 5.1.17.2 DSL Syntax

#### Syntax Introduction

The Algorithm query API is a graph DSL provided by GES. You can use it to query and calculate graphs. During service planning, various query operators and fine-grained basic computing pattern operators are added to DSL so that DSL can support user-defined graph traversal, multi-hop filtering query, pattern matching, similarity algorithms, community algorithms, recommendation algorithms, path analysis, and custom service algorithms.

For example, to query the neighborhood vertex set of the second hop from vertices 1 and 2, run the following command:

```
Match<Vertex> v(['1','2']); v.repeat(bothV()).times(2).limit(3); return v;
```

#### DSL Statement Construction

Statements are DSL program fragments that are executed in sequence. To enrich DSL's capability of expressing user-defined algorithms, DSL supports the following statements: selection statement, loop statement, expression statement, operator operand, declaration statement, and jump statement. Typically, the DSL structure consists of a declaration statement, an expression statement, and a jump statement. For example:

```
GlobalAcc<Sum,int> g=0; // Declaration statement, which declares an aggregator variable g
g+=1*3+2; // Expression statement, which is used to perform the aggregation operation on the
aggregator g. g = g + 1 x 3 + 2
return g; // Jump statement, which ends DSL and returns g
```

#### NOTE

Currently, only the following selection and loop statements are supported, which are written in selection, loop, and lambda expressions:

1. Assignment/aggregation of various expressions
2. Update operator of the Vertex matcher
3. Gather operator of the Vertex matcher



**Table 5-328** DSL syntax description

| Syntax                       | Keyword                                       | Description   |
|------------------------------|---|---|
| Select statement             | if  | Executes a statement conditionally.   |
| Loop statement               | while   | Executes statements repeatedly.   |
| Jump statement               | return  | Returns different types of results.   |
| Declaration statement        | Match< ? >                                    | Declares a matcher Match to quickly match graph data for more operations.   |
|                              | VertexAcc< ?, ?>                              | Declares a vertex aggregator to define additional properties/variables on a vertex. Match can be used to perform operations on additional variables on vertices in batches. |
|                              | GlobalAcc< ?, ?>                              | Declares a global aggregator to define globally operable variables.   |
| Expression statement         | operator=                                     | Assigns values to variables.  |
|                              | operator+=                                    | Variable aggregation operation  |
|                              | Arithmetic operations<br>+, -, *, /           | Arithmetic operations   |
|                              | Comparison operations<br>>, >=, ==, <, <=, != | Comparative operations  |
|                              | lambda expression                             | Anonymous expressions   |
| Operator operation statement | repeat  | Allows users to perform multi-hop filtering query. The syntax includes emit, times, and limit.  |
|                              | update  | Performs a set of batch operations defined on the matcher   |
|                              | gather  | Performs a set of batch operations defined on the matcher   |
|                              | pick  | Quickly obtains <i>N</i> random vertices  |
|                              | init  | Reinitializes the vertex set of the vertex matcher  |
|                              | insert  | Adds a matching vertex set to the vertex matcher  |

| Syntax | Keyword      | Description  |
|--------|--------------|--|
|        | move         | Quickly moves the vertex set in <b>other_match_vertex</b> to the <b>match_vertex</b> matcher                           |
|        | subgraph     | It is mainly used in the return statement. It can return guidance subgraphs of the <b>match_vertex</b> vertex matcher. |
|        | intersection | Takes the intersection of two vertex matchers  |
|        | pattern      | Executes a complete Cypher statement and places the result in the vertex matcher                                       |

1. **Select statement: if**

```
if(expression) {
    true Branching statement
}
```

Executes a statement conditionally.

```
GlobalAcc<Sum, int> threshold = 10;
Match<Vertex> v(['1', '2']);
if(threshold < 20) {
    v.repeat(outV().has('name', 'peter')).times(2).emit();
}
```

2. **Loop statement: while**

```
while(expression) {
    true Branching statement
}
```

Executes some code repeatedly and conditionally

```
GlobalAcc<Sum, int> loop = 0;
VertexAcc<Sum, int> score = 1;
Match<Vertex> v(['1', '2']);
while(loop < 10) {
    loop +=1;
    v.update((v)->{v.score = 1 + 2 * v.score;});
}
```

3. **Jump statement: return**

Returns JSON files of different types and formats

| Type      | Return Type             | Description  |
|-----------|-------------------------|--|
| Match     | Vertex set              | -  |
| VertexAcc | map                     | By default, values that are not changed after initialization are not output. |
| GlobalAcc | Single value            | -  |
| subgraph  | Vertex set and edge set | -  |

If some vertices are matched, they can be directly returned:

```
Match<Vertex> v(['1','2']);
return v;
{
  "vertices": [
    {
      "id": "1",
      "label": "movie",
      "properties": {
        "genres": [
          "Comedy"
        ]
      }
    }
  ]
}
```

Alternatively, ACC values can be directly returned.

```
GlobalAcc<Max,int> g1=10;
g1+=2;
return g1;
{
  "data": {
    "value": 10
  }
}
```

Assume that you want to obtain the guidance subgraph of a match vertex set for subsequent task execution (for example, drawing on the canvas).

```
Match<Vertex> v(['1','2']);
return v.subgraph();
{
  "data": {
    "vertices": [
      {
        "id": "1",
        "label": "user",
        "properties": {
          //balabala
        }
      },
      {
        "id": "2",
        "label": "movie",
        "properties": {
          //balabala
        }
      }
    ],
    "edges": [
      {
        "index": "0",
        "source": "1",
        "label": "rate",
        "properties": {
          //balabala
        },
        "target": "2"
      }
    ]
  }
}
```

#### 4. Expression statement

An expression is a sequence of operators and operands, which specifies a calculation. Different statements may use expressions. For example, loop statements and selection statements may use expressions to express branches or loop conditions. Expressions can also be used to assign values to variables and perform aggregation operations.

- The following table lists common operators:

| Common Operator | Form                       | Symbol               |
|-----------------|----------------------------|----------------------|
| Assignment      | a=b;                       | =                    |
| Aggregation     | a+=b                       | +=                   |
| Mathematical    | a+b;a-b;a*b;a/b            | +, -, *, /           |
| Comparison      | a>b;<br>a=b;a<=b;a==b;a!=b | >, >=, ==, <, <=, != |

 NOTE

The calculation of aggregation operations is determined by the definition of [Aggregator](#).

- The following table lists the priorities among expressions:

| Priority | Operator  | Combination   |
|----------|---|---------------|
| 1        | () Function call []<br>Subscript .Member access a++<br>Suffix auto-increment a--<br>Suffix auto-decrement | Left to right |
| 2        | a*b a/b Multiplication and division   | Left to right |
| 3        | a+b a-b Addition and subtraction  | Left to right |
| 4        | < <= > >= Relational operators  | Left to right |
| 5        | == != Equality operators  | Left to right |
| 6        | and Logical AND   | Left to right |
| 7        | or Logical OR   | Left to right |
| 8        | = += Assignment, aggregation operations   | Right to left |

- **Expression types**

Due to the particularity of variables (different types of aggregators), DSL classifies expressions into the following types:

- i. numeric expression - Numerical expression
- ii. vertexacc expression - Expression containing the vertexacc variable
- iii. globalacc expression - Expression containing the globalacc variable

```
1+2*3 // Numerical expression

GlobalAcc<Sum, int> diff = 0;
VertexAcc<Sum, int> score = 0;
Match<Vertex> v(['Tom']);

1+diff*2 // Expression with globalAcc, which belongs to globalacc expression
1+v.score*2 // Expression with vertexAcc, which belongs to vertexacc expression
```

- **Expression upgrade table**

There are different constraints on assignment and calculation for different types of expressions. That is, whether the assignment and calculation between different types are valid is restricted.

The following expressions are used to indicate whether assignment/aggregation operations support different expressions.

| Left Value/<br>Right Value | numeric<br>expression   | vertexacc<br>expression                                 | globalacc<br>expression |
|----------------------------|-------------------------|---|-------------------------|
| vertexacc<br>variable      | vertexacc<br>expression | Partially<br>supported, the<br>same matcher<br>required | vertexacc<br>expression |
| globalacc<br>variable      | globalacc<br>expression | Not supported   | globalacc<br>expression |

- **Value promotion strategy**

During the calculation of an expression, a right value can be converted into a value of another type. The following lists only the supported value promotion strategies:

- i. Integer value promotion: Boolean values can be converted to integer values. The value **false** changes to **0**, and the value **true** changes to **1**.
- ii. Floating-point value promotion: Right values of the floating-point type can be converted to values of the double type without changing the values.

- **Value conversion**

Value conversion is currently not supported. Unlike promotion, value conversions can change values and have potential precision losses.

- **Lambda expressions**

```
($parameters)->{statements;}
```

It is mainly used to receive some steps of a function. For example, both the gather and update steps support transferring lambda expressions as parameters. Variables such as **GlobalAcc** and **VertexAcc** are used in lambda expressions to assign and aggregate values. Note the following:

- i. The parameter type does not need to be declared and can be identified in a unified manner.

- ii. GlobalAcc in the context can be accessed by lambda expressions.
- iii. Lambda expressions can be executed as independent function bodies. The input parameters must be empty.

```
GlobalAcc<Sum,int> g1=0;
g1+=2*10+1;
()->{g1+=2*10+1;} // Same result as the previous statement

// update receives lambda as a parameter:
v.update((v1)->{g1+=v1.acc1*2+g2+v1.acc2;});
```

## Variable Declaration

### Match matcher

1. Match matcher description

DSL allows you to define objects such as vertices, paths, and submaps that can be matched and on which related operations can be performed. After declaring and initializing the matcher, you can use the matcher operator to perform operations on vertex sets in batches. For details, see [Match matcher operators](#). Run the following statement to declare and initialize the matcher:

```
Match<[Vertex|Path|Subgraph]> $variable;
```

DSL allows you to define vertices, paths, and submaps that can be matched and on which related operations can be performed.

```
Match<Vertex> Quickly match and perform operations on vertices
```

2. Match<Vertex> matcher: Match matcher that can be used to quickly match and perform operations on vertex sets

```
Match<[ Vertex ]> $match_vertex_variable;
Match<[ Vertex ]> $match_vertex_variable($VertexList);
```

You can use Match<Vertex> to match or initialize vertex sets.

```
Match<Vertex> v(['1', '2']); // Directly match by vertex ID.
Match<Vertex> v(); // Only Match variables are defined. The vertices to be matched are empty.
Match<Vertex> v;
v.pattern('match (n:user) where n.age>30 return n limit 10'); // Uses the results of the Cypher
statements to match the filtered vertices.
```

DSL provides a large number of operators for Match<Vertex>, such as **pattern**, **init**, and **pick**. For details, see [Match matcher operators](#).

3. Relationship between match matcher and aggregators

You can define an aggregator to hold values and calculations. Each type of matcher can perform operations on its corresponding aggregator in batches.

For example, you can use Match<Vertex> to perform aggregation calculation on a specified vertex set.

### Aggregator

1. Aggregator description

Aggregators are used to simplify the expression of values during calculation. Different types of aggregators define different scopes of batch aggregation operations.

 **NOTE**

DSL does not support direct definition of variables of various numeric types to carry data generated during calculation. Instead, DSL provides aggregators to simplify operations in various application scenarios.

1. VertexAcc can be used to define additional "properties (variables)" on vertices, and Match<Vertex> can be used to perform operations on additional variables on vertices in batches.
2. GlobalAcc can be used to define globally operable variables.

Aggregator declaration syntax:

```
Accumulator<Aggregator Operator, NumericType> v;
```

Parameter description:

NumericType: int, (float and double types are not supported currently)

Aggregator Operator: Sum,Max,Min

DSL allows you to define different aggregators to simplify algorithm operations. Currently, two types of aggregate variables are supported:

- a. VertexAcc<Aggregator Operator, type>
- b. GlobalAcc<Aggregator Operator, type>

You can perform the following operations on an aggregator:

- a. Initialization: Define an aggregator and assign initial values.
- b. Assignment: Reset the aggregator values.
- c. Aggregation: Perform aggregation operations based on defined aggregator operators.

## 2. Aggregator operator description

Different aggregation operations are provided, such as Sum, Max, and Min. Variables are updated using operators + and =.

```
$match_vertex.$vertex_accumulator += $value;  
$global_accumulator += $value;
```

## 3. Vertex aggregator VertexAcc<Aggregator Operator, type>

VertexAcc can be used to quickly define additional "property (variables)" on vertices. and Match<Vertex> can be used to perform operations on additional variables on vertices in batches. This significantly facilitates the calculation process.

```
VertexAcc<Sum, float> score = 0.5;
```

Each time a VertexAcc is defined, DSL allocates a variable of the type type to each vertex in the full graph. You can use Match<Vertex> to perform operations on the defined VertexAcc. For example:

```
VertexAcc<Sum, int> score = 0;  
// Tom.score = 0, Jack.score = 0  
Match<Vertex> v(['Tom', 'Jack']);  
// Tom.score = 0, Jack.score = 0  
v.score += 1; // This operation aggregates the scores of Tom and Jack. That is, their scores are  
increased by 1.  
// Tom.score = 1, Jack.score = 1  
v.score = 10; //This operation assigns values to the scores of Tom and Jack. That is, their scores are  
updated to 10.  
// Tom.score = 10, Jack.score = 10  
v.score += 5; // This operation aggregates the scores of Tom and Jack. That is, their scores are  
increased by 5.  
// Tom.score = 15, Jack.score = 15
```

VertexAcc can be operated by expressions. Currently, numeric, GlobalAcc, and VertexAcc expressions can be used for aggregation and assignment.

```
VertexAcc<Sum, int> score = 0;
VertexAcc<Sum, int> factor = 1;
GlobalAcc<Sum, int> alpha = 10;
Match<Vertex> v(['Tom']);
// Tom.score = 0.
v.score = alpha x 2 + 3; // This operation assigns a value to Tom's score.
// Tom.score = alpha x 2 + 3 = 10 x 2 + 3 = 23

v.score += v.factor*2; // This operation assigns a value to Jack's score.
// Tom.score = Tom.score + Tom.factor x 2 = 23 + 1 x 2 = 25
```

4. Global aggregator GlobalAcc<Aggregator Operator, type>

Each time a GlobalAcc is defined, DSL creates a variable of the type type in the DSL scope. Then, perform operations on GlobalAcc directly. For example:

```
GlobalAcc<Sum, int> diff = 0;// Define a GlobalAcc.
diff += 1;// Perform an aggregation operation, that is, diff = Sum (0,1)
// diff = 1
diff = 2 x 3;// Assign a value, that is, diff = 2 x 3
// diff = 6
GlobalAcc<Sum, int> g2 = 6;// Define a GlobalAcc.
diff += g2;// Perform an aggregation operation, that is, diff = Sum(6,g2)
// diff = 12
```

GlobalAcc can be operated by expressions. Currently, numeric and GlobalAcc expressions can be used for aggregation and assignment.

```
GlobalAcc<Sum, int> alpha = 0;
GlobalAcc<Sum, int> beta= 10;
alpha = beta*2+3;
// alpha = beta x 2 + 3 = 10 x 2 + 3 = 23
```

## Operator Introduction

### Match matcher operators

After a **Match matcher** is defined, different operation operators, such as repeat, gather, update, and pattern, can be used to assist graph calculation and query.

The operation operator varies with the match type.

| Operator        | Description  | Matcher | Anonymous Expression | Select/Loop Statement |
|-----------------|--|---------|----------------------|-----------------------|
| repeat          | Performs multi-hop filtering like Gremlin.                 | Vertex  | ×                    | ×                     |
| pick            | Randomly selects vertices.                                 | Vertex  | ×                    | ×                     |
| pattern(2.3.11) | Runs Cypher statements.                                    | Vertex  | ×                    | ×                     |
| update          | Performs a set of batch operations defined on the matcher. | Vertex  | √                    | √                     |



| Operator             | Description  | Matcher | Anonymous Expression | Select/Loop Statement |
|----------------------|--|---------|----------------------|-----------------------|
| gather               | Performs a set of batch operations defined on the matcher.   | Vertex  | √                    | √                     |
| init(2.3.12)         | Initializes the matching vertex set based on a specified ID. | Vertex  | ×                    | ×                     |
| insert(2.3.12)       | Add new vertices to the matcher.                             | Vertex  | ×                    | ×                     |
| move(2.3.12)         | Quickly moves other matcher vertex sets.                     | Vertex  | ×                    | ×                     |
| intersection(2.3.12) | Calculates the intersection of vertices.                     | Vertex  | ×                    | ×                     |
| subgraph(2.3.12)     | Obtains guidance subgraphs.                                  | Vertex  | ×                    | ×                     |

- **repeat**

It allows you to perform multi-hop filtering query. The syntax of repeat is similar to that of repeat in Gremlin. Its semantic expression ability is rich, especially suitable for relational link query.

For example, if a two-hop query of filtering name=peter is executed from vertex (1,2) to the outgoing direction, it may be expressed in Gremlin as:

```
g.V('1','2').repeat(out().has('name','peter')).times(2).emit().dedup()
```

In DSL, the preceding functions can be written as follows:

```
Match<Vertex> v(['1','2']);  
v.repeat(outV().has('name','peter')).times(2).emit();
```

- The repeat step contains some unique associated steps:

| Parameter | Mandatory | Type           | Default Value | Description               |
|-----------|-----------|----------------|---------------|---------------------------|
| repeat    | Yes       | traversal step | None          | Executes the repeat rule. |

| Parameter | Mandatory | Type | Default Value | Description   |
|-----------|-----------|------|---------------|---|
| times     | No        | int  | 2             | Maximum number of steps. The default value is <b>2</b> , and the maximum value is <b>20</b> .   |
| emit      | No        | bool | false         | Whether all elements will be returned. The default value is <b>false</b> . If <b>select-as</b> or <b>path</b> is set for output, this parameter determines whether vertices that are not on the final complete path will be returned. |
| limit     | No        | int  | 10000         | Number of vertices, edges, or paths.  |

- Rules in **repeat** consist of traversal and filter criteria. They can be multiple rules separated by commas (,).

**Table 5-329** Traversal process description

| Step   | Description                               |
|--------|---|
| outV   | Neighbor vertex in the outgoing direction |
| inV    | Neighbor vertex in the ingoing direction  |
| bothV  | Neighbor vertex in both directions        |
| outE   | Edge in the outgoing direction            |
| inE    | Edge in the ingoing direction             |
| bothE  | Edge in both directions                   |
| otherV | Neighborhood vertex                       |

**Table 5-330** Filter criteria description

| Filter Criterion | Description                          |
|------------------|--------------------------------------|
| has(key)         | Whether the property name key exists |

| Filter Criterion                          | Description   |
|---|---|
| has(key, value)                           | Whether the value of the property name <b>key</b> is <b>value</b>                                 |
| hasLabel(values)<br>(V2.3.5)              | Whether the label value is one of <b>values</b>   |
| and(filter operator A, filter operator B) | Logical operator of filter criteria. Criteria A and B must be both met. They can be nested.       |
| or(filter operator A, filter operator B)  | Logical operator of filter criteria. Either criterion A or B needs to be met. They can be nested. |

```
and(has('person'), or(has('name', 'peter'), has('age', '30')))
```

```
has('person') // The property name person exists.
has('name', 'peter') // The value of the property name name is peter.
hasLabel('movie','user') // The label value is movie or user.
and( has('name', 'peter'), has('age', '30')) // The property name is peter and age is 30.
```

- **update**

```
$match_vertex.update($lambda_func);
```

The update operator is used to define a group of batch operations on the matcher. Currently, only Match<Vertex> is supported.

The Update operation on Match<Vertex> applies all operations defined in the input lambda function to the vertices matched by Match.

The vertex matcher Match<Vertex> receives only lambda expressions that contain one input parameter.

```
Match<Vertex> v(['1','2']);
VertexAcc<Max,int> acc=1;
GlobalAcc<Sum,int> g=0;
v.update( (v1)->{g+=v1.acc*2;}); //g='1'.acc*2+'2'.acc*2=1*2+1*2=4
return g; // Return g = 4;
```

- **gather**

```
$match_vertex.gather($lambda_func);
```

The gather operator is used to define a group of batch operations on the matcher. Currently, only Match<Vertex> is supported.

- **gather of Match<Vertex>**

The Gather operation on Match<Vertex> applies all operations defined in the input lambda function to the edges of the vertices matched by Match.

The vertex matcher Match receives only lambda expressions that contain two input parameters. The first parameter indicates the source vertex on the edge, and the second parameter indicates the target vertex on the edge.

```
Match<Vertex> v(['1','2']);
VertexAcc<Max,int> acc=1;
GlobalAcc<Sum,int> g=0;
v.gather( (s,t)->{g+=s.acc*2+t.acc;}); // This operation takes effect on the edges of vertices 1 and 2. That is, 1-2, 1-3, 2-3, 2-1.
//g=g + 1.acc*2+2.acc + 1.acc*2+3.acc + 2.acc*2+1.acc + 2.acc*2+3.acc
//g=0 + 3 + 3 + 3 + 3=12
return g;
```

- **pick**  
`$match_vertex.pick(n);`  
**pick** allows you to randomly select  $N$  vertices from `Match<Vertex>`. This function helps you quickly obtain  $N$  random vertices.  
`Match<Vertex> v();  
v.pick(10); // Select 10 vertices randomly from v.  
return v; // Return information about 10 random vertices.`
- **init(2.3.12)**  
`$match_vertex.init([vertex_list]); // Specify the vertex ID list.  
$match_vertex.init(vertexid); // Specify a single ID.`  
Resets the vertex set of the vertex matcher.  
`Match<Vertex> v(['1','2']);  
return v; // Return information about vertices 1 and 2.  
Match<Vertex> v(['1','2']);  
v.init(['3','4']);  
return v; // Return information about vertices 3 and 4.`
- **insert(2.3.12)**  
`$match_vertex.insert([vertex_list]); // Specify a list of vertex IDs.  
$match_vertex.insert(vertexid); // Specify a single ID.`  
Adds a matching vertex set to the vertex matcher.  
`Match<Vertex> v(['1','2']);  
return v; // Return information about vertices 1 and 2.`  
`Match<Vertex> v(['1','2']);  
v.insert(['3','4']);  
return v; // Return information about vertices 1, 2, 3, and 4.`
- **move(2.3.12)**  
`$match_vertex.move(other_match_vertex);`  
Quickly moves the vertex set in **other\_match\_vertex** to the **match\_vertex** matcher. This operator can assign a value to another matcher by one matcher, but does not produce a replication result. Instead, it is similar to the move semantics in C++, that is, moving the original matcher vertex set to the new matcher. After the operation, the original matcher vertex set is cleared.  
`Match<Vertex> v1(['1','2']);  
Match<Vertex> v2(['3','4']);  
v1.move(v2); // v1=[3,4], v2=[]  
return v1; // Return information about vertices 3 and 4.`
- **subgraph(2.3.12)**  
`$match_vertex.subgraph(); // Return guidance subgraphs of the matcher vertex set.  
$match_vertex.subgraph(filter_step); // Return guidance subgraphs of the matcher vertex set with edge filtering.`  
It is mainly used in the return statement. It can return the induced subgraph of the **match\_vertex** vertex matcher.  
When obtaining a subgraph, you can set the filter criterion **filter\_step** on the edge. For details about the syntax, see the filter operator in **repeat**.  
`Match<Vertex> v(['1','2']);  
return v.subgraph(); // Return vertices 1 and 2 as well as the edge set between vertices 1 and 2.`  
`Match<Vertex> v(['1','2','3']);  
return v.subgraph(has(year, 2022)); // Return vertices 1 and 2 as well as the edge set between vertices 1 and 2.`
- **Set basic operations (2.3.12)**  
Set operations can be performed on objects in each matcher, such as union and intersection.  
intersection (2.3.12)

```
Match<Vertex> start(['1','2']);
Match<Vertex> target(['2']);
Match<Vertex> set;
start.intersection(target);
return start;// Return vertex 2.
```

- **Pattern Matching: pattern (2.3.11)**

DSL uses the Cypher syntax to express pattern matching, such as vertex set, path, and subgraph matching.

Statements that support Cypher:

| Statement      | Supported or Not    |
|----------------|---------------------|
| match(2.3.11)  | Partially supported |
| limit(2.3.11)  | Supported           |
| return(2.3.11) | Supported           |

DSL uses the step-pattern of the Match variable to implement pattern matching.

For example, you need to use a certain rule to match vertices.

```
Match<Vertex> v;
v.pattern('match (n:user) where n.age>30 return n limit 10');
```

The pattern is a complete Cypher statement, which filters 10 vertices and returns them to **Match<Vertex>**.

### String in Cypher

The pattern syntax contains a complete Cypher statement in single quotation marks. When a string to be expressed, such as an ID or property value, appears in the Cypher statement, double quotation marks are required.

When using REST APIs to call DSL, you need to add the escape character \ to the double quotation marks. For example:

```
Match<Vertex> v;
v.pattern('match (n) where id(n)="12\" return n');
return v;
```

## 5.1.18 Updating Specified Properties of Vertices and Edges by Importing a File

### Function

This API is used to update specified properties of vertices and edges by importing a file.

#### NOTE

To prevent failures in restoring the updated graph data during system restarting, do not delete the data stored on OBS when the graph is in use.

### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=import-properties

**Table 5-331** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Example Request

Import a file to update specified properties of vertices and edges. The vertex file directory is **datasets/movie/movie.csv**, and the vertex data set format is CSV. The edge file directory is **datasets/movie/ranking\_edge.csv**, and the edge data set format is CSV.

```
POST http://Endpoint/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties
{
  "vertexsetPath": "datasets/movie/movie.csv",
  "vertexsetFormat": "csv",
  "vertexProperties": [
    {
      "label": "movie",
      "properties": [
        "genres"
      ]
    }
  ],
  "edgesetPath": "datasets/movie/ranking_edge.csv",
  "edgesetFormat": "csv",
  "edgeProperties": [
    {
      "label": "rate",
      "properties": [
        "Datetime"
      ]
    }
  ],
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ],
  "delimiter": ",",
  "trimQuote": "\"",
  "obsParameters": {
    "accessKey": "XXXXXXX",
    "secretKey": "XXXXXXX"
  },
  "vertexFileContainLabel": true
}
```

## Request Parameters

 NOTE

The format of the CSV file for updating properties is as follows:

- Vertex file (including **label**): vertex ID,**label**,**property\_1...property\_n**
- Vertex file (excluding **label**): vertex ID,**property\_1...property\_n**
- Edge file: source vertex ID, target vertex ID, label, edge ID, **property\_1...property\_n**

**Table 5-332** Request body parameters

| Parameter        | Mandatory   | Type      | Description  |
|------------------|---|-----------|--|
| vertexsetPath    | Either <b>vertexsetPath</b> or <b>edgesetPath</b> is mandatory. | String    | Vertex file directory or name  |
| vertexsetFormat  | No  | String    | Format of the vertex data set. Currently, only the CSV format is supported. The CSV format is used by default.                                 |
| vertexProperties | Mandatory if <b>vertexsetPath</b> exists                        | Object    | Label of a vertex and list of properties to be updated in a vertex file, in JSONArray format   |
| edgesetPath      | Either <b>vertexsetPath</b> or <b>edgesetPath</b> is mandatory. | String    | Edge file directory or name  |
| edgesetFormat    | No  | String    | Format of the edge data set. Currently, only the CSV format is supported. The CSV format is used by default.                                   |
| edgeProperties   | Mandatory if <b>edgesetPath</b> exists                          | Object    | Label of an edge and list of properties to be updated in an edge file, in JSONArray format   |
| targetProperties | Mandatory if <b>edgesetPath</b> exists                          | Object    | Indicates property information used to distinguish duplicate edges in the edge file, in JSONArray format.                                      |
| delimiter        | No  | Character | Field separator in a CSV file. The default value is comma (.). The default element separator in a field of the list/set type is semicolon (;). |

| Parameter               | Mandatory | Type      | Description  |
|-------------------------|-----------|-----------|--|
| trimQuote               | No        | Character | Field quote character in a CSV file. The default value is double quotation marks ("). It is used to enclose a field if the field contains separators or line breaks. |
| obsParameters           | Yes       | String    | OBS authentication parameters. For details, see <a href="#">Table 5-272</a> .  |
| vertexFileContainsLabel | No        | Boolean   | Whether the vertex file contains label information. This parameter is optional. The default value is <b>true</b> .   |

**Table 5-333** vertexProperties parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| label      | Yes       | String | Name of a label  |
| properties | Yes       | Object | Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the vertex file. |

**Table 5-334** edgeProperties parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| label      | Yes       | String | Name of a label  |
| properties | Yes       | Object | Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the edge file. |

**Table 5-335** targetProperties parameter description

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| label      | Yes       | String | Name of a label   |
| properties | Yes       | Object | Edge ID properties, in JSONArray format. Currently, only one property is supported. |



## Response Parameters

**Table 5-336** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| job_id       | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                 |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter format error",
  "errorCode": "GES.8013"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |
| 404 Not Found    | No resources found.       |

| Return Value              | Description            |
|---------------------------|------------------------|
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

## 5.1.19 Deleting Vertices and Edges by Reading Files

### Function

This API is used to delete vertices and edges by reading the files.

### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=delete-by-file

**Table 5-337** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

### Example Request

Delete vertices and edges by reading files. The vertex file directory is **datasets/movie/movie.csv**, and the vertex data set format is CSV. The edge file directory is **datasets/movie/ranking\_edge.csv**, and the edge data set format is CSV.

```
POST http://Endpoint/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file
{
  "vertexsetPath": "datasets/movie/movie.csv",
  "vertexsetFormat": "csv",
  "edgesetPath": "datasets/movie/ranking_edge.csv",
  "edgesetFormat": "csv",
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ],
  "delimiter": ",",
  "trimQuote": "\""
```

```
"obsParameters": {
  "accessKey": "XXXXXXXX",
  "secretKey": "XXXXXXXX"
}
```

## Request Parameters

**Table 5-338** Request body parameters

| Parameter        | Mandatory   | Type      | Description   |
|------------------|---|-----------|---|
| vertexsetPath    | Either <b>vertexsetPath</b> or <b>edgesetPath</b> is mandatory. | String    | Vertex file directory or name   |
| vertexsetFormat  | No  | String    | Format of the vertex data set. Currently, only the CSV format is supported. The CSV format is used by default.  |
| edgesetPath      | Either <b>vertexsetPath</b> or <b>edgesetPath</b> is mandatory. | String    | Edge file directory or name   |
| edgesetFormat    | No  | String    | Format of the edge data set. Currently, only the CSV format is supported. The CSV format is used by default.  |
| targetProperties | No  | Object    | Indicates property information used to distinguish duplicate edges in the edge file, in JSONArray format. For details, see <a href="#">Table 5-339</a> .                |
| delimiter        | No  | Character | Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).                          |
| trimQuote        | No  | Character | Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks. |

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| obsParameters | Yes       | String | OBS authentication parameters. For details about the parameters, see <a href="#">Table 5-272</a> . |

**Table 5-339** targetProperties parameter description

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| label      | Yes       | String | Name of a label   |
| properties | Yes       | Object | Edge ID properties, in JSONArray format. Currently, only one property is supported. |

 **NOTE**

The format of the CSV file for updating properties is as follows:

- Vertex file: Vertex ID
- Edge file (excluding the label): source vertex ID and destination vertex ID
- Edge file (including the label): source vertex ID, destination vertex ID, label, and edge ID

## Response Parameters

**Table 5-340** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Querying Job Status on the Service Plane</a> .               |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter format error",
  "errorCode": "GES.8013"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.1.20 Granular Permission Control APIs

### 5.1.20.1 Authorization

#### Function

GES graph instances support granular permission control. The granularity is refined to the traverse, read, and write permissions set for specific properties of specific labels. The following table describes the GES granular permission control data model. This API is used to grant permissions to a user.

| Access | Operation | Object | Graph Instance | Application Scope | Scenario |
|--------|-----------|--------|----------------|-------------------|----------|
|--------|-----------|--------|----------------|-------------------|----------|

|                |          |  |        |                                     |  |
|----------------|----------|--|--------|-------------------------------------|--|
| Grant / Revoke | traverse | LABEL (* indicates all labels.)                      | Single | -                                   | Vertex: External IDs and label names of vertices can be accessed.<br>Edge: Two vertices have the traverse permission to access the edge ID and label name. |
| Grant / Revoke | read     | PROPERTY (Separate multiple values with commas (,)). | Single | Label Name (* indicates all labels) | The traverse permission is required.   |
| Grant / Revoke | write    | PROPERTY (Separate multiple values with commas (,)). | Single | Label Name (* indicates all labels) | The traverse permission is required.   |
| Grant / Revoke | schema   | GRAPH  | Single | graph                               | Metadata management (modifying property names, clearing schemas, and importing schemas)  |

 NOTE

- The granular permission control APIs are available only when RBAC is enabled for the created graph instance. For details, see the [Creating a Graph](#). You need to add the **enableRBAC** parameter and set it to **true** when you call the graph creation API.
- To authorize granular permission, you must be a . To call the granular permission APIs, the token obtained for accessing a domain is required..
- Users with the traverse permission can view all vertices and edges with the same label, but cannot view the properties of these vertices and edges.

## URI

- URI format  
POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/rbac/action?action\_id=grant
- Parameter description

**Table 5-341** URI parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

| Parameter  | Mandatory | Type   | Description |
|------------|-----------|--------|-------------|
| graph_name | Yes       | String | Graph name  |

## Request

- Request example

POST `http://{{SERVER_URL}}/ges/v1.0/{project_id}/graphs/{graph_name}/rbac/action?action_id=grant`

```
{
  "userId": "test1",
  "acl": [
    {
      "type": "read",
      "detail": [
        {
          "label": "person",
          "properties": ["crime", "Occupation"]
        },
        {
          "label": "inmate"
        }
      ]
    },
    {
      "type": "write",
      "detail": [
        {
          "label": "person",
          "properties": ["crime", "Occupation"]
        },
        {
          "label": "inmate"
        },
        {
          "label": "Leader"
        },
        {
          "label": "Friend"
        },
        {
          "label": "Colleague"
        },
        {
          "label": "QQGroup"
        },
        {
          "label": "QQ_owner"
        },
        {
          "label": "QQ"
        },
        {
          "label": "phone"
        },
        {
          "label": "Phone_owner"
        }
      ]
    },
    {
      "type": "traverse",
      "detail": [
        {
          "label": "person"
        },
        {
          "label": "inmate"
        },
        {
          "label": "Leader"
        },
        {
          "label": "Friend"
        },
        {
          "label": "Colleague"
        }
      ]
    }
  ]
}
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

- Request body parameter description

**Table 5-342** Request body parameter description

| Parameter  | Mandatory | Type      | Description   |
|------------|-----------|-----------|---|
| graph_name | Yes       | String    | Graph name  |
| userId     | Yes       | String    | ID of the authorized user.  |
| acl        | Yes       | JsonArray | Authorization details   |
| type       | Yes       | String    | Permission type. The value can be <b>read</b> , <b>write</b> , <b>traverse</b> , or <b>schema</b> . |
| detail     | Yes       | JsonArray | Permission details  |
| label      | Yes       | String    | Label name  |
| properties | No        | List      | Properties  |

## Response

- Parameter description

**Table 5-343** Parameter description

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| errorMessage | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |

- Response example (successful request)

Http Status Code: 200

- Response example (failed request)

**Status code: 400**

Example response (failed request)

Http Status Code: 400

```
{
  "errorMessage": "grant acl is null",
  "errorCode": "GES.8503"
}
```



## Status Code

| Response Code             | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.20.2 Canceling Authorization

#### Function

This API is used to cancel the authorization.

#### URI

- URI format  
POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/rbac/action?action\_id=revoke
- Parameter description

**Table 5-344** URI parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request

- Request example  
POST http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/rbac/action?action\_id=revoke  
{  
  "userId": "test1",  
  "acl": [{  
    "type": "read",  
    "detail": [{  
      "label": "person",

```

"properties": ["crime", "Occupation"]
}, {
"label": "inmate"
}]
}, {
"type": "write",
"detail": [{
"label": "person",
"properties": ["crime", "Occupation"]
}, {
"label": "inmate"
}, {
"label": "Leader"
}, {
"label": "Friend"
}, {
"label": "Colleague"
}, {
"label": "QQGroup"
}, {
"label": "QQ_owner"
}, {
"label": "QQ"
}, {
"label": "phone"
}, {
"label": "Phone_owner"
}]
}, {
"type": "traverse",
"detail": [{
"label": "person"
}, {
"label": "inmate"
}, {
"label": "Leader"
}, {
"label": "Friend"
}, {
"label": "Colleague"
}]
}]
}

```

 NOTE

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- Request body parameter description

**Table 5-345** Request body parameter description

| Parameter  | Mandatory | Type      | Description   |
|------------|-----------|-----------|---|
| graph_name | Yes       | String    | Graph name  |
| userId     | Yes       | String    | ID of the authorized user.  |
| acl        | Yes       | JSONArray | Authorization details   |
| type       | Yes       | String    | Permission type. The value can be <b>read</b> , <b>write</b> , <b>traverse</b> , or <b>schema</b> . |

| Parameter  | Mandatory | Type      | Description        |
|------------|-----------|-----------|--------------------|
| detail     | Yes       | JsonArray | Permission details |
| label      | Yes       | String    | Label name         |
| properties | No        | List      | Properties         |

## Response

- Parameter description

**Table 5-346** Parameter description

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| errorMessage | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | No        | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |

- Response example (successful request)

Http Status Code: 200

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "grant acl is null",
  "errorCode": "GES.8503"
}
```

## Status Code

| Response Code             | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

### 5.1.20.3 Querying Authorization

#### Function

This API is used to query all label and property permissions of the current user.

#### URI

- URI format  
GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/rbac
- Parameter description

**Table 5-347** URI parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request

- Request example  
GET http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/rbac

#### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

#### Response

```
{
  "data": {
    "acl": [
      {
        "detail": [],
        "type": "traverse"
      },
      {
        "detail": [
          {
            "label": "movie",
            "properties": [
              "movieid",
              "title"
            ]
          }
        ]
      }
    ]
  }
}
```

```

    "label": "user",
    "properties": [
      "gender",
      "age",
      "userid"
    ]
  },
  "type": "read"
},
{
  "detail": [],
  "type": "write"
}
]
}
}

```

## Status Code

| Response Code             | Description                           |
|---------------------------|---------------------------------------|
| 400 Bad Request           | Request error.                        |
| 401 Unauthorized          | Authentication failed.                |
| 403 Forbidden             | No operation permission.              |
| 404 Not Found             | The requested resource was not found. |
| 500 Internal Server Error | Internal service error.               |
| 503 Service Unavailable   | Service unavailable.                  |

## Error Code

See [Error Code](#).

## 5.1.21 O&M Monitoring APIs

### 5.1.21.1 Viewing Monitoring Metrics

#### Function

This API is used to view monitoring metrics, including node metrics and graph instance performance monitoring metrics.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/om/metrics?  
real\_time=&with\_performance\_metrics=

**Table 5-348** URI parameters

| Parameter                | Mandatory | Type    | Description   |
|--------------------------|-----------|---------|---|
| project_id               | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name               | Yes       | String  | Graph name  |
| real_time                | No        | Boolean | Whether to query real-time monitoring metrics. The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> . <ul style="list-style-type: none"> <li><b>false</b>: Metrics for a graph instance within 2 minutes are queried.</li> <li><b>true</b>: Real-time monitoring metrics are queried, and the query is responded to in 3 to 5 seconds.</li> </ul> |
| with_performance_metrics | No        | Boolean | Whether to query performance metrics. The graph instance performance metrics and metrics of each node are returned. The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> . If set to <b>true</b> , only node metrics are returned, and the response time is reduced by 1 to 2 seconds.   |

## Request Parameters

None

## Response Parameters

**Table 5-349** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| project_id   | String | Schema structure query result  |

| Parameter           | Type              | Description         |
|---------------------|-------------------|---------------------|
| id                  | String            | Graph ID            |
| name                | String            | Graph name          |
| timestamp           | long              | Current timestamp   |
| node_metrics        | List<NodeMetrics> | Node metrics        |
| performance_metrics | Object            | Performance metrics |

**Table 5-350** NodeMetrics parameter description

| Parameter       | Type                | Description                 |
|-----------------|---------------------|-----------------------------|
| overview        | String              | Node overview               |
| disk_details    | List<DiskDetail>    | Disk details of the node    |
| network_details | List<NetworkDetail> | Network details of the node |

**Table 5-351** overview parameter description

| Parameter         | Type   | Description                       |
|-------------------|--------|-----------------------------------|
| ges_instance_name | String | Node name                         |
| instance_id       | String | Node ID                           |
| work_ip           | String | Number of vertices with the label |
| role              | String | Node role                         |
| cpu_usage         | Double | CPU usage                         |
| cpu_usage_usr     | Double | CPU usage in user mode            |
| cpu_usage_sys     | Double | CPU usage in kernel mode          |
| cpu_iowait        | Double | CPU I/O wait rate                 |
| cpu_idle          | Double | CPU idle rate                     |
| mem_total         | Double | Total memory size, in GB          |
| mem_usage         | Double | Used memory size, in GB           |
| mem_free          | Double | Available memory size, in GB      |

| Parameter       | Type    | Description                         |
|-----------------|---------|-------------------------------------|
| mem_cached      | Double  | Memory cache size, in GB            |
| mem_buffer      | Double  | Memory buffer size, in GB           |
| disk_total      | Double  | Total disk space, in GB             |
| disk_usage_avg  | Double  | Average disk usage                  |
| disk_used       | Double  | Used disk space, in GB              |
| disk_available  | Double  | Total available disk space, in GB   |
| disk_io_read    | Double  | Disk read rate of a node, in KB/s   |
| disk_io_write   | Double  | Disk read rate of a node, in KB/s   |
| swap_total      | Double  | Total swap disk space of a node     |
| swap_free       | Double  | Remaining swap disk space of a node |
| network_io_rate | Double  | Network I/O rate of a node, in KB/s |
| host_stat       | Integer | Node status                         |

**Table 5-352** DiskDetail parameter description

| Parameter       | Type    | Description                            |
|-----------------|---------|--|
| disk_name       | String  | Disk name                              |
| disk_type       | String  | Disk type                              |
| total           | Double  | Total disk space                       |
| available       | Double  | Available disk space                   |
| used            | Double  | Used disk space                        |
| used_percentage | Integer | Disk usage                             |
| svctm           | Long    | Disk I/O service time, in milliseconds |
| await           | Long    | Disk I/O wait time, in milliseconds    |
| util            | Double  | Disk I/O usage                         |
| write_rate      | Double  | Disk read rate                         |
| read_rate       | Double  | Disk write rate                        |



**Table 5-353** NetworkDetail parameter description

| Parameter      | Type    | Description                |
|----------------|---------|----------------------------|
| status         | Integer | NIC status                 |
| interface_name | String  | NIC name                   |
| packets_recv   | Long    | Number of received packets |
| packets_send   | Long    | Number of sent packets     |
| packets_drop   | Long    | Number of lost packets     |
| send_rate      | Double  | Sending rate, in KB/s      |
| recv_rate      | Double  | Receiving rate, in KB/s    |

**Table 5-354** performance\_metrics parameter description

| Parameter                 | Type   | Description                      |
|---------------------------|--------|----------------------------------|
| cpu_usage                 | Double | CPU usage                        |
| memory_usage              | Double | Memory usage                     |
| disk_usage                | Double | Average disk usage               |
| disk_io_rate              | Double | Disk I/O rate                    |
| network_io_rate           | Double | Network I/O rate                 |
| swap_disk_usage           | Double | Swap disk usage                  |
| tomcat_connections_usage  | Double | Tomcat connection usage          |
| qps                       | Long   | Number of requests per second    |
| vertex_number             | Long   | Number of vertices               |
| vertex_capacity           | Long   | Vertex capacity                  |
| vertex_usage              | Double | Vertex usage                     |
| edge_number               | Long   | Number of edges                  |
| edge_capacity             | Long   | Edge capacity                    |
| edge_usage                | Double | Edge usage                       |
| read_waiting_queue_length | Long   | Length of the read waiting queue |
| read_running_queue_length | Long   | Length of the read running queue |

| Parameter                  | Type | Description                       |
|----------------------------|------|-----------------------------------|
| write_waiting_queue_length | Long | Length of the write waiting queue |
| write_running_queue_length | Long | Length of the write running queue |

## Example Request

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?real_time=true
```

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "project_id": "xxx",
  "id": "xxxxx",
  "name": "baiwan_demo",
  "timestamp": 1699506387592,
  "node_metrics": [
    {
      "overview": {
        "network_io_rate": 2.99,
        "role": "slave",
        "disk_io_write": 107.23,
        "mem_cached": 2.57,
        "cpu_usage_usr": 7.15,
        "cpu_usage_sys": 3.07,
        "disk_io_read": 3.03,
        "ges_instance_name": "baiwan_demo-ges-cn-cn-2-1",
        "disk_used": 5.48,
        "swap_total": 0,
        "mem_buffer": 192.5,
        "disk_available": 144.47,
        "cpu_iowait": 0.17,
        "cpu_idle": 89.61,
        "mem_total": 15.15,
        "instance_id": "xxxxxxxx",
        "mem_usage": 7.22,
        "disk_total": 149.95,
        "host_stat": 200,
        "mem_free": 11.29,
        "swap_free": 0,
        "cpu_usage": 10.22,
        "disk_usage_avg": 3.65,
        "work_ip": "172.16.25.224",
        "host_name": "baiwan_demo-ges-cn-cn-2-1"
      },
      "disk_details": [
        {
          "svctm": 0,
          "total": 50,
          "util": 0.61,
          "write_rate": 96.48,
          "disk_name": "vda",
          "disk_type": "system",
          "used_percentage": 0.09,
          "available": 45.68,
          "await": 18.16,
          "read_rate": 2.99,
        }
      ]
    }
  ]
}
```

```
    "used": 4.32
  }
],
"network_details": [
  {
    "send_rate": 0.05,
    "packets_drop": 0,
    "packets_rcv": 1001419,
    "packets_send": 342518,
    "interface_name": "eth0",
    "rcv_rate": 0.06,
    "status": 1
  }
]
}
},
"performance_metrics": {
  "tomcat_connections_usage": 0,
  "network_io_rate": 3.11,
  "swap_disk_usage": 0,
  "vertex_capacity": 1200000,
  "memory_usage": 23.83,
  "vertex_number": 1071803,
  "jvm_heap_usage": 0.23,
  "edge_capacity": 1200000,
  "read_waiting_queue_length": 0,
  "disk_io_rate": 0,
  "qps": 0,
  "write_running_queue_length": 0,
  "write_waiting_queue_length": 0,
  "disk_usage": 1.77,
  "edge_number": 1200000,
  "edge_usage": 100,
  "cpu_usage": 9.23,
  "read_running_queue_length": 0,
  "vertex_usage": 89.32
}
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "query metrics error.",
  "errorCode": "GES.8602"
}
```

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.1.21.2 Viewing Real-Time Requests

#### Function

This API is used to view the real-time requests on the current primary node.

#### URI

GET/ges/v1.0/{project\_id}/graphs/{graph\_name}/om/real-time-queries?summary=

**Table 5-355** URI parameters

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| project_id | Yes       | String | Project ID. Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name | Yes       | String | Graph name  |
| summary    | No        | Bool   | Whether to query only the summary information about real-time requests. The default value is <b>false</b> . If set to <b>true</b> , only the summary information is returned. |

## Request Parameters

None

## Response Parameters

**Table 5-356** Response body parameters

| Parameter       | Type   | Description  |
|-----------------|--------|--|
| errorMessage    | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode       | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| instance_name   | String | Instance name  |
| edge_capacity   | Long   | Edge capacity  |
| edge_number     | Long   | Number of edges  |
| edge_usage      | Double | Edge usage   |
| vertex_capacity | String | Vertex capacity  |
| vertex_number   | Long   | Number of vertices   |
| vertex_usage    | Long   | Vertex usage   |

| Parameter                  | Type               | Description                       |
|----------------------------|--------------------|-----------------------------------|
| read_waiting_queue_length  | Long               | Length of the read waiting queue  |
| read_running_queue_length  | Long               | Length of the read running queue  |
| write_waiting_queue_length | Long               | Length of the write waiting queue |
| write_running_queue_length | Long               | Length of the write running queue |
| current_queries            | List<CurrentQuery> | Details of the current query list |

**Table 5-357** CurrentQuery parameter description

| Parameter        | Type   | Description  |
|------------------|--------|--|
| task_name        | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul> |
| request_id       | String | Request ID   |
| status           | String | Request execution status   |
| request          | String | Request parameter  |
| running_duration | Double | Request execution duration, in seconds   |
| pending_duration | Double | Request blocking duration, in seconds  |
| begin_time       | String | Request start time   |
| progress         | Double | Request execution progress   |

## Example Request

View real-time requests.

GET [http://{{SERVER\\_URL}}/ges/v1.0/{project\\_id}/graphs/{graph\\_name}/om/real-time-queries](http://{{SERVER_URL}}/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries)

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "instance_name": "ges_demo-ges-dn-2-1",
  "edge_capacity": 0,
  "edge_number": 0,
  "edge_usage": 0,
  "vertex_capacity": 0,
  "vertex_number": 0,
  "vertex_usage": 0,
  "read_waiting_queue_length": 2,
  "read_running_queue_length": 1,
  "write_waiting_queue_length": 0,
  "write_running_queue_length": 0,
  "current_queries": [
    {
      "task_name": "incremental_load_graph",
      "request": {
        "vertex_file_format": "csv",
        "parallel_edge": "allow",
        "graph_name": "ges_demo",
        "vertex_file_path": "wepbucket/movie1/sit_ges_data_uat/nodes/",
        "edge_file_path": "wepbucket/movie1/sit_ges_data_uat/edges/",
        "schema_file_path": "wepbucket/movie1/sit_ges_data_uat/sit_ges_metadata_v6.xml",
        "vidSerialize": true,
        "offline": false,
        "trim_quote": "\\\"",
        "ignore_label": true,
        "delimiter": ",",
        "edge_file_format": "csv",
        "parameters": {
          "secret_key": "xxxxxxx",
          "access_key": "xxxxxxx",
          "region": "xxx"
        }
      },
      "pending_duration": 9.91311,
      "request_id": "0c56e2d14369586da38d7fe3b81bb1bd",
      "status": "pending",
      "begin_time": "",
      "running_duration": 0,
      "progress": 0
    }
  ]
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "query metrics error.",
  "errorCode": "GES.8602"
}
```

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 5.2 Database Edition

### 5.2.1 Specification Description

As a new graph database product, the database edition provides graph management, graph data addition, deletion, modification, query, and analysis

capabilities, supports Cypher, and provides graph data persistence to flush data written by users to disks. In this way, multiple copies and hardware redundancy are used to achieve high availability and fast fault recovery. Currently, the database edition has completed storage and query of hundreds of billions and trillions of graphs.

The following sections describe the APIs supported by database graphs.

## Data Types

| Type   | Description   |
|--------|---|
| char   | Character   |
| float  | Float type (32-bit float)   |
| double | Double type (64-bit float)  |
| bool   | Boolean type. Available values are <b>0/1</b> and <b>true/false</b> .   |
| long   | Long integer (value range: $-2^{63}$ to $2^{63}-1$ )  |
| int    | Integer (value range: $-2^{31}$ to $2^{31}-1$ )   |
| date   | Date. Currently, the following formats are supported: <ul style="list-style-type: none"> <li>• YYYY-MM-DD HH:MM:SS</li> <li>• YYYY-MM-DD</li> </ul> <b>NOTE</b><br>The value of <i>MM</i> or <i>DD</i> must consist of two digits. If the value contains only one digit, add 0 before it, for example, <b>05-01</b> . |
| string | Variable-length string  |
| enum   | Enumeration type. The maximum number is 65535. Each enumeration type is a string.   |

## Data Import Restrictions

There are some restrictions when you import data to a database edition graph:

- Importing data concurrently  
 Multiple data files can be imported at the same time. To accelerate the import, split a large file into multiple OBS files that each does not exceed 5 GB.
- Uploading import logs to OBS  
 Set the **logDir** parameter to save import logs where you can find the error cause.
- Importing edge data  
 If you import only edges of a database edition graph, DLI cannot automatically generate vertices for the graph. If only edges are imported, you cannot query vertices or access the graph from vertices. You are advised to import vertex data too.

## 5.2.2 Vertex Operation APIs

### 5.2.2.1 Querying Vertex Details

#### Function

This API is used to query the vertex information (such as the label and property) based on the vertex ID.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/detail?  
vertexIds={vertex\_ids}

**Table 5-358** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| vertex_ids | Yes       | String | IDs of the vertices to be queried. You can specify only one vertex ID for a database edition graph.      |

#### Request Parameters

**Table 5-359** Request body parameter

| Parameter | Type | Description  |
|-----------|------|--|
| data      | List | Vertex details you want to query. For details, see <a href="#">data parameters</a> . |

**Table 5-360** data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| vertices  | List | Vertex result set. If no corresponding vertices are found, the value of <b>vertices</b> is empty. |



## Response Parameters

**Table 5-361** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | Object | Query results.  |

## Example Request

Query node information by node ID and return node details.

GET `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds=Ray`

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example of a successful response

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "46",
        "labels": [
          "user"
        ],
        "properties": {
          "user": {
            "userid": [
              0
            ],
            "gender": [
              "F"
            ],
            "age": [
              "25-34"
            ],
            "occupation": [
```

```
    "artist"  
  ],  
  "Zip-code": [  
    "98133"  
  ]  
}  
}  
]  
},  
"result": "success"  
}
```

**Status code: 400**

Example of a failed response

```
Http Status Code: 400  
{  
  "errorMessage": "graph [demo] is not found",  
  "errorCode": "GES.8204"  
}
```

### Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

### Error Codes

See [Error Codes](#).

### 5.2.2.2 Querying Vertices in Batches

#### Function

This API is used to query the vertex data (such as the labels and properties) in batches based on the vertex IDs.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-query

**Table 5-362** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-363** Request body parameter

| Parameter | Mandatory | Type   | Description                              |
|-----------|-----------|--------|--|
| vertices  | Yes       | String | Vertex IDs you use to query the vertices |

## Response Parameters

**Table 5-364** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | Object | The <b>data</b> field is contained when the query is successful, and the <b>data</b> field contains the vertices query result.  |
| result       | String | Query results. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .   |

## Example Request

Query nodes in batches by node ID. The vertex IDs to be queried are **27003509\_Station Building** and **39636392\_Badaling Great Wall**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query
{
  "vertices":
  ["0","1"]
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "0",
        "labels": [
          "movie"
        ],
        "properties": {
          "movie": {
            "movieid": [
              0
            ],
            "title": [
              "American Beauty (1999)"
            ],
            "genres": [
              "Comedy|Drama"
            ]
          }
        }
      },
      {
        "id": "1",
        "labels": [
          "movie"
        ],
        "properties": {
          "movie": {
            "movieid": [
              1
            ],
            "title": [
              "Airplane! (1980)"
            ],
            "genres": [
              "Comedy"
            ]
          }
        }
      }
    ]
  }
}
```

```
"result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.2.2.3 Adding Vertices in Batches**

**Function**

This API is used to add vertices in batches.

**URI**

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-add

**Table 5-365** URI parameters

| Parameter  | Mandato<br>ry | Type   | Description  |
|------------|---------------|--------|--|
| project_id | Yes           | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes           | String | Graph name   |

## Request Parameters

**Table 5-366** Request body parameters

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| vertices       | Yes       | JSON    | Vertices you want to add. A maximum of 10,000 vertices can be added at a time. For details about this array, see the <a href="#">vertices parameters</a> .  |
| overrideExists | No        | Boolean | Checks whether the vertex to be inserted exists. The default value is <b>false</b> . <ul style="list-style-type: none"> <li>If this parameter is set to <b>false</b>, an error is reported as long as there is one vertex, and all vertices fail to be written.</li> <li>If this parameter is set to <b>true</b>, the existing vertices are overwritten.</li> </ul> |

**Table 5-367** vertices parameter description

| Parameter  | Mandatory | Type   | Description            |
|------------|-----------|--------|------------------------|
| vertex     | Yes       | String | Vertex ID              |
| label      | Yes       | String | Vertex label           |
| properties | No        | JSON   | Value of each property |

## Response Parameters

**Table 5-368** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |

| Parameter | Type   | Description  |
|-----------|--------|--|
| result    | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> . |

## Example Request

Add vertices in batches. The names of the vertices to be added are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "label": "movie",
      "properties": {
        "movieid": [
          "6"
        ],
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ],
  "overrideExists": true
}
```

### NOTE

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertex **6** already exists in the graph, properties of vertex **6** are overwritten.

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
```

```
"result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] already exists",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.2.4 Deleting Vertices in Batches

#### Function

This API is used to delete vertices in batches based on the vertex IDs.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/action?  
action\_id=batch-delete

**Table 5-369** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |



## Request Parameters

**Table 5-370** Request body parameters

| Parameter | Mandatory | Type   | Description                   |
|-----------|-----------|--------|-------------------------------|
| vertices  | Yes       | String | Vertex ID array to be deleted |

## Response Parameters

**Table 5-371** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Delete nodes in batches by node ID. The vertex IDs to be deleted are **Vivian** and **46**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete
{
  "vertices": [
    "Vivian",
    "46"
  ]
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.2.5 Updating Vertex Properties in Batches

#### Function

This API is used to update vertex properties in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/properties/action?  
action\_id={actionId}

**Table 5-372** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .   |
| graph_name | Yes       | String | Graph name   |
| actionId   | Yes       | String | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>batch-update</b>: Update the value of a property.</li> <li>• <b>batch-add</b>: Add the value to a property. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of <b>batch-update</b>. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>batch-del</b>: Delete a property value.</li> </ul> |

## Request Parameters

**Table 5-373** Request body parameters

| Parameter | Mandatory | Type | Description  |
|-----------|-----------|------|--|
| vertices  | Yes       | JSON | Vertex array to be updated For details about this array, see the <a href="#">vertices parameters</a> . |

**Table 5-374** vertices parameter description

| Parameter  | Mandatory | Type   | Description                          |
|------------|-----------|--------|--------------------------------------|
| vertex     | Yes       | String | Vertex ID                            |
| label      | No        | String | Vertex label                         |
| properties | Yes       | JSON   | Value of each property to be updated |

## Response Parameters

**Table 5-375** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update vertex properties in batches. The vertex names to be updated are **150** and **6**.

POST http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/vertices/properties/action?  
action\_id=batch-update

```
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "properties": {
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ]
}
```

 NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8220"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.2.3 Edge Operation APIs

### 5.2.3.1 Querying Edge Details

#### Function

This API is used to query the detailed information about an edge based on the source vertex, target vertex, and index of the edge. Information about edges and properties is returned.

## URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source={sourceVertex}&target={targetVertex}&label={label}&sortKey={sortKey}&sortKeyType={sortKeyType}
```

**Table 5-376** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-377** Request body parameters

| Parameter    | Mandatory | Type   | Description  |
|--------------|-----------|--------|--|
| project_id   | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name   | Yes       | String | Graph name   |
| sourceVertex | Yes       | String | Source vertex of an edge   |
| targetVertex | Yes       | String | Target vertex of an edge   |

## Response Parameters

**Table 5-378** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |

| Parameter | Type   | Description  |
|-----------|--------|--|
| data      | Object | Query results. If the query is successful, the query result will be returned. If the query fails, this parameter will be left blank. |
| result    | String | Query result. If the query is successful, the value is <b>success</b> . If the query fails, the value is <b>failed</b> .             |

**Table 5-379** data parameter

| Parameter | Mandatory | Type | Description  |
|-----------|-----------|------|--|
| edges     | Yes       | List | Edge result set. If no edge is found, this parameter will be left blank. |

## Example Request

Query details about an edge.

```
GET/ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source=46&target=39&label=rate&sortKey=5&sortKeyType=int
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
{
  "data": {
    "edges": [
      {
        "source": "46",
        "target": "39",
        "label": "rate",
        "sortKey": 5,
        "properties": {
          "Rating": [
            5
          ],
          "Datetime": [
            "2018-01-0120:30:05"
          ]
        }
      }
    ]
  },
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8107"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.3.2 Querying Edges in Batches

#### Function

This API is used to query the detailed information about edges in batches based on the source vertices, target vertices, and indexes of the edges. Information about edges and properties is returned.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-query

**Table 5-380** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |



## Request Parameters

**Table 5-381** Request body parameter

| Parameter | Mandatory | Type   | Description              |
|-----------|-----------|--------|--------------------------|
| edges     | Yes       | Object | Edge array to be queried |

**Table 5-382** edges parameter description

| Parameter | Mandatory | Type   | Description              |
|-----------|-----------|--------|--------------------------|
| source    | Yes       | String | Source vertex of an edge |
| target    | Yes       | String | Target vertex of an edge |

## Response Parameters

**Table 5-383** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | String | The <b>data</b> field is contained when the query is successful, and the <b>data</b> field contains the edges query result.   |

## Example Request

Query details about edges in batches based on the source vertex, target vertex, and index. The source vertex of the edges to be queried is **Vivian**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query
{
  "edges": [
    {
      "source": "Vivian",
      "target": "Lethal Weapon",
```

```
    "label": "rate"
  },
  {
    "source": "Vivian",
    "target": "Raising Arizona"
  }
]
}
```

 **NOTE**

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
{
  "data": {
    "edges": [
      {
        "source": "Vivian",
        "target": "Raising Arizona",
        "label": "rate",
        "properties": {
          "Score": [
            4
          ],
          "Datetime": [
            "2000-12-27 23:51:42"
          ]
        }
      },
      {
        "source": "Vivian",
        "target": "Lethal Weapon",
        "label": "rate",
        "properties": {
          "Score": [
            5
          ],
          "Datetime": [
            "2000-12-27 23:44:41"
          ]
        }
      }
    ]
  },
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter does not contain 'source'",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.3.3 Adding Edges in Batches

#### Function

This API is used to add edges in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-add

**Table 5-384** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

| Parameter    | Mandatory | Type   | Description                |
|--------------|-----------|--------|----------------------------|
| edges        | Yes       | Object | Edge array to be added     |
| parallelEdge | No        | Object | Repetitive edge processing |

| Parameter       | Mandatory | Type    | Description  |
|-----------------|-----------|---------|--|
| action          | No        | String  | <p>Processing method. The value can be <b>override</b> or <b>ignore</b>.</p> <ul style="list-style-type: none"> <li>If the value is <b>override</b>, the previous repetitive edges are overwritten.</li> <li>If the value is <b>ignore</b> and an edge already exists, the previous edge will not be overwritten. If no edge exists, the system adds an edge.</li> </ul> |
| ignoreLabel     | No        | Boolean | <p>Whether to ignore labels on repetitive edges. The value can only be <b>false</b>.</p> <p><b>false</b>: Edges with the same source vertex, target vertex, and label are duplicate edges.</p>   |
| createNotExists | No        | Boolean | <p>Whether to add source or target vertices that do not exist in the <b>edges</b> parameter before adding edges. The value can only be <b>false</b>.</p> <p><b>false</b>: Edges can be added regardless of whether the source or target vertex exists.</p>   |

**Table 5-385** edges parameter description

| Parameter  | Mandatory | Type   | Description              |
|------------|-----------|--------|--------------------------|
| source     | Yes       | String | Source vertex of an edge |
| target     | Yes       | String | Target vertex of an edge |
| label      | Yes       | String | Edge label               |
| properties | No        | Object | Value of each property   |

## Response Parameters

**Table 5-386** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Add edges in batches. The source vertex is **46**, the target vertices are **39** and **38**, and the edge label is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "label": "rate",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "label": "rate",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ],
  "parallelEdge": {
    "action": "override",
```

```
"ignoreLabel": false
},
"createNotExists": false
}
```

 **NOTE**

- **SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertices **666** and **777** are not in the original graph, create vertices **666** and **777**, retain the default value of each label, and add an edge.

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.3.4 Deleting Edges in Batches

#### Function

This API is used to delete edges in batches based on the source vertices, target vertices, and indexes of the edges.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/action?action\_id=batch-delete

**Table 5-387** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-388** Request body parameters

| Parameter   | Mandatory | Type    | Description  |
|-------------|-----------|---------|--|
| edges       | Yes       | Object  | Edge array to be deleted   |
| ignoreError | No        | Boolean | Whether to ignore errors, for example, the edge to delete does not exist. The default value is <b>false</b> , indicating that errors will not be ignored. Errors in JSON format cannot be ignored. |

**Table 5-389** edges parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| source    | Yes       | String | Source vertex of an edge   |
| target    | Yes       | String | Target vertex of an edge   |
| index     | No        | String | Edge index   |
| label     | No        | String | Label of an edge. If the <b>index</b> parameter is set, this parameter is ignored. If the <b>index</b> parameter is not set, an edge that meets the <b>source</b> , <b>target</b> , and <b>label</b> conditions is deleted. If the specified <b>label</b> value does not exist in the schema or the edge with the same <b>label</b> does not exist, no edge will be deleted. |

## Response Parameters

**Table 5-390** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Delete edges in batches. The source vertices of the edges are **39631050\_Landscape** and **27803870\_Landmark building**, and the target vertices of the edges are **27803870\_Landmark building** and **27661363\_Villa hot spring**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete
{
  "edges": [
    {
      "source": "39631050_Landscape",
      "target": "27803870_Landmark building"
    },
    {
      "index": "0",
      "source": "27803870_Landmark building",
      "target": "27661363_Villa hot spring"
    }
  ],
  "ignoreError": true
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)



```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter does not contain 'source'",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.3.5 Updating Edge Properties in Batches

#### Function

This API is used to update edge properties in batches.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/edges/properties/action?  
action\_id={actionId}

**Table 5-391** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| graph_name | Yes       | String | Graph name   |
| actionId   | Yes       | String | Operator. Possible values: <ul style="list-style-type: none"> <li>• <b>batch-update</b>: Update the value of a property.</li> <li>• <b>batch-add</b>: Add the value to a property. When the property's <b>cardinality</b> is <b>single</b>, the operation is the same as that of <b>batch-update</b>. When <b>cardinality</b> is <b>list</b> or <b>set</b>, the operator adds a value to a set.</li> <li>• <b>batch-del</b>: Delete a property value.</li> </ul> |

## Request Parameters

**Table 5-392** Request body parameters

| Parameter | Mandatory | Type | Description              |
|-----------|-----------|------|--------------------------|
| edges     | Yes       | JSON | Edge array to be updated |

**Table 5-393** edges parameter description

| Parameter  | Mandatory | Type   | Description              |
|------------|-----------|--------|--------------------------|
| source     | Yes       | String | Source vertex of an edge |
| target     | Yes       | String | Target vertex of an edge |
| label      | No        | String | Edge label               |
| properties | Yes       | Object | Value of each property   |

## Response Parameters

**Table 5-394** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update edge properties in batches. The source vertex of the edge is **46**, and the target vertices of the edge are **39** and **38**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=batch-update
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "index": "0",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ]
}
```

 NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

Http Status Code: 200

```
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

Http Status Code: 400

```
{
  "errorMessage": "edge source vertex [46] does not exist",
  "errorCode": "GES.8221"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.2.4 Metadata Operation APIs

### 5.2.4.1 Adding a Label

#### Function

This API is used to add a label.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/labels

**Table 5-395** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

- Request parameters (OBS scenario)

**Table 5-396** Request body parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| name       | Yes       | String | Name of a label<br>A label name can contain a maximum of 256 characters.<br>Only letters, digits, spaces, and special characters %, @, #, \$, :, ;, *, ., +, - are allowed.  |
| type       | No        | String | Label type, indicating that the label is used for vertices or edges. The options are as follows: <ul style="list-style-type: none"> <li><b>vertex</b>: indicates that the label is used for vertices.</li> <li><b>edge</b>: indicates that the label is used for edges.</li> <li><b>all</b>: indicates that the label is used for vertices and edges.</li> </ul> The default value is <b>all</b> . |
| properties | Yes       | Object | Properties you want to add to the label. For details about the parameters, see <a href="#">Table 5-397</a> .   |

**Table 5-397** properties parameter description

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| property  | No        | Object | Label properties. For details about the parameters, see <a href="#">Table 5-398</a> . |

**Table 5-398** property parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| name        | Yes       | String | Property name<br>1. A property name can contain a maximum of 256 characters.<br>2. A property name cannot contain <, >, &, ASCII 14,15 or 30.<br>3. The property under a label must be unique. |
| cardinality | Yes       | String | Composite type of a property. Currently, only <b>single</b> is supported.  |
| dataType    | Yes       | String | Data type of a property. For details, see the metadata types in <a href="#">Specification Description</a> .  |

## Response Parameters

**Table 5-399** Parameter description

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .   |

## Example Request

Add a label. The label name is **book**. The label has one property to add.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
{
  "name": "book",
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    }
  ]
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |

| Return Value              | Description            |
|---------------------------|------------------------|
| 404 Not Found             | No resources found.    |
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

### 5.2.4.2 Updating a Label

#### Function

In the current version, this API can only add properties to the end of existing labels, but cannot delete existing properties or update the property sequence.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema?label={labelName}

**Table 5-400** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| label_name | Yes       | String | Label name   |



## Request Parameters

**Table 5-401** Request body parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| type       | No        | String | Label type, indicating that the label is used for vertices or edges. The options are as follows: <ul style="list-style-type: none"> <li>• <b>vertex</b>: indicates that the label is used for vertices.</li> <li>• <b>edge</b>: indicates that the label is used for edges.</li> <li>• <b>all</b>: indicates that the label is used for vertices and edges.</li> </ul> The default value is <b>all</b> . |
| properties | Yes       | Object | Property array to be appended. <a href="#">Table 5-402</a> describes the parameters in an array.   |

**Table 5-402** properties parameter description

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| property  | No        | Object | Label properties. For details about the parameters, see <a href="#">Table 5-403</a> . |

**Table 5-403** property parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| name        | Yes       | String | Property name <ol style="list-style-type: none"> <li>1. A property name can contain a maximum of 256 characters.</li> <li>2. A property name cannot contain &lt;, &gt;, &amp;, ASCII 14,15 and 30.</li> <li>3. The property under a label must be unique.</li> </ol> |
| cardinality | Yes       | String | Composite type of a property. Currently, only <b>single</b> is supported.  |
| dataType    | Yes       | String | Data type of a property. For details, see the metadata types in <a href="#">Specification Description</a> .  |

## Response Parameters

**Table 5-404** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .  |

## Example Request

Update a label. The label name is **book**. The label has one property to update.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
{
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    }
  ]
}
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.4.3 Querying Labels

#### Function

This API is used to query a label.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema?label={labelName}

**Table 5-405** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

| Parameter  | Mandatory | Type   | Description |
|------------|-----------|--------|-------------|
| label_name | Yes       | String | Label name  |

## Response Parameters

**Table 5-406** Parameter description

| Parameter    | Type                  | Description  |
|--------------|-----------------------|--|
| data         | <b>data</b><br>Object | Query results. This parameter is left blank when the request fails.  |
| result       | String                | Request result. If the request is successful, the value is <b>success</b> . If the request fails, the value is <b>failed</b> .   |
| errorMessage | String                | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String                | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |

**Table 5-407** data parameter description

| Parameter  | Type   | Description  |
|------------|--------|--|
| properties | Object | Property array.  |
| type       | String | Label type, indicating that the label is used for vertices or edges. |

## Example Request

Query labels.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
```

 NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "type": "vertex",
    "properties": [
      {
        "name": "Rating",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "Datetime",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  "result": "success"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8204"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.2.4.4 Querying Graph Metadata Details

### Function

This API is used to query graph metadata details.

### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema

**Table 5-408** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

### Response Parameters

**Table 5-409** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data         | Object | Query results. This parameter is left blank when the request fails.   |

**Table 5-410** data parameter description

| Parameter | Type | Description   |
|-----------|------|---|
| schema    | List | Definition of each label and their associated property fields |

**Table 5-411** schema parameter description

| Parameter  | Type   | Description   |
|------------|--------|---|
| label      | String | label name  |
| properties | Object | Property array. For details, see <a href="#">properties parameter description</a> . |
| type       | String | Label type, indicating that the label is used for vertices or edges.                |

**Table 5-412** properties parameter description

| Parameter   | Type   | Description  |
|-------------|--------|--|
| name        | String | Property name<br><ol style="list-style-type: none"> <li>1. A property name can contain a maximum of 256 characters.</li> <li>2. A property name cannot contain &lt;, &gt;, &amp;, ASCII 14, 15, or 30.</li> <li>3. The property under a label must be unique.</li> </ol> |
| cardinality | String | Composite type of a property. Currently, only <b>single</b> is supported.  |
| dataType    | String | Data type of a property. For details, see the metadata types in <a href="#">Specification Description</a> .  |

## Example Request

Query metadata details of a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "schema": [
      {
        "label": "__DEFAULT__",
        "type": "all"
      }
    ]
  }
}
```

```

    "label": "friends",
    "type": "vertex"
  },
  {
    "label": "movie",
    "type": "vertex",
    "properties": [
      {
        "name": "ChineseTitle",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Year",
        "type": "int",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "user",
    "type": "vertex",
    "properties": [
      {
        "name": "Name",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Occupation",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Zip-code",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "rate",
    "type": "edge",
    "properties": [
      {
        "name": "Score",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "Datetime",
        "type": "date",
        "cardinality": "single"
      }
    ]
  }
]
}
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8003"
}

```



## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.4.5 Generating Data Assets

#### Function

This API is used to generate data assets.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/data-assets

**Table 5-413** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

## Response Parameters

**Table 5-414** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                            |

### Example Request

Generate data assets.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
{ }
```

### Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorCode": "GES.8301",
  "errorMessage": "Label index in vertices or edges is not found, please build the index first."
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.4.6 Obtaining Data Assets

#### Function

This API is used to obtain data assets.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/schema/data-assets

**Table 5-415** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

## Response Parameters

**Table 5-416** Response parameters

| Parameter          | Type   | Description  |
|--------------------|--------|--|
| generating         | Bool   | Whether data assets are being generated.   |
| progress           | String | Progress of generating data assets   |
| last_generate_time | String | Last time when a data asset is generated   |
| data_assets        | Object | Number of vertices and edges under different labels. For details, see <a href="#">Table 5-417</a> .  |
| errorCode          | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| errorMessage       | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |

**Table 5-417** data\_assets parameter description

| Parameter | Type   | Description                               |
|-----------|--------|---|
| vertex    | Object | Number of vertices under different labels |
| edge      | Object | Number of edges under different labels    |

### Example Request

Obtain data assets.

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
```

### Example Response

**Status code: 200**

There are several scenarios for example response for successful requests, including:

- A data asset has never been generated.

```
Http Status Code: 200
{ }
```

- A data asset has never been generated and is being generated.

```
Http Status Code: 200
{
  "progress": "10.05%",
  "generating": true
}
```

- A data asset has been generated and a latest one is being generated.

```
Http Status Code: 200
{
  "last_generate_time": "2022-1-3 12:34:12",
  "data_assets": {
    "vertex": {
      "label1": 3,
      "label2": 14
    },
    "edge": {
      "label3": 123,
      "label4": 435
    }
  },
  "progress": "10.05%",
  "generating": true
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorCode": "GES.8301",
  "errorMessage": "Generate data assets progress is not exist!"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes for Service Plane APIs](#).

**5.2.5 Index Operation APIs**

## 5.2.5.1 Creating an Index

### Function

This API is used to create indexes based on the specified information such as `indexName` and `IndexType`. Currently, composite indexes and full-text indexes are supported.

- Composite indexes include **GlobalCompositeVertexIndex**, **GlobalCompositeEdgeIndex**, **CompositeVertexIndex**, and **CompositeEdgeIndex**. A composite index is created for a fixed combination of properties. A local index can be created on a specified label. You do not need to specify a label for a global composite index. As long as a label contains a specified property, an index is automatically created on the label. Indexes can be used to accelerate queries.
- Full-text indexes (`FullTextIndex`) can implement functions such as full-text search and fuzzy search. If you search data immediately after an update, you may get the old data. You are advised to query the data 1 second after the update. For details about how to use full-text indexes, see [Querying Vertices That Meet Filter Criteria](#) and [Querying Edges That Meet Filter Criteria](#). You can also use full-text indexes in Cypher statements. For details, see [Performing Cypher Queries](#).

#### NOTE

- Currently, the full-text indexing feature is supported only in the ECS/BMS+MRS deployment mode.
- After an index is created, wait for 30 seconds for index synchronization. After the synchronization is complete, Cypher queries can be accelerated using the index.

### Indexes

| Feature           | Fuzzy Search | Speed | Flexibility                        |
|-------------------|--------------|-------|------------------------------------|
| Composite indexes | No           | Fast  | Fixed composite property keys only |

### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices

**Table 5-418** URI parameters

| Parameter               | Mandatory | Type   | Description  |
|-------------------------|-----------|--------|--|
| <code>project_id</code> | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| <code>graph_name</code> | Yes       | String | Graph name   |

## Request Parameters

**Table 5-419** Request body parameters

| Parameter     | Mandatory   | Type   | Description  |
|---------------|---|--------|--|
| indexName     | Yes   | String | Index name. Only letters, digits, hyphens (-), and underscores (_) are allowed. Other characters are not allowed. The index name can contain a maximum of 63 characters.   |
| indexType     | Yes   | String | Index type, which is case sensitive <ul style="list-style-type: none"> <li>• <b>GlobalCompositeVertexIndex</b> is a global composite vertex index.</li> <li>• <b>GlobalCompositeEdgeIndex</b> is a global composite edge index.</li> <li>• <b>FullTextIndex</b> is a full-text index.</li> </ul> |
| indexLabel    | No  | List   | Labels on which indexes are created. This parameter is available only when <b>indexType</b> is set to <b>CompositeVertexIndex</b> or <b>CompositeEdgeIndex</b> . This parameter is mandatory.  |
| indexProperty | No (If <b>hasLabel</b> is <b>false</b> or <b>null</b> , this parameter is mandatory.) | String | Index property list.<br>The property types that can be used to create indexes include integer, float, double, long, enum, char array, string, and date.  |

 **NOTE**

- If a property is of the string or char array type, the value must be no more than 40 bytes. The excess part will be deleted.
- Cypher queries can be accelerated with indexes whose **hasLabel** is **True**.
  - If **indexProperty** is left blank, the created index is a label index, which accelerates label filtering.
  - If **indexProperty** is specified, the created index is a property index, which accelerates property filtering.

## Response Parameters

**Table 5-420** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>                                 |
| errorCode    | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>                                    |
| jobId        | String  | ID of an asynchronous job<br><b>NOTE</b> <ul style="list-style-type: none"> <li>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a>.</li> </ul> |
| jobType      | Integer | Type of an asynchronous job   |
| result       | String  | If the execution is successful, the value of <b>result</b> is <b>success</b> .  |

## Example Request

Create a composite index. The index name is **ageIndex** and the index type is global vertex index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "ageIndex",
  "indexType": "GlobalCompositeVertexIndex",
  "hasLabel": true,
  "indexProperty": []
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
```



```
"jobType": 8
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8603"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.2.5.2 Deleting an Index**

**Function**

This API is used to delete an index based on the specified indexName.

**URI**

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices/{indexName}

**Table 5-421** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| indexName | Yes       | String | Index name  |

## Response Parameters

**Table 5-422** Parameter description

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobType      | Integer | Type of an asynchronous job   |

## Example Request

Delete an index by name. The index name is **ageIndex**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/ageIndex
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232",
  "jobType": 9
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8604 "
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.5.3 Querying Indexes

#### Function

This API is used to query all indexes created on a graph.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices

**Table 5-423** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Response Parameters

**Table 5-424** Response body parameters

| Parameter     | Type   | Description   |
|---------------|--------|---|
| errorMessage  | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode     | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| data          | Object | Index data you want to query  |
| result        | String | Query results. If the query is successful, <b>success</b> is displayed.   |
| indices       | List   | Indexes of the query results  |
| indexType     | String | Index types of the query results  |
| indexName     | String | Index names of the query results  |
| indexLabel    | List   | Labels of local indexes   |
| indexProperty | List   | Index properties of the query results   |

### Example Request

Query all indexes created on a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

### Example Response

**Status code: 200**

Example response (successful request)

Http Status Code: 200

```
{
  "data": {
    "indices": [
      {
        "indexType": "GlobalCompositeVertexIndex",
        "indexName": "ageIdx",

```

```

    "indexProperty": [
      "age"
    ],
    "hasLabel": true
  }
]
},
"result": "success"
}

```

**Status code: 400**

Example response (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8605"
}

```

### Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

### Error Codes

See [Error Codes](#).

### 5.2.5.4 Creating Indexes in Batches

#### Function

This API is used to create indexes in batches. By doing so, the number of data scans is reduced and the overall time required is shortened. The types of indexes that can be created are the same as those of the index creation API. For details, see [Creating an Index](#).

 **NOTE**

After an index is created, wait for 30 seconds for index synchronization. After the synchronization is complete, Cypher queries can be accelerated using the index.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/indices/action?action\_id=batch-build

**Table 5-425** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-426** Request body parameter

| Parameter | Mandatory | Type  | Description  |
|-----------|-----------|-------|--|
| indices   | Yes       | Array | Index array. The number of new indexes cannot exceed the maximum number of supported indexes minus the number of existing indexes. Currently, a maximum of 10 indexes can be created. For details about index parameters, see Table 2. |

**Table 5-427** indices parameter description

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| indexName | Yes       | String | Index name. Only letters, digits, hyphens (-), and underscores (_) are allowed. The index name can contain a maximum of 63 characters.   |
| indexType | Yes       | String | Index type, which is case sensitive <ul style="list-style-type: none"> <li>• <b>GlobalCompositeVertexIndex</b> is a global composite vertex index.</li> <li>• <b>GlobalCompositeEdgeIndex</b> is a global composite edge index.</li> <li>• <b>FullTextIndex</b> is a full-text index.</li> </ul> |

| Parameter     | Mandatory   | Type   | Description   |
|---------------|---|--------|---|
| indexProperty | No (If <b>hasLabel</b> is <b>false</b> or <b>null</b> , this parameter is mandatory.) | String | Index property list<br>Indexes can be created for the following property types: <b>integer</b> , <b>float</b> , <b>double</b> , <b>long</b> , <b>enum</b> , <b>char array</b> , <b>string</b> , and <b>date</b> . |

 NOTE

- If a property is of the string or char array type, the value must be no more than 40 bytes. The excess part will be deleted.
- Cypher queries can be accelerated with indexes whose **hasLabel** is **True**.
  - If **indexProperty** is left blank, the created index is a label index, which accelerates label filtering.
  - If **indexProperty** is specified, the created index is a property index, which accelerates property filtering.

## Response Parameters

**Table 5-428** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>                                |
| errorCode    | String  | System prompt code <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>                              |
| jobId        | String  | ID of an asynchronous job<br><b>NOTE</b> <ul style="list-style-type: none"> <li>• You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a>.</li> </ul> |
| jobType      | Integer | Type of an asynchronous job   |

| Parameter | Type   | Description   |
|-----------|--------|---|
| result    | String | If the execution is successful, the value is <b>success</b> . |

## Example Request

Create multiple composite indexes at a time. The index names are **vertexIndex**, **edgeIndex**, and **useridIndex**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/action?action_id=batch-build
{
  "indices": [
    {
      "indexName": "vertexIndex",
      "indexType": "GlobalCompositeVertexIndex",
      "hasLabel": true,
      "indexProperty": []
    },
    {
      "indexName": "edgeIndex",
      "indexType": "GlobalCompositeEdgeIndex",
      "hasLabel": true,
      "indexProperty": []
    },
    {
      "indexName": "useridIndex",
      "indexType": "GlobalCompositeEdgeIndex",
      "hasLabel": false,
      "indexProperty": [
        "userid"
      ]
    }
  ]
}
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 8
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8603"
}
```



## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.2.6 HyG Graph Management APIs

### 5.2.6.1 Creating a HyG Graph

#### Function

This API is used to create a HyG graph.

#### NOTE

Computing of graphs of the database edition relies on the HyG engine. Before executing an algorithm, you need to create a HyG graph and synchronize data from the graph database to the HyG engine.

#### URI

POST /ges/v1.0/{project\_id}/hyg/{graph\_name}

**Table 5-429** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-430** Request body parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| policy    | No        | String  | Graph partitioning policy. Currently, the oec policy is supported. The default policy is <b>oec</b> .  |
| inEdge    | No        | Boolean | Whether the graph contains incoming edges. The default value is <b>false</b> . If this parameter is set to <b>true</b> , the data synchronization performance will be affected.<br><br>For some algorithms, such as <code>shortest_path</code> , <code>sssp</code> , and <code>k_hop</code> , if the incoming edges are not contained, the algorithm performance may deteriorate or an error may be reported. For details, see the parameter description of the corresponding algorithm. |

## Response Parameters

**Table 5-431** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Execution results <ul style="list-style-type: none"> <li>If the execution is successful, the value is <b>success</b>.</li> <li>If the execution fails, the value is <b>failed</b>.</li> </ul>                        |

## Example Request

To create a HyG graph, set the partitioning policy to **oec** and configure the graph to contain incoming edges.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}
{
  "policy": "oec",
  "inEdge": true
}
```

### NOTE

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorCode": "GES.8011",
  "errorMessage": "graph : movie2 is not exist"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 5.2.6.2 Synchronizing HyG Graph Data

### Function

This API is used to synchronize graph database updates to the HyG computing engine.

### URI

POST /ges/v1.0/{project\_id}/hyg/{graph\_name}/sync

**Table 5-432** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

### Request Parameters

**Table 5-433** Request body parameters

| Parameter | Mandatory | Type | Description   |
|-----------|-----------|------|---|
| vertex    | No        | JSON | Vertex property list. If the list is empty, vertex properties will not be synchronized. For details about the parameters, see <a href="#">Table 5-434</a> .<br><br>During the initial data synchronization, this parameter will be applied. For subsequent synchronizations, this parameter will default to the value specified during the first synchronization. |
| edge      | No        | JSON | Edge property list. If the list is empty, edge properties will not be synchronized. For details about the parameters, see <a href="#">Table 5-434</a> .<br><br>During the initial data synchronization, this parameter will be applied. For subsequent synchronizations, this parameter will default to the value specified during the first synchronization.     |

**Table 5-434** vertex and edge parameters

| Parameter | Mandatory | Type             | Description   |
|-----------|-----------|------------------|---|
| label     | Yes       | String           | Label name  |
| property  | Yes       | Array of strings | Property name. The property must belong to the label. |

## Response Parameters

**Table 5-435** Parameter description

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                        |

 **NOTE**

Suspended edges in a graph database (edges that do not have either a source or a target node) are not synchronized to the HyG engine.

## Example Request

Synchronize the update information of the graph database to the HyG computing engine, with an empty vertex property list, an edge property list with the property name **Rating**, and a label name of **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/sync
{
  "vertex": [],
  "edge": [
    {
      "property": [
        "Rating"
      ]
    }
  ]
}
```

```

    ],
    "label": "rate"
  }
]
}

```

 **NOTE**

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```

Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}

```

### Status code: 400

Example response for a failed request

```

Http Status Code: 400
{
  "errorCode": "GES.8011",
  "errorMessage": "graph : movie2 is not exist"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.6.3 Querying General Information About a HyG Graph

#### Function

This API is used to query general information about a HyG graph, such as the number of vertices, number of edges, properties, and partitioning policies.

## URI

GET /ges/v1.0/{project\_id}/hyg/{graph\_name}/summary

**Table 5-436** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

None

## Response Parameters

**Table 5-437** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | If the query is successful, the <b>data</b> field is returned. For details about the parameters, see <a href="#">Table 5-438</a> .   |
| status       | String | Job status returned for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . This parameter is left blank when the query fails.   |
| result       | String | Execution results <ul style="list-style-type: none"> <li>If the execution is successful, the value is <b>success</b>.</li> <li>If the execution fails, the value is <b>failed</b>.</li> </ul>                        |

**Table 5-438** data parameter description

| Parameter  | Type    | Description                              |
|------------|---------|--|
| vertex     | JSON    | Vertex labels and properties             |
| edge       | JSON    | Edge labels and properties               |
| policy     | String  | Partitioning policy                      |
| inEdge     | Boolean | Whether incoming edges are contained     |
| idIndex    | Boolean | Whether the vertex ID index is contained |
| updateTime | String  | Time when a graph is updated             |
| vertexNum  | Integer | Number of vertices                       |
| edgeNum    | Integer | Number of edges                          |

## Example Request

Query general information about a HyG graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/summary
```

### NOTE

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "inEdge": true,
    "idIndex": true,
    "policy": "oec",
    "updateTime": "2023-08-03 15:13:16",
    "vertex": [],
    "edge": [
      {
        "label": "rate",
        "property": [
          "Rating"
        ]
      }
    ],
    "vertexNum": 150,
    "edgeNum": 1659
  },
  "result": "success"
}
```

**Status code: 400**

Example response for a failed request



```

Http Status Code: 400
{
  "status": "complete",
  "errorCode": "GES.8011",
  "errorMessage": "graph [movie2] is not found",
  "result": "failed"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.6.4 Deleting a HyG Graph

#### Function

This API is used to delete a HyG graph.

#### URI

DELETE /ges/v1.0/{project\_id}/hyg/{graph\_name}

**Table 5-439** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

## Response Parameters

**Table 5-440** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>                          |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul>                        |
| jobId        | String | ID of an asynchronous job<br><b>NOTE</b> <ul style="list-style-type: none"> <li>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a>.</li> </ul> |
| status       | String | Job status returned for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . This parameter is left blank when the query fails.  |
| result       | String | Execution results <ul style="list-style-type: none"> <li>If the execution is successful, the value is <b>success</b>.</li> <li>If the execution fails, the value is <b>failed</b>.</li> </ul>   |

## Example Request

Delete a HyG graph.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}
```

### NOTE

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
```

```
"jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232"
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "status": "complete",
  "errorCode": "GES.8011",
  "errorMessage": "graph [movie2] is not found",
  "result": "failed"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes for Service Plane APIs](#).

**5.2.6.5 Listing HyG Graphs**

**Function**

This API is used to list HyG graphs.

**URI**

GET /ges/v1.0/{project\_id}/hyg

**Table 5-441** URI parameter

| Parameter  | Mandator<br>y | Type   | Description  |
|------------|---------------|--------|--|
| project_id | Yes           | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

None

## Response Parameters

**Table 5-442** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| data         | Object | If the query is successful, the <b>data</b> field is returned. For details about the parameters, see <a href="#">Table 5-443</a> .   |
| result       | String | Execution results <ul style="list-style-type: none"> <li>If the execution is successful, the value is <b>success</b>.</li> <li>If the execution fails, the value is <b>failed</b>.</li> </ul>                        |

**Table 5-443** data parameter description

| Parameter  | Type    | Description                              |
|------------|---------|--|
| name       | String  | Graph name                               |
| vertex     | JSON    | Vertex labels and properties             |
| edge       | JSON    | Edge labels and properties               |
| policy     | String  | Partitioning policy                      |
| inEdge     | Boolean | Whether incoming edges are contained     |
| idIndex    | Boolean | Whether the vertex ID index is contained |
| updateTime | String  | Time when a graph is updated             |
| vertexNum  | Integer | Number of vertices                       |

| Parameter | Type    | Description     |
|-----------|---------|-----------------|
| edgeNum   | Integer | Number of edges |

## Example Request

List HyG graphs.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/hyg
```

### NOTE

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": [
    {
      "name": "test",
      "inEdge": true,
      "idIndex": true,
      "policy": "oec",
      "vertexNum": 0,
      "edgeNum": 0
    },
    {
      "name": "movie",
      "inEdge": true,
      "idIndex": false,
      "policy": "oec",
      "updateTime": "2023-11-15 18:11:00",
      "vertex": [],
      "edge": [],
      "vertexNum": 146,
      "edgeNum": 1659
    }
  ],
  "result": "success"
}
```

### Status code: 400

Example response for a failed request

```
HttpStatusCode: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.6.6 Importing an HyG Graph

#### Function

This API is used to import HyG graph data.

#### URI

POST /ges/v1.0/{project\_id}/hyg/{graph\_name}/import-graph

**Table 5-444** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-445** Request body parameters

| Parameter   | Mandatory | Type   | Description   |
|-------------|-----------|--------|---|
| edgesetPath | Yes       | String | Edge file directory or edge file name. Currently, only CSV files can be imported. |

| Parameter     | Mandatory | Type      | Description   |
|---------------|-----------|-----------|---|
| vertexsetPath | Yes       | String    | Vertex file directory or vertex file name. Currently, only CSV files can be imported.   |
| schemaPath    | Yes       | String    | OBS path of the metadata file of the new data   |
| logDir        | No        | String    | Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.  |
| delimiter     | No        | Character | Field separator in a CSV file. The default value is comma (,).  |
| trimQuote     | No        | Character | Field quote character in a CSV file. The default value is double quotation marks ("). It is used to enclose a field if the field contains separators or line breaks.  |
| obsParameters | Yes       | Object    | OBS parameters. For details about the parameters, see <a href="#">Table 5-446</a> .   |
| vertex        | No        | Object    | List of vertex properties. The specified properties must belong to the schema file. If the list is empty, vertex properties will not be imported. For details about the parameters, see <a href="#">Table 5-447</a> . |
| edge          | No        | Object    | List of edge properties. The specified properties must belong to the schema file. If the list is empty, edge properties will not be imported. For details about the parameters, see <a href="#">Table 5-447</a> .     |

**Table 5-446** obsParameters parameter description

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | String | User AK     |
| secretKey | Yes       | String | User SK     |

**Table 5-447** vertex and edge parameters

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| label     | Yes       | String | Label name  |

|          |     |                  |  |
|----------|-----|------------------|--|
| property | Yes | Array of strings | Property name. The property must belong to the label. Supported property types include string, bool, int, long, double, and float. |
|----------|-----|------------------|--|

## Response Parameters

**Table 5-448** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                       |

## Example Request

Import a graph. The edge file directory is **testbucket/demo\_movie/edges/** and the edge data set format is CSV; the vertex file directory is **testbucket/demo\_movie/vertices/** and the vertex data set format is CSV; the OBS path of the metadata file of the new data is **testbucket/demo\_movie/schema.xml** and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "schemaPath": "testbucket/demo_movie/schema.xml",
  "logDir": "testbucket/importlogdir",
  "delimiter": ",",
  "trimQuote": "\"",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  },
  "vertex": [
    {
      "property": [
```



```

        "title",
        "movieid"
      ],
      "label": "movie"
    }
  ],
  "edge": [
    {
      "property": [
        "Rating",
        "Datetime"
      ],
      "label": "rate"
    }
  ]
}

```

 **NOTE**

**SERVER\_URL:** URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```

Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}

```

### Status code: 400

Example response for a failed request

```

Http Status Code: 400
{
  "errorCode": "GES.8013",
  "errorMessage": "graph [movie2] is not found"
}

```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 5.2.7 HyG Algorithm APIs

### 5.2.7.1 Running Algorithms

#### Function

This API is used to run specified algorithms based on entered parameters.

#### URL

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/hyg/algorithm

**Table 5-449** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/hyg/algorithm
{
  "algorithmName":"pagerank",
  "data_id": "0-91494cd9-e7da-4fb9-ba56-b2301967688d0000000019090",
  "parameters":{
    "alpha":0.85,
    "convergence":0.00001,
    "max_iterations":1000,
    "directed":true
  }
}
```

#### Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

## Response Parameters

**Table 5-450** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>  |
| jobId        | String | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "status": "complete",
  "errorCode": "GES.8011",
  "errorMessage": "graph : movie2 is not exist",
  "result": "failed"
}
```

## Status Codes

| Return Value     | Description           |
|------------------|-----------------------|
| 400 Bad Request  | Request error.        |
| 401 Unauthorized | Authorization failed. |

| Return Value              | Description               |
|---------------------------|---------------------------|
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## 5.2.7.2 Algorithm API Parameter Reference

### 5.2.7.2.1 Common Algorithm Parameters

#### Request Parameters

**Table 5-451** Request body parameters

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| algorithmName | Yes       | String | Algorithm name<br>Available values are as follows (algorithm names you can call): <ul style="list-style-type: none"> <li>• pagerank</li> <li>• kcore</li> <li>• k_hop</li> <li>• shortest_path</li> <li>• sssp</li> <li>• connected_component</li> </ul> |
| data_id       | Yes       | String | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management."  |
| parameters    | Yes       | Object | Algorithm parameters. For details, see the parameter description of each algorithm.  |

#### Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the **job\_id** returned by the algorithm to obtain the algorithm execution result.

**Status code: 200**

Example response for a successful request

```
{
  "data": {
    "outputs": {
      $response_data //Result of each algorithm. The results vary with the algorithm.
      "runtime": 1.365867,
      "data_return_size": 3,
      "data_offset": 0,
      "data_total_size": 100
    }
  },
  "status": "complete"
}
```

 **NOTE**

**response\_data** indicates the result of each algorithm. The results vary with algorithms.

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Running algorithm [XXXX] error: YYYYYYYYYY!",
  "errorCode": "GES.8301"
}
```

### 5.2.7.2.2 Shortest Path

#### Function

This API is used to execute the Shortest Path algorithm based on entered parameters.

The Shortest Path algorithm is used to find the shortest path between two nodes in a graph.

#### URI

POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm

**Table 5-452** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graphName | Yes       | String | Graph name   |

## Request Parameters

**Table 5-453** Request body parameters

| Parameter     | Mandatory | Type                              | Description   |
|---------------|-----------|-----------------------------------|---|
| algorithmName | Yes       | String                            | Algorithm name  |
| data_id       | Yes       | String                            | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes       | <a href="#">parameters</a> Object | Algorithm parameters  |

**Table 5-454** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| source    | Yes       | String  | Source vertex ID of a path   |
| target    | Yes       | String  | Target vertex ID of a path, which is not the same as the source vertex ID  |
| directed  | No        | Boolean | Whether to consider the edge direction. The value can be <b>true</b> or <b>false</b> .<br><b>NOTE</b> <ul style="list-style-type: none"> <li><b>false</b>: The current version does not support this function on weighted graphs.</li> <li>When the dataset does not contain incoming edges, if this parameter is set to <b>true</b> and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to <b>false</b>, an error is reported.</li> </ul> |
| weight    | No        | String  | Weight of an edge. The value can be empty or a string. <ul style="list-style-type: none"> <li>Empty: The default weight and distance of edges are <b>1</b>.</li> <li>String: The property of the edge is the weight. If the edge does not have a property, the weight is <b>1</b> by default.</li> </ul>   |

## Response Parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.  |
| errorCode    | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType      | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

```
POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName":"shortest_path",
  "data_id":"0-91494cd9-e7da-4fb9-ba56-b2301967688d000000019090",
  "parameters":{
    "source":"1",
    "target":"5",
    "directed":true,
    "weight":"","
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "graph [demo] is not found",
"errorCode": "GES.8402"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.7.2.3 SSSP

## Function

This API is used to execute the SSSP algorithm based on entered parameters.

This algorithm provides the shortest path from a given node (source node) to all other nodes.

## URI

```
POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
```

**Table 5-455** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graphName | Yes       | String | Graph name   |



## Request Parameters

**Table 5-456** Request body parameters

| Parameter     | Mandatory | Type                              | Description   |
|---------------|-----------|-----------------------------------|---|
| algorithmName | Yes       | String                            | Algorithm name  |
| data_id       | Yes       | String                            | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes       | <a href="#">parameters</a> Object | Algorithm parameters  |

**Table 5-457** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| source    | Yes       | String  | Source vertex ID of a path   |
| directed  | No        | Boolean | Whether to consider the edge direction. The value can be <b>true</b> or <b>false</b> .<br><b>NOTE</b> <ul style="list-style-type: none"> <li>• <b>false</b>: The current version does not support this function on weighted graphs.</li> <li>• When the dataset does not contain incoming edges, if this parameter is set to <b>true</b> and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to <b>false</b>, an error is reported.</li> </ul> |
| weight    | No        | String  | Weight of an edge. The value can be empty or a string. <ul style="list-style-type: none"> <li>• Empty: The default weight and distance of edges are <b>1</b>.</li> <li>• String: The property of the edge is the weight. If the edge does not have a property, the weight is <b>1</b> by default.</li> </ul>   |

## Response Parameters

**Table 5-458** Response body parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.  |
| errorCode    | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType      | Integer | Job type. This parameter is left blank if the request fails.  |

### Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName": "sssp ",
  "data_id": "0-91494cd9-e7da-4fb9-ba56-b2301967688d000000019090",
  "parameters": {
    "source": "1",
    "directed": true,
    "weight": ""
  }
}
```

 **NOTE**

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

### Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "graph [demo] is not found",
"errorCode": "GES.8402"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.7.2.4 K-Hop

## Function

This API is used to run the K-Hop algorithm based on entered parameters.

This algorithm is used to search all nodes in the k layer that are associated with the source node through Breadth-First Search (BFS). The found sub-graph is the source node's ego-net. This algorithm returns the number of nodes in the ego-net.

## URI

```
POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
```

**Table 5-459** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graphName | Yes       | String | Graph name   |

## Request Parameters

**Table 5-460** Request body parameters

| Parameter     | Mandatory | Type                              | Description   |
|---------------|-----------|-----------------------------------|---|
| algorithmName | Yes       | String                            | Algorithm name  |
| data_id       | Yes       | String                            | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes       | <a href="#">parameters</a> Object | Algorithm parameters  |

**Table 5-461** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| k         | Yes       | Integer | Number of hops. The value ranges from 1 to 100.  |
| source    | Yes       | String  | ID of the start node   |
| mode      | No        | String  | Search direction. Possible values are: <ul style="list-style-type: none"> <li>• <b>OUT</b>: Hop from the outgoing edges (default value)</li> <li>• <b>IN</b>: Hop from the incoming edges</li> <li>• <b>ALL</b>: Hop along both incoming and outgoing edges</li> </ul> <p><b>NOTE</b><br/>When the dataset does not contain incoming edges, if this parameter is set to <b>OUT</b> and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to <b>IN</b> or <b>ALL</b>, an error is reported.</p> |
| switch    | No        | Integer | When the number of activated edges reaches the total number of edges/switch value, the pull/push mode is switched. The value ranges from 1 to 2000, and the default value is <b>40</b> .   |

## Response Parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.  |
| errorCode    | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType      | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName":"k_hop",
  "data_id":"0-91494cd9-e7da-4fb9-ba56-b2301967688d000000019090",
  "parameters":{
    "k":3,
    "source":66,
    "mode":"ALL"
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage":"graph [demo] is not found",
  "errorCode":"GES.8402"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.7.2.5 PageRank

## Function

This API is used to run the PageRank algorithm based on entered parameters.

PageRank, also known as web page ranking, is a hyperlink analysis algorithm used to rank web pages (nodes) based on their search engine results. PageRank is a way of measuring the relevance and importance of web pages (nodes).

- If a web page is linked to many other web pages, the web page is important. That is, the PageRank value of the web page is relatively high.
- If a web page with a high PageRank value is linked to another web page, the PageRank value of the linked web page increases accordingly.

## URI

POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm

**Table 5-462** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graphName | Yes       | String | Graph name   |

## Request Parameters

**Table 5-463** Request body parameters

| Parameter     | Mandatory | Type                              | Description   |
|---------------|-----------|-----------------------------------|---|
| algorithmName | Yes       | String                            | Algorithm name  |
| data_id       | Yes       | String                            | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes       | <a href="#">parameters</a> Object | Algorithm parameters  |

**Table 5-464** parameters

| Parameter      | Mandatory | Type   | Description  |
|----------------|-----------|--------|--|
| alpha          | No        | Double | Weight coefficient (also called damping coefficient) The value range is between 0 and 1, excluding 0 and 1. The default value is <b>0.85</b> . |
| convergence    | No        | Double | Convergence. The value range is between 0 and 1, excluding 0 and 1. The default value is <b>0.00001</b> .                                      |
| max_iterations | No        | Int    | Maximum iterations. The value ranges from 1 to 2000. The default value is <b>1000</b> .  |

## Response Parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |

|         |         |   |
|---------|---------|---|
| jobId   | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName": "pagerank",
  "data_id": "0-91494cd9-e7da-4fb9-ba56-b2301967688d000000019090",
  "parameters": {
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}
```

## Status Codes

| Return Value     | Description           |
|------------------|-----------------------|
| 400 Bad Request  | Request error.        |
| 401 Unauthorized | Authorization failed. |



| Return Value              | Description               |
|---------------------------|---------------------------|
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.7.2.6 Connected Component

## Function

This API is used to run the Connected Component algorithm based on entered parameters.

A connected component represents a sub-graph, in which all nodes are connected to each other. Path directions are involved only in strongly connected components and are not considered in weakly connected components. This algorithm generates weakly connected components.

## URI

POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm

**Table 5-465** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graphName | Yes       | String | Graph name   |

## Request Parameters

**Table 5-466** Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|

|               |     |        |   |
|---------------|-----|--------|---|
| algorithmName | Yes | String | Algorithm name  |
| data_id       | Yes | String | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes | Object | Algorithm parameters  |

## Response Parameters

| Parameter    | Type    | Description   |
|--------------|---------|---|
| errorMessage | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.  |
| errorCode    | String  | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType      | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName":"connected_component",
  "data_id": "0-91494cd9-e7da-4fb9-ba56-b2301967688d0000000019090",
  "parameters":{
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
```

```
"jobType": 1
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage":"graph [demo] is not found",
  "errorCode":"GES.8402"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.2.7.2.7 K-core**

**Function**

This API is used to run the K-core algorithm based on entered parameters.

K-core is a classic graph algorithm used to calculate the number of cores of each node. The calculation result is one of the most commonly used reference values for determining the importance of a node so that the propagation capability of the node can be better understood.

**URI**

POST /ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm

**Table 5-467** URI parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| projectId | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| graphName | Yes       | String | Graph name  |

## Request Parameters

**Table 5-468** Request body parameters

| Parameter     | Mandatory | Type                        | Description   |
|---------------|-----------|-----------------------------|---|
| algorithmName | Yes       | String                      | Algorithm name  |
| data_id       | Yes       | String                      | Dataset ID. For details about how to obtain the dataset ID, see section 5.4.0 "HyG Dataset Management." |
| parameters    | Yes       | <b>parameters</b><br>Object | Algorithm parameters  |

**Table 5-469** parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| k         | Yes       | Integer | Number of cores. The algorithm returns vertices whose number of cores is greater than or equal to k. The value must be greater than or equal to <b>0</b> . |

## Response Parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |

|         |         |   |
|---------|---------|---|
| jobId   | String  | ID of the algorithm execution job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Querying Job Status on the Service Plane</a> . |
| jobType | Integer | Job type. This parameter is left blank if the request fails.  |

## Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/hyg/algorithm
{
  "algorithmName":"kcore",
  "data_id":"0-91494cd9-e7da-4fb9-ba56-b2301967688d000000019090",
  "parameters":{
    "k":10
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage":"graph [demo] is not found",
  "errorCode":"GES.8402"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |

| Return Value              | Description            |
|---------------------------|------------------------|
| 404 Not Found             | No resources found.    |
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

### 5.2.7.3 Algorithm Results in CSV Format

**Table 5-470** Algorithm results in CSV format

| Algorithm                | Supported On    | header   | content   | e.g.  |
|--------------------------|-----------------|--|---|---|
| all_pairs_shortest_paths | Local host, OBS | # runtime:<br>{runtime}<br># paths_number:<br>{paths_number}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># batch_paths: | Each line contains multiple paths of a pair. The format is as follows:<br>{sourceID},<br>{targetID}, "[[{sourceID},<br>{v1},...,<br>{targetID}],...]" | # runtime:<br>4.411<br>#<br>paths_number:<br>20<br>#<br>data_total_size:<br>25<br>#<br>data_return_size:<br>25<br># data_offset: 0<br># batch_paths:<br>"121","66","[[<br>121","25","66"]<br>]" |

| Algorithm                         | Supported On    | header  | content   | e.g.   |
|-----------------------------------|-----------------|---|---|--|
| all_shortest_paths                | Local host, OBS | <pre># runtime: {runtime} # source: {source} # target: {target} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # paths:</pre> | Each line is a path. The format is as follows: <pre>{sourceID}, {vertexID1},..., {targetID}</pre> | <pre># runtime: 0.207 # source: 121 # target: 66 # paths_number: 2 # data_total_size: 2 # data_return_size: 2 # data_offset: 0 # paths: 121,7,66 121,25,66</pre>                               |
| all_shortest_paths_of_vertex_sets | Local host, OBS | <pre># runtime: {runtime} # source: {source} # target: {target} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # paths:</pre> | Each line is a path. The format is as follows: <pre>{sourceID}, {vertexID1},..., {targetID}</pre> | <pre># runtime: 2.772 # sources: 48,129,34,36 # targets: 46,66,101 # paths_number: 15 # data_total_size: 15 # data_return_size: 15 # data_offset: 0 # paths: 36,72,101 36,59,46 36,73,46</pre> |

| Algorithm   | Supported On    | header   | content                      | e.g.  |
|-------------|-----------------|--|------------------------------|---|
| betweenness | Local host, OBS | # runtime:<br>{runtime}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># betweenness:  | {vertexID},<br>{betweenness} | # runtime:<br>1.593<br>#<br>data_total_size:<br>32<br>#<br>data_return_size:<br>32<br># data_offset: 0<br># betweenness:<br>79,20.69722222<br>222223<br>80,12.29058441<br>5584414<br>81,1.5                         |
| bigclam     | Local host, OBS | # runtime:<br>{runtime}<br>#<br>community_num:<br>{community_num}<br># log_likelihood:<br>{log_likelihood}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># communities: | {vertexID},<br>{community}   | # runtime:<br>2.754<br>#<br>community_num: 1<br>#<br>log_likelihood:<br>-5593.4549824<br>494925<br>#<br>data_total_size:<br>32<br>#<br>data_return_size:<br>32<br># data_offset: 0<br># communities:<br>6,0<br>13,0 |



| Algorithm | Supported On    | header   | content                    | e.g.  |
|-----------|-----------------|--|----------------------------|---|
| cesna     | Local host, OBS | # runtime:<br>{runtime}<br>#<br>community_num:<br>{community_num}<br># log_likelihood:<br>{log_likelihood}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># communities: | {vertexID},<br>{community} | # runtime:<br>40114.213<br>#<br>community_num<br>#<br>log_likelihood<br>#<br>data_total_size:<br>1344<br>#<br>data_return_size:<br>1344<br># data_offset: 0<br># communities:<br>3850,3<br>3858,3<br>3866,3 |
| closeness | Local host, OBS | # runtime:<br>{runtime}<br># source:<br>{source}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># closeness:   | {closeness}                | # runtime:<br>0.394<br># source: 12<br>#<br>data_total_size:<br>1<br>#<br>data_return_size:<br>1<br># data_offset: 0<br># closeness:<br>0.50877192982<br>45614  |

| Algorithm                              | Supported On    | header  | content                           | e.g.   |
|--|-----------------|---|-----------------------------------|--|
| cluster_coefficient (statistic = true) | Local host, OBS | # runtime: {runtime}<br># cluster_coefficient: {cluster_coefficient}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># vertex_cluster_coefficient: | {vertexID}, {cluster_coefficient} | # runtime: 0.661<br># cluster_coefficient: 0.13517429595852912<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># vertex_cluster_coefficient: |
| common_neighbors_of_vertex_sets        | Local host, OBS | # runtime: {runtime}<br># common_neighbors: {common_neighbors}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># vertices:                         | {vertexID}                        | # runtime: 0.42<br># common_neighbors: 26<br># data_total_size: 26<br># data_return_size: 26<br># data_offset: 0<br># vertices: 103<br>138<br>98                       |

| Algorithm           | Supported On    | header   | content  | e.g.  |
|---------------------|-----------------|--|--|---|
| connected_component | Local host, OBS | # runtime:<br>{runtime}<br>#<br>community_num:<br>{community_num}<br>#<br>Max_WCC_size:<br>{Max_WCC_size}<br># Max_WCC_id:<br>{Max_WCC_id}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>#<br>data_offset:<br>{data_offset}<br># community: | {vertexID},<br>{community}                       | # runtime:<br>0.263<br>#<br>community_num: 1<br>#<br>Max_WCC_size<br># Max_WCC_id<br>#<br>data_total_size:<br>32<br>#<br>data_return_size: 32<br># data_offset: 0<br># community:<br>2,0<br>6,0<br>13,0                             |
| edge_betweenness    | Local host, OBS | # runtime:<br>{runtime}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>#<br>data_offset:<br>{data_offset}<br>#<br>edge_betweenness:  | {sourceID},<br>{targetID},<br>{edge_betweenness} | # runtime:<br>153.006<br>#<br>data_total_size:<br>311<br>#<br>data_return_size:<br>311<br># data_offset: 0<br>#<br>edge_betweenness:<br>51,20,1.333333<br>3333333333<br>51,33,7.192099<br>567099566<br>51,10,3.476190<br>4761904763 |

| Algorithm | Supported On    | header   | content                 | e.g.   |
|-----------|-----------------|--|-------------------------|--|
| infomap   | Local host, OBS | # runtime: {runtime}<br># min_code_length: {min_code_length}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># community: | {vertexID}, {community} | # runtime: 98.158<br># min_code_length: 6.2680095519443135<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># community: 2,200000000556,2000000005013,20000000014 |
| k_hop     | Local host, OBS | # runtime: {runtime}<br># source: {source}<br># k: {k}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># vertices:        | {vertexID}              | # runtime: 0.442<br># source: 76<br># k: 6<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># vertices: 2613  |

| Algorithm         | Supported On    | header  | content                 | e.g.   |
|-------------------|-----------------|---|-------------------------|--|
| kcore             | Local host, OBS | # runtime: {runtime}<br># kmax: {kmax}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># coreness: | {vertexID}, {coreness}  | # runtime: 10.882<br># kmax: 15<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># coreness: 2,14<br>6,15<br>13,15              |
| label_propagation | Local host, OBS | # runtime: {runtime}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># community:                  | {vertexID}, {community} | # runtime: 2.624<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># community: 2,10000000024<br>6,10000000024<br>13,10000000024 |

| Algorithm       | Supported On    | header   | content                 | e.g.   |
|-----------------|-----------------|--|-------------------------|--|
| link_prediction | Local host, OBS | # runtime: {runtime}<br># source: {source}<br># target: {target}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># link_prediction: | {link_prediction}       | # runtime: 0<br># source: 123<br># target: 43<br># data_total_size: 1<br># data_return_size: 1<br># data_offset: 0<br># link_prediction: 0.07017543859649122                           |
| louvain         | Local host, OBS | # runtime: {runtime}<br># modularity: {modularity}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># community:                     | {vertexID}, {community} | # runtime: 45.835<br># modularity: 0.16375671670152867<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># community: 2,200000000626,2000000005013,20000000050 |

| Algorithm      | Supported On    | header   | content   | e.g.   |
|----------------|-----------------|--|---|--|
| n_paths        | Local host, OBS | # runtime: {runtime}<br># source: {source}<br># target: {target}<br># paths_number: {paths_number}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># paths: | Each line is a path. The format is as follows:<br>{sourceID}, {vertexID1},..., {targetID} | # runtime: 8.025<br># source: 123<br># target: 87<br># paths_number: 100<br># data_total_size: 100<br># data_return_size: 100<br># data_offset: 0<br># paths:<br>123,21,87<br>123,13,87<br>123,32,87 |
| od_betweenness | Local host, OBS | # runtime: {runtime}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># edge_betweenness:  | {sourceID}, {targetID}, {edge_betweenness}  | # runtime: 1.391<br># data_total_size: 311<br># data_return_size: 311<br># data_offset: 0<br># edge_betweenness:<br>51,20,0<br>51,33,0<br>51,10,0  |

| Algorithm    | Supported On    | header   | content                       | e.g.  |
|--------------|-----------------|--|-------------------------------|---|
| pagerank     | Local host, OBS | # runtime:<br>{runtime}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># pagerank:     | {vertexID},<br>{pagerank}     | # runtime:<br>4.044<br>#<br>data_total_size:<br>32<br>#<br>data_return_size:<br>32<br># data_offset: 0<br># pagerank:<br>2,0.0078889040<br>51903298<br>6,0.0132158636<br>92849642<br>13,0.018605301<br>99450448 |
| personalrank | Local host, OBS | # runtime:<br>{runtime}<br>#<br>data_total_size:<br>{data_total_size}<br>#<br>data_return_size:<br>{data_return_size}<br>}<br># data_offset:<br>{data_offset}<br># personalrank: | {vertexID},<br>{personalrank} | # runtime:<br>2.326<br># source: 46<br>#<br>data_total_size:<br>49<br>#<br>data_return_size:<br>49<br># data_offset: 0<br># personalrank:<br>0,0.0021350905<br>350732297<br>1,0.0045911514<br>06893241          |



| Algorithm                    | Supported On    | header  | content   | e.g.  |
|------------------------------|-----------------|---|---|---|
| shortest_path                | Local host, OBS | # runtime: {runtime}<br># source: {source}<br># target: {target}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># path: | Each line is a path. The format is as follows:<br>{sourceID}, {vertexID1},..., {targetID} | # runtime: 0.308<br># source: 123<br># target: 5<br># data_total_size: 1<br># data_return_size: 1<br># data_offset: 0<br># path: 123,10,137,5 |
| shortest_path_of_vertex_sets | Local host, OBS | # runtime: {runtime}<br># source: {source}<br># target: {target}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># path: | Each line is a path. The format is as follows:<br>{sourceID}, {vertexID1},..., {targetID} | # runtime: 1.832<br># source: 24<br># target: 121<br># data_total_size: 1<br># data_return_size: 1<br># data_offset: 0<br># path: 24,121      |

| Algorithm                       | Supported On    | header   | content  | e.g.   |
|---------------------------------|-----------------|--|--|--|
| single_vertex_circles_detection | Local host, OBS | <pre># runtime: {runtime} # source: {source} # min_circle_length: {min_circle_length} # max_circle_length: {max_circle_length} # limit_circle_number: {limit_circle_number} # circle_number: {circle_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # circles:</pre> | <p>Each line is a path. The format is as follows:</p> <pre>{sourceID}, {vertexID1},..., {sourceID}</pre> | <pre># runtime: 37.46 # source: 122 # target: # min_circle_length: 3 # max_circle_length: 10 # limit_circle_number: 100 # circle_number: 100 # data_total_size: 100 # data_return_size: 100 # data_offset: 0 # circles: 122,82,79,76,65,122 122,125,135,77,65,122 122,82,114,96,65,122</pre> |

| Algorithm | Supported On    | header  | content                   | e.g.  |
|-----------|-----------------|---|---------------------------|---|
| sssp      | Local host, OBS | <pre># runtime: {runtime}  # source: {source}  # data_total_size: {data_total_size}  # data_return_size: {data_return_size }  # data_offset: {data_offset}  # distance:</pre> | {vertexID},<br>{distance} | <pre># runtime: 0.452  # source: 32  # data_total_size: 48  # data_return_size: 48  # data_offset: 0  # distance: 0,2 5,2 7,2</pre> |

| Algorithm         | Supported On    | header  | content  | e.g.  |
|-------------------|-----------------|---|--|---|
| subgraph_matching | Local host, OBS | <pre># runtime: {runtime} # pattern_graph: {pattern_graph} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # subgraphs:</pre> | <p>Each line is a matched subgraph. The format is as follows:</p> <pre>{vertexID1}, {vertexID2},..., {vertexIDn}</pre> | <pre>----- statistics = true----- # runtime: 1.376 # pattern_graph: 2,3,1 # data_total_size: 1 # data_return_size: 1 # data_offset: 0 # subgraph_number: 1556 ----- statistics = false----- # runtime: 0.956 # pattern_graph: 2,3,1 # subgraph_number: 0 # data_total_size: 100 # data_return_size: 100 # data_offset: 0 # subgraphs: 0,51,126 0,51,131 0,126,113</pre> |

| Algorithm                         | Supported On    | header  | content                        | e.g.  |
|-----------------------------------|-----------------|---|--------------------------------|---|
| topic_rank                        | Local host, OBS | # runtime: {runtime}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># topicrank:  | {vertexID}, {topicrank}        | # runtime: 1.11<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># topicrank: 2,0.00663068274092574<br>6,0.007278130208954746<br>13,0.007869137668788257 |
| triangle_count (statistic = true) | Local host, OBS | # runtime: {runtime}<br># triangle_count: {triangle_count}<br># data_total_size: {data_total_size}<br># data_return_size: {data_return_size}<br># data_offset: {data_offset}<br># vertex_triangles: | {vertexID}, {vertex_triangles} | # runtime: 0.491<br># triangle_count: 1653<br># data_total_size: 32<br># data_return_size: 32<br># data_offset: 0<br># vertex_triangles :   |

 **NOTE**

Example response for a failed algorithm request:

```
Http Status Code: 400
{
  "errorMessage": "Unsupported output file format",
  "errorCode": "GES.8301"
}
```

## 5.2.7.4 Executing the DSL Algorithm

### Function

This API is used to provide flexible DSLs to help users design and run algorithms at low costs. For details about the DSL algorithm, see [DSL Syntax](#).

 **NOTE**

After the DSL algorithm is executed, you need to use the "Dumping HyG Algorithm Results" API to dump the DSL execution results to OBS.

### URI

POST /ges/v1.0/{project\_id}/hyg/{graph\_name}/dsl

**Table 5-471** URI parameters

| Parameter  | Mandator y | Type   | Description  |
|------------|------------|--------|--|
| project_id | Yes        | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes        | String | Graph name   |

### Request Parameters

**Table 5-472** Request body parameters

| Parameter     | Mandator y | Type   | Description   |
|---------------|------------|--------|---|
| scriptPath    | Yes        | String | Path of the DSL algorithm file that the user has written.                     |
| obsParameters | Yes        | Object | OBS authentication parameters. For details, see <a href="#">Table 5-473</a> . |

**Table 5-473** obsParameters parameter description

| Parameter | Mandator y | Type   | Description |
|-----------|------------|--------|-------------|
| accessKey | Yes        | string | AK          |
| secretKey | Yes        | string | SK          |

## Response Parameters

**Table 5-474** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Execution results <ul style="list-style-type: none"> <li>• If the execution is successful, the value is <b>success</b>.</li> <li>• If the execution fails, the value is <b>failed</b>.</li> </ul>                        |

## Example Request

Example request 1: Cancel a submitted job.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/dsl
{
  "scriptPath": "bucket/run_sssp.py",
  "obsParameters": {
    "accessKey": "XXX",
    "secretKey": "XXX"
  }
}
```

### NOTE

**SERVER\_URL**: URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "6-57222f3d-f6b8-41ba-b492-60ed9b879223"
}
```

**Status code: 400**

Example response for a failed request

```
HttpStatusCode: 400
{
```

```
"errorCode": "GES.8011",
"errorMessage": "graph : movie2 is not exist"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.7.5 DSL Syntax

#### 5.2.7.5.1 Graph Operation APIs

Graph operation APIs aim to provide users with an end-to-end process APIs for input, computation, and output.

### Graph Property Value Type

The current Python DSL supports three data types: int, float, and bool, which correspond to the basic data types int64\_t, double, and bool in C++.

### Combiner Type

The Combiner is used to perform local calculations on data during the computation process that satisfies the commutative and associative laws, reducing data exchange. The current Python DSL provides three types of Combiners: sum, max, and min, which support local calculations on int and float data.

### Graph Loading

```
hyg.analytics.graph.load_base_graph(graph_name:str)
```

This API returns a BaseGraph object, where graph\_name specifies the path of the graph data.



### 5.2.7.5.2 API for Running Custom Algorithms (Currently, the Pregel Programming Model Is Supported)

When the built-in graph analysis algorithms cannot meet the user's needs, HyG allows users to implement custom algorithms using the Python language in the Pregel programming model. The API for running custom Pregel algorithms is:

```
BaseGraph.run_pregel(model:class, result_filter=None, debug_mode=False)
```

The **model** is a class type and a subclass of **hyg.analytics.model.PregelModel**. Before calling the **run\_pregel** interface, users need to first implement the algorithm calculation logic in their own PregelModel subclass, as shown in Figure 1. First, users need to specify the vertex value type (**ntype**) and message type (**mtype**) in the **@pregel\_type** decorator. The **mtype** can be left unset, and it will default to the same as **ntype**. Next, users need to implement the vertex-centered **init** and **compute** methods in **UserPregelAlgorithm**. The **init** method is executed only once at the beginning of the algorithm, while the **compute** method is iterated multiple times. If these methods are not implemented, they default to empty. Additionally, the **combiner** parameter combines messages sent to the same target vertex to reduce communication overhead. By default, it is set to **None**, which means messages are not combined.

Pregel programming model:

```
from hyg.analytics.model import pregel_types, PregelModel
@pregel_types(ntype=None, mtype=None, combiner=None)
class PregelModel:
    @staticmethod
    def init(ctx, nid):
        pass

    @staticmethod
    def compute(ctx, nid, msgs):
        pass
```

The **result\_filter** parameter is of type **function** and supports lambda functions. It takes (**ctx, nid**) as input and returns a boolean value, which is used to filter the Pregel calculation results. The **debug\_mode** is of type **bool**. When set to **True**, the UDF is not compiled into native code but is interpreted by the Python interpreter. In this case, the number of concurrent threads in the HyG framework process is forced to 1, and users can use **print** statements to print debugging information in the UDF. When **debug\_mode** is set to **False**, using **print** statements in the UDF will result in an error.

In addition, the BaseGraph also has an interface **BaseGraph.nid(ext\_id:str) -> int**, which is used to obtain the internal ID of a vertex.

### 5.2.7.5.3 Pregel Programming API

When implementing the **init** and **compute** methods in **UserPregelAlgorithm**, users mainly rely on the **PregelContext** object, which provides the following APIs:

**Table 5-475** PregelContext API

| Method and Attribute            | Description  | Remarks   |
|---------------------------------|--|---|
| ext_id(nid)->int                | Obtains the user-defined external ID of the current vertex. Only IDs that can be converted to the <b>int</b> type are supported. | Basic operations on graph data (values and topologies)  |
| value(nid)->int/float/bool      | Obtains the value of the current vertex.   |   |
| set_value(nid, new_value)->None | Sets the value of the current vertex.  |   |
| out_edges(nid)-> List[int]      | Obtains the list of outgoing edges of the current vertex.  |   |
| edge_dst(eid)->int              | Obtains the target vertex of the current edge.   |   |
| num_nodes                       | Obtains the number of vertices in a full graph.  |   |
| num_edges                       | Obtains the total number of edges in a full image.   |   |
| send(dst_nid, msg)->None        | Sends messages to the target vertex.   |   |
| halt(nid) ->None                | Sets the current vertex to the <b>halt</b> state.  | When all vertices in the graph are in the <b>halt</b> state and there are no messages, or when supersteps reach the maximum number of iterations, the algorithm terminates. |
| superstep -> int                | Obtain the current number of supersteps.   |   |

#### 5.2.7.5.4 Programming Example of Creating Custom Graph Analysis Algorithms

##### Creating a Custom SSSP Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")
# Obtain the internal ID based on the external ID.
SOURCE_NODE = graph.nid(100)

# Implement the custom SSSP algorithm based on the Pregel model and set n_type to int.
# By default, set m_type to the same value as n_type, and set combiner to min.
@pregel_types(n_type=int, combiner=min)
```

```
class PregelSSSP(PregelModel):
    @staticmethod
    def compute(ctx, nid, msgs):
        if ctx.superstep == 0:
            ctx.set_value(nid, 10000)

        min_dist = 0 if nid == SOURCE_NODE else 10000

        if len(msgs) != 0:
            min_dist = min(min_dist, min(msgs))

        if min_dist < ctx.value(nid):
            ctx.set_value(nid, min_dist)
            for e in ctx.out_edges(nid):
                ctx.send(ctx.edge_dst(e), min_dist + 1)

        ctx.halt(nid)

# Run the custom SSSP algorithm and obtain the result.
result = graph.run_pregel(PregelSSSP)
print(type(result), result)
```

## Creating a Custom PageRank Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")

# Implement the custom PageRank algorithm based on the Pregel model.
# Set ntype to float and retain the default value for mtime.
# Set combiner to sum.
@pregel_types(ntype=float, combiner=sum)
class PregelPageRank(PregelModel):
    @staticmethod
    def init(ctx, nid):
        ctx.set_value(nid, 1.0)

    @staticmethod
    def compute(ctx, nid, msgs):
        if ctx.superstep >= 1:
            new_value = 0.85 * sum(msgs) + 0.15 / ctx.num_nodes

            if (abs(new_value - ctx.value(nid)) < 0.001
                or ctx.superstep == 1000):
                ctx.halt(nid)
                return

            ctx.set_value(nid, new_value)
            # Only when debug_mode is set to True,
            # You can use print debugging statements in the custom algorithm.
            print(f"in step {ctx.superstep}, node nid {nid}, "
                  f"value {ctx.value(nid)}")
            out_edges = ctx.out_edges(nid)
            if len(out_edges) > 0:
                new_msg = ctx.value(nid) / len(out_edges)
                for e in out_edges:
                    ctx.send(ctx.edge_dst(e), new_msg)

# Run the custom PageRank algorithm in debug mode and obtain the result.
result = graph.run_pregel(PregelPageRank, debug_mode=True)
print(type(result), result)
```

## Creating a Custom K-Hop Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
```

```

from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")

# Set global values for the custom algorithm.
INFINITY_VALUE = False
SOURCE_NODE = graph.nid("100")
MAX_HOP = 3

# Implement the custom K-Hop algorithm based on the Pregel model.
# Set ntype to bool, indicating whether the vertex value is traversed.
# The parameter combiner cannot be set to bool. Even if it is set, the value will be ignored.
@pregel_types(ntype=bool)
class PregelKHop(PregelModel):
    @staticmethod
    def compute(ctx, nid, msgs) -> None:
        if ctx.superstep == 0:
            if nid == SOURCE_NODE:
                ctx.set_value(nid, True)
                if ctx.superstep < MAX_HOP:
                    for e in ctx.out_edges(nid):
                        ctx.send(ctx.edge_dst(e), True)
            else:
                ctx.set_value(nid, INFINITY_VALUE)
        else:
            if ctx.value(nid) == INFINITY_VALUE:
                ctx.set_value(nid, True)
                if ctx.superstep < MAX_HOP:
                    for e in ctx.out_edges(nid):
                        ctx.send(ctx.edge_dst(e), True)

        ctx.halt(nid)

# Run the custom K-Hop algorithm and filter the results using the result_filter parameter.
result = graph.run_pregel(PregelKHop,
    result_filter=
        lambda ctx, nid: nid != SOURCE_NODE
            and ctx.value(nid))
print(len(result), result.keys())

```

## 5.2.8 HyG Job Management APIs

### 5.2.8.1 Dumping HyG Algorithm Results

#### Function

This API is used to dump the execution results of the algorithm (jobId) to OBS.

#### URI

POST /ges/v1.0/{project\_id}/hyg/{graph\_name}/jobs/{job\_id}/export-result

**Table 5-476** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| graph_name | Yes       | String | Graph name  |
| job_id     | Yes       | String | Job ID of the algorithm task in the response result |

## Request Parameters

**Table 5-477** Request body parameters

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| exportPath    | Yes       | String  | Dump path   |
| obsParameters | Yes       | String  | OBS authentication parameters. For details, see <a href="#">Table 5-478</a> .   |
| erase         | No        | Boolean | Whether to delete the original job result after dumping. The value can be <b>true</b> or <b>false</b> . The default value is <b>true</b> , indicating that the job result is deleted and resources are released by default. |

**Table 5-478** obsParameters parameter description

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | string | AK          |
| secretKey | Yes       | string | SK          |

## Response Parameters

**Table 5-479** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| errorCode | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId     | String | ID of an asynchronous job. This parameter is left blank if the request fails.<br><b>NOTE</b><br>You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see <a href="#">Job Management APIs</a> . |

## Example Request

Dumps algorithm results and returns the job ID.

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}/export-result
{
  "exportPath": "demo_movie/",
  "erase": true,
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

### NOTE

**SERVER\_URL:** URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

### Status code: 400

Example response for a failed request

```
HttpStatusCode: 400
{
  "errorCode": "GES.8011",
  "errorMessage": "graph : movie2 is not exist"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.8.2 Canceling a HyG Job

#### Function

This API is used to cancel a submitted HyG job. (Currently, only jobs in a queue can be canceled.)

#### URI

DELETE /ges/v1.0/{project\_id}/hyg/{graph\_name}/jobs/{job\_id}

**Table 5-480** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |
| job_id     | Yes       | String | Job ID of the algorithm task in the response result  |

#### Request Parameters

None

## Response Parameters

**Table 5-481** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>• If the execution succeeds, this parameter may be left blank.</li> <li>• If the execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Execution results <ul style="list-style-type: none"> <li>• If the execution is successful, the value is <b>success</b>.</li> <li>• If the execution fails, the value is <b>failed</b>.</li> </ul>                        |

### Example Request

- Example request 1: Cancel a submitted job.  
DELETE http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs/{job\_id}
- Example request 2: Cancel all jobs in a queue.  
DELETE http://{SERVER\_URL}/ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs

 **NOTE**

**SERVER\_URL:** URL for accessing the graph. For details about its value, see [Using Service Plane APIs](#).

### Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

**Status code: 400**

Example response for a failed request

```
HttpStatusCode: 400
{
  "errorMessage": "Graph [
  {project_id}
  -movie1] does not exist, please check project_id and graph_name.",
  "errorCode": "GES.8000",
```



```
"result": "failed"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

## 5.2.9 Native Algorithm APIs

### 5.2.9.1 Executing an Algorithm

#### Function

This API is used to run specified algorithms based on entered parameters.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-algorithm

**Table 5-482** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Example Request

Run the **shortest\_path** algorithm to find the shortest route between vertices **46** and **138**, taking into account the direction of the edges.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName":"shortest_path",
  "executionMode": "sync",
  "parameters":{
    "source":"46",
    "target":"138",
    "directed":true
  }
}
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

## Response Parameters

Table 5-483 Response body parameters

| Parameter    | Type    | Description  |
|--------------|---------|--|
| errorMessage | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId        | String  | ID of the algorithm execution job. This parameter is left blank when the request fails. <p><b>NOTE</b></p> You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |
| jobType      | Integer | Job type. This parameter is left blank when the request fails.   |

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

### Status code: 4200

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

## 5.2.9.2 Algorithm API Parameter References

### 5.2.9.2.1 Common Algorithm Parameters

## Request Example

```
{
  "algorithmName": "XXX",
  "parameters": {
    ...
  }
}
```

## Request Parameters

**Table 5-484** Request body parameters

| Parameter     | Mandatory | Type   | Description   |
|---------------|-----------|--------|---|
| algorithmName | Yes       | String | Algorithm name.<br>Available values are as follows (algorithm names you can call): <ul style="list-style-type: none"> <li>• shortest_path</li> <li>• shortest_path_of_vertex_sets</li> <li>• common_neighbors_of_vertex_sets</li> </ul> |
| parameters    | Yes       | Object | Algorithm parameters. For details, see the parameter description of each algorithm.   |

**Table 5-485** New Body parameters of version 2.1.7

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| executionMode | No        | String  | <ul style="list-style-type: none"> <li>• <b>sync</b>: synchronous</li> <li>• <b>async</b>: asynchronous</li> </ul> The default value is <b>async</b> .<br>Supported algorithms are as follows (algorithm names you can call): <ul style="list-style-type: none"> <li>• shortest_path</li> <li>• shortest_path_of_vertex_sets</li> </ul>   |
| offset        | No        | Integer | Synchronization result offset. The default value is <b>0</b> .<br><b>NOTE</b><br>This parameter is available when <b>executionMode</b> is <b>sync</b> .<br>Supported algorithms are as follows (algorithm names you can call): <ul style="list-style-type: none"> <li>• shortest_path</li> <li>• shortest_path_of_vertex_sets</li> <li>• common_neighbors_of_vertex_sets</li> </ul> |

| Parameter | Mandatory | Type    | Description   |
|-----------|-----------|---------|---|
| limit     | No        | Integer | <p>Maximum number of returned synchronization results. The maximum value is <b>100000</b>. The default value is <b>100000</b>.</p> <p><b>NOTE</b><br/>This parameter is available when <b>executionMode</b> is <b>sync</b>.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> <li>• shortest_path</li> <li>• shortest_path_of_vertex_sets</li> <li>• common_neighbors_of_vertex_sets</li> </ul> |

## Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the **job\_id** returned by the algorithm to obtain the algorithm execution result.

### Status code: 200

Example response (successful request)

```
{
  "data": {
    "outputs": {
      $response_data //Result of each algorithm. The results vary with the algorithm.
      "runtime": 1.365867,
      "data_return_size": 3,
      "data_offset": 0,
      "data_total_size": 100
    }
  },
  "status": "complete"
}
```

### NOTE

**response\_data** indicates the result of each algorithm. The results vary with algorithms.

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Running algorithm [XXXX] error: YYYYYYYYYY!",
  "errorCode": "GES.8301"
}
```

## Response Parameters

**Table 5-486** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.                           |
| errorCode    | String | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.                              |
| status       | String | Returned job status for a successful query. Possible values are <b>waiting</b> , <b>running</b> , and <b>complete</b> . This parameter is left blank when the query fails. |
| data         | Object | Algorithm execution result. This parameter is left blank when the query fails.   |

### 5.2.9.2.2 Shortest Path

**Table 5-487** parameters parameter description

| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| source     | Yes       | String  | Source vertex ID of a path   |
| target     | Yes       | String  | Target vertex ID of a path   |
| directed   | No        | Boolean | Whether to consider the edge direction. The value is <b>true</b> .   |
| timeWindow | No        | Object  | Time window for time filtering. For details, see <a href="#">Table 5-488</a> .<br><b>NOTE</b><br><b>timeWindow</b> does not support the shortest path with weight. That is, parameters <b>timeWindow</b> and <b>weight</b> cannot be both specified. |

**Table 5-488** timeWindow parameter description

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| filterName | Yes       | String | Character string: The property on the corresponding vertex/edge is used as the time. |

| Parameter  | Mandatory | Type   | Description   |
|------------|-----------|--------|---|
| filterType | No        | String | Filters vertices or edges. The default value is <b>BOTH</b> . <ul style="list-style-type: none"> <li>• <b>V</b>: filtering by vertex</li> <li>• <b>E</b>: filtering by edge</li> <li>• <b>BOTH</b>: filtering by vertex and edge</li> </ul> |
| startTime  | No        | String | Start time, which is a string of the date type or a timestamp.  |
| endTime    | No        | String | End time, which is a string of the date type or a timestamp.  |

**Table 5-489** response\_data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| path      | List   | Shortest path. The format is as follows:<br>[vertexId,...]<br>where<br><b>vertexId</b> is of the string type. |
| source    | String | Source vertex ID  |
| target    | String | Target vertex ID  |

### 5.2.9.2.3 Shortest Path of Vertex Set

**Table 5-490** Parameter description

| Parameter | Mandatory | Description          | Type   | Value Range   | Default Value |
|-----------|-----------|----------------------|--------|---|---------------|
| sources   | Yes       | Source vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |
| targets   | Yes       | Target vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br>The maximum ID number is 100000. | -             |

| Parameter  | Mandatory | Description                            | Type    | Value Range                                    | Default Value |
|------------|-----------|--|---------|--|---------------|
| directed   | No        | Whether to consider the edge direction | Boolean | true or false                                  | false         |
| timeWindow | No        | Time window used for time filtering    | Object  | For details, see <a href="#">Table 5-491</a> . | -             |

**Table 5-491** timeWindow parameter description

| Parameter  | Mandatory | Description                                       | Type   | Value Range  | Default Value |
|------------|-----------|---|--------|--|---------------|
| filterName | No        | Name of the time property used for time filtering | String | Character string: The property on the corresponding vertex/edge is used as the time.                         | -             |
| filterType | No        | Filtering by vertex or edge                       | String | <b>V</b> : filtering by vertex<br><b>E</b> : filtering by edge<br><b>BOTH</b> : filtering by vertex and edge | <b>BOTH</b>   |
| startTime  | No        | Start time  | String | Date character string or timestamp   | -             |
| endTime    | No        | End time  | String | Date character string or timestamp   | -             |



**Table 5-492** response\_data parameter description

| Parameter | Type   | Description   |
|-----------|--------|---|
| path      | List   | Shortest path. The format is as follows:<br>[vertexId,...]<br>where<br><b>vertexId</b> is of the string type. |
| source    | String | Source vertex ID  |
| target    | String | Target vertex ID  |

### 5.2.9.2.4 Common Neighbors of Vertex Sets

**Table 5-493** Parameter description

| Parameter          | Mandatory | Description          | Type   | Value Range   | Default Value |
|--------------------|-----------|----------------------|--------|---|---------------|
| sources<br>(2.2.6) | Yes       | Source vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Alice, Nana</b> .<br><br>The maximum ID number is 100000. | -             |
| targets<br>(2.2.6) | Yes       | Target vertex ID set | String | The value is in the standard CSV format. IDs are separated by commas (,), for example, <b>Mike,Amy</b> .<br><br>The maximum ID number is 100000.    | -             |

| Parameter           | Mandatory | Description                            | Type    | Value Range   | Default Value |
|---------------------|-----------|--|---------|---|---------------|
| restricted (2.2.13) | No        | Whether other constraints are included | Boolean | <p><b>true</b> or <b>false</b></p> <ul style="list-style-type: none"> <li><b>false</b>: There is no additional constraint. The found common neighbors are the intersection of the neighborhoods corresponding to the source vertex set and target vertex set.</li> <li><b>true</b>: There are additional constraints. The found common neighbors are not only the intersection of the neighborhoods corresponding to the source vertex set and target vertex set, but each vertex in the common neighbor set has at least two neighboring vertices in the source vertex set and target vertex set.</li> </ul> | true          |

**Table 5-494** response\_data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| vertices         | List    | Common neighbor vertices. The format is as follows:<br>[vertexId,...],<br>where <b>vertexId</b> is of the string type. |
| common_neighbors | Integer | Number of common neighbors   |

## 5.2.10 Graph Statistics APIs

### 5.2.10.1 Querying General Information About a Graph

#### Function

This API is used to query the general information about a graph, such as the numbers of vertices and edges.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/summary?  
label\_details={labelDetails}

**Table 5-495** URI parameters

| Parameter    | Mandatory | Type    | Description   |
|--------------|-----------|---------|---|
| project_id   | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> .  |
| graph_name   | Yes       | String  | Graph name  |
| labelDetails | No        | Boolean | Whether to return the number of vertices and edges under each label. The default value is <b>false</b> . If this parameter is set to <b>true</b> , the numbers of vertices and edges under each label are returned. |

#### Example Request

Query general information about a graph, such as the numbers of vertices and edges. The value **true** indicates that the numbers of vertices and edges of different labels are returned.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details=true
```

#### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Response Parameters

**Table 5-496** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Querying Job Status on the Service Plane</a> .             |

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8001"
}
```

## Status Codes

| Return Value     | Description               |
|------------------|---------------------------|
| 400 Bad Request  | Request error.            |
| 401 Unauthorized | Authorization failed.     |
| 403 Forbidden    | No operation permissions. |
| 404 Not Found    | No resources found.       |

| Return Value              | Description            |
|---------------------------|------------------------|
| 500 Internal Server Error | Internal server error. |
| 503 Service Unavailable   | Service unavailable.   |

## Error Codes

See [Error Codes](#).

### 5.2.10.2 Querying the Graph Version

#### Function

This API is used to query the graph version.

#### URI

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/version

**Table 5-497** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

None

#### Response Parameters

**Table 5-498** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |

| Parameter | Type   | Description  |
|-----------|--------|--|
| errorCode | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| version   | String | Query results. This parameter is left blank when the request fails.  |

## Example Request

Query the graph version.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/version
```

### NOTE

**SERVER\_URL:** Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "version": "2.0.0"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |

| Return Value            | Description          |
|-------------------------|----------------------|
| 503 Service Unavailable | Service unavailable. |

## Error Codes

See [Error Codes](#).

## 5.2.11 Graph Operation APIs

### 5.2.11.1 Importing a Graph

#### Function

This API is used to import graph data.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=import-graph

**Table 5-499** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Request Parameters

**Table 5-500** Request body parameters

| Parameter     | Mandatory | Type   | Description  |
|---------------|-----------|--------|--|
| edgesetPath   | No        | String | Edge file directory or name  |
| edgesetFormat | No        | String | Format of the edge data set. Currently, only the CSV format is supported. The CSV format is used by default. |
| vertexsetPath | No        | String | Vertex file directory or name  |

| Parameter       | Mandatory | Type      | Description  |
|-----------------|-----------|-----------|--|
| vertexsetFormat | No        | String    | Format of the vertex data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.  |
| schemaPath      | No        | String    | OBS path of the metadata file of the new data  |
| logDir          | No        | String    | Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.   |
| parallelEdge    | No        | Object    | Repetitive edge processing   |
| action          | No        | String    | Way of processing repetitive edges. If the value of <b>action</b> is <b>override</b> , the previous repetitive edges are overwritten. If the value of <b>action</b> is <b>ignore</b> , repetitive edges are ignored. This means if an edge already exists, the edge will be ignored and will not be overwritten. |
| ignoreLabel     | No        | Boolean   | Whether to ignore labels on repetitive edges. The value is <b>false</b> .<br><b>false</b> : Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.  |
| delimiter       | No        | Character | Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).   |
| trimQuote       | No        | Character | Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.  |



| Parameter     | Mandatory | Type    | Description  |
|---------------|-----------|---------|--|
| offline       | No        | Boolean | Whether to import a graph offline. The value is <b>false</b> .<br><b>false</b> : Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import. |
| obsParameters | Yes       | Object  | OBS parameters   |

**Table 5-501** parallelEdge parameter type

| Parameter     | Mandatory | Type       | Description  |
|---------------|-----------|------------|--|
| sortKeyColumn | No        | Int/String | Index of the sort key column in the edge file. The value starts from 1. If this parameter is set to <b>last Column</b> , the sort key column is the last column.<br>The options are as follows: <ul style="list-style-type: none"> <li>• A positive integer</li> <li>• A positive integer of the string type</li> <li>• "<b>lastColumn</b>"</li> </ul> |
| sortKeyType   | No        | String     | Type of the sortKey value. The options are <b>int</b> and <b>string</b> .  |

**Table 5-502** obsParameters parameters

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | String | User AK     |
| secretKey | Yes       | String | User SK     |

 **NOTE**

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

## Response Parameters

**Table 5-503** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                              |

## Example Request

Import a graph. The edge file directory is **testbucket/demo\_movie/edges/** and the edge data set format is CSV; the vertex file directory is **testbucket/demo\_movie/vertices/** and the vertex data set format is CSV; the OBS path of the metadata file of the new data is **testbucket/demo\_movie/incremental\_data\_schema.xml** and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true
  },
  "delimiter": ",",
  "trimQuote": "\"\"",
  "offline": true,
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400
{
  "errorCode": "GES.8013",
  "errorMessage": "graph [movie2] is not found"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.11.2 Clearing a Graph

#### Function

This API is used to clear graph data.

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=clear-graph

**Table 5-504** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-505** Request body parameter

| Parameter     | Mandatory | Type    | Description   |
|---------------|-----------|---------|---|
| clearMetadata | No        | Boolean | Whether to clear schema data. The default value is <b>false</b> . |

## Response Parameters

**Table 5-506** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                              |

## Example Request

Clear a graph by deleting its schema data.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph
{
```

```
"clearMetadata": true  
}
```

## Example Response

### Status code: 200

Example response (successful request)

```
Http Status Code: 200  
{  
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"  
}
```

### Status code: 400

Example response (failed request)

```
Http Status Code: 400  
{  
  "errorCode": "GES.8012",  
  "errorMessage": "graph [movie2] is not found"  
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.11.3 Exporting a Graph

#### Function

This API is used to export graph data.

 **NOTE**

If you choose to export CSV files to your local host, the files are opened using the spreadsheet software by default. You are advised to open the files in a text editor. If the data contains special characters such as plus signs (+), minus signs (-), equal signs (=), and at signs (@), the data will be parsed into formulas by the software. To ensure system security, pay attention to the following when opening such files:

1. Do not select **Enable Dynamic Data Exchange Server Launch (not recommended)**.
2. Do not select **Enable** or **Yes** if a dialog box indicating a security issue is displayed.

## URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=export-graph

**Table 5-507** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

**Table 5-508** Request body parameters

| Parameter       | Mandatory | Type   | Description  |
|-----------------|-----------|--------|--|
| graphExportPath | Yes       | String | OBS path a graph is exported to  |
| edgeSetName     | Yes       | String | Name of the exported edge data set   |
| vertexSetName   | Yes       | String | Name of the exported vertex data set   |
| schemaName      | Yes       | String | Name of the exported metadata file   |
| paginate        | No        | Object | Pagination parameters. For details, see <a href="#">paginate parameter description</a> . |
| obsParameters   | Yes       | String | OBS parameters. For details, see <a href="#">obsParameters parameter description</a> .   |
| accessKey       | Yes       | String | User AK  |

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| secretKey | Yes       | String | User SK     |

**Table 5-509** paginate parameter description

| Parameter       | Mandatory | Type    | Description  |
|-----------------|-----------|---------|--|
| rowCountPerFile | No        | Integer | Maximum number of rows in each file when graphs are exported by page. The default value is <b>100000</b> .   |
| numThread       | No        | Integer | Number of concurrent threads when graphs are exported by page. The default value is <b>8</b> . The maximum value cannot exceed the number of kernels.    |
| maxSizePerFile  | No        | Integer | Maximum size of each file when graphs are exported by page, in bytes. By default, the size cannot exceed the maximum size of the file that was imported. |

**Table 5-510** obsParameters parameter description

| Parameter | Mandatory | Type   | Description |
|-----------|-----------|--------|-------------|
| accessKey | Yes       | string | AK          |
| secretKey | Yes       | string | SK          |

 **NOTE**

**obsParameters**, **accessKey**, and **secretKey** are mandatory in OBS scenarios and optional in HDFS scenarios.

## Response Parameters

**Table 5-511** Response body parameters

| Parameter    | Mandatory | Description  |
|--------------|-----------|--|
| errorMessage | No        | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | No        | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | No        | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                       |

## Example Request

Export a graph. The OBS path for exporting the graph is **demo\_movie/**, the name of the exported edge data set is **set\_edge**, the name of the exported vertex data set is **set\_vertex**, and the name of the exported metadata file is **set\_schema.xml**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "paginate":{
    "numThread":8,
    "rowCountPerFile":1000000
  },
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

### Example request 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```



```
}  
}
```

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"  
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorCode": "GES.8301",  
  "errorMessage": "Graph [10001-movie] does not exist, please check projectId and graphName."  
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.11.4 Creating a Graph

#### Function

This API is used to create a graph.

#### URI

POST /ges/v1.0/{project\_id}/graphs

**Table 5-512** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 5-513** Request body parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| graphName | Yes       | String  | Graph name. The value can contain 4 to 32 characters and must start with a letter. Only letters, digits, and underscores ( <code>_</code> ) are allowed. |
| idType    | Yes       | String  | ID type. The value can be <b>hash</b> or <b>fixedLengthString</b> .  |
| idLength  | No        | Integer | This parameter is mandatory if <b>id_type</b> is <b>fixedLengthString</b> . The value ranges from 1 to 128.  |

## Response Parameters

**Table 5-514** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| result       | String | Query result. The value is <b>success</b> for a successful query and <b>failed</b> for a failed query.   |

## Example Request

```
POST /ges/v1.0/{project_id}/graphs
{
  "graphName": "xxx",
  "idType": "fixedLengthString",
  "idLength": 20
}
```

## Example Response

### Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

### Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [movie] already exists",
  "errorCode": "GES.8012",
  "result": "failed"
}
```

## Status Codes

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes for Service Plane APIs](#).

### 5.2.11.5 Deleting a Graph

#### Function

This API is used to delete a graph.

## URI

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}

**Table 5-515** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

## Request Parameters

None

## Response Parameters

**Table 5-516** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code <ul style="list-style-type: none"> <li>If the execution succeeds, this parameter may be left blank.</li> <li>If the execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of an asynchronous job<br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                                       |

## Example Request

DELETE /ges/v1.0/{project\_id}/graphs/{graph\_name}

## Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "3-f78ec641-ed66-4983-bf93-7f9b3a716c78000000019090"
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [10001-movie1006] does not exist, please check project_id and graph_name.",
  "errorCode": "GES.8301"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes for Service Plane APIs](#).

**5.2.11.6 Listing Graphs**

**Function**

This API is used to list graphs.

**URI**

GET /ges/v1.0/{project\_id}/graphs

**Table 5-517** URI parameter

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

None

## Response Parameters

**Table 5-518** Response body parameters

| Parameter | Mandatory                      | Description  |
|-----------|--------------------------------|--|
| graphs    | Array of <b>graphs</b> objects | Returned graph list  |
| result    | String                         | Query result. The value is <b>success</b> for a successful query and <b>failed</b> for a failed query. |

**Table 5-519** graphs

| Parameter         | Mandatory | Description         |
|-------------------|-----------|---------------------|
| graphName         | String    | Graph name          |
| idType            | String    | ID type             |
| idLength          | Integer   | ID length           |
| averageImportRate | Integer   | Average import rate |

## Example Request

```
GET /ges/v1.0/{project_id}/graphs
```

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "graphs": [
    {
      "graphName": "movieee",
      "idType": "fixedLengthString",
      "idLength": 20,
      "averageImportRate": 0.0
    },
    {
      "graphName": "ldbc",
      "idType": "fixedLengthString",
      "idLength": 20,
      "averageImportRate": 0.0
    }
  ],
}
```

```
"result": "success"
}
```

**Status code: 400**

Example response for a failed request

```
Http Status Code: 400
{
  "result": "failed"
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes for Service Plane APIs](#).

**5.2.12 Job Management APIs**

**5.2.12.1 Querying the Job List**

**Function**

After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned.

**URI**

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?
limit={limit}&offset={offset}
```

**Table 5-520** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| job_id     | Yes       | String | ID of the job corresponding to the response  |

## Request Parameters

For details, see the URI parameters.

## Response Parameters

**Table 5-521** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| result       | String | Query result. If the query is successful, the value is <b>success</b> . If the query fails, the value is <b>failed</b> .  |
| jobs         | Object | Job status list stored in the system. If execution succeeds, this parameter is contained in the response. <a href="#">Table 5-522</a> describes the structure of a single jobs field.                       |

**Table 5-522** Job status structure

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job name.  |
| request   | Object | Request content, including the command, URL, and body. |



| Parameter | Type   | Description   |
|-----------|--------|---|
| status    | String | Job status. The value can be <b>pending</b> , <b>running</b> , or <b>complete</b> . |

## Example Request

Query the job list and return the job ID and status of each job.

```
GET /ges/v1.0/{project_id}/graphs/movie/jobs/status
```

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "jobs": [
    {
      "jobId": "62582163123991943683d0f9aa3-f701-48be-a662-360e6a0455da",
      "status": "complete",
      "request": {
        "command": "import_graph",
        "url": "/ges/v1.0/10001/graphs/moviejx/action?action_id=import-graph",
        "body": {
          "edgesetPath": "file:///root/ges-install/auDatas/ranking_edge-sp.csv",
          "vertexsetPath": "file:///root/ges-install/auDatas/movies_vertex_new.csv",
          "schemaPath": "file:///root/ges-install/auDatas/schema_aikv.xml.bak"
        }
      }
    },
    {
      "jobId": "62582163123991943683fe74caf-f4d3-48b3-b3ee-66daaedcd2ca",
      "status": "complete",
      "request": {
        "command": "import_graph",
        "url": "/ges/v1.0/10001/graphs/moviejx/action?action_id=import-graph",
        "body": {
          "edgesetPath": "file:///root/ges-install/auDatas/ranking_edge-sp.csv",
          "vertexsetPath": "file:///root/ges-install/auDatas/movies_vertex_new.csv",
          "schemaPath": "file:///root/ges-install/auDatas/schema_aikv.xml.bak"
        }
      }
    },
    {
      "jobId": "6258216312399194368daa80df3-e3bd-440d-9764-74f4622a550f",
      "status": "complete",
      "request": {
        "command": "import_graph",
        "url": "/ges/v1.0/10001/graphs/moviejx/action?action_id=import-graph",
        "body": {
          "edgesetPath": "file:///root/ges-install/auDatas/ranking_edge-sp.csv",
          "vertexsetPath": "file:///root/ges-install/auDatas/movies_vertex_new.csv",
          "schemaPath": "file:///root/ges-install/auDatas/schema_aikv.xml.bak"
        }
      }
    },
    {
      "jobId": "62582163123991943680ed2761f-01f7-4fbf-b867-0a9aae6d9c12",
      "status": "complete",
      "request": {
        "command": "import_graph",
```

```

    "url": "/ges/v1.0/10001/graphs/moviejx/action?action_id=import-graph",
    "body": {
      "edgesetPath": "file:///root/ges-install/auDatas/ranking_edge-sp.csv",
      "vertexsetPath": "file:///root/ges-install/auDatas/movies_vertex_new.csv",
      "schemaPath": "file:///root/ges-install/auDatas/schema_aikv.xml.bak"
    }
  },
  "result": "success"
}

```

**Status code: 400**

Example response for a failed request

```

Http Status Code: 400
{
  "errorMessage": "graph : movidde not exist",
  "errorCode": "GES.8000",
  "result": "failed"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.2.12.2 Querying Job Status**

**Function**

This API is used to query the execution status of a job. After asynchronous APIs such as those for querying vertices and edges or executing algorithms are used, job IDs are returned. You can use the job ID to query the execution status of each job.

**URI**

GET /ges/v1.0/{project\_id}/graphs/{graph\_name}/jobs/{job\_id}/status?  
offset=*offset*&limit=*limit*

**Table 5-523** URI parameters

| Parameter  | Mandatory | Type    | Description  |
|------------|-----------|---------|--|
| project_id | Yes       | String  | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| job_id     | Yes       | String  | Job ID   |
| offset     | No        | Integer | Offset of a query. The default value is <b>0</b> .   |
| limit      | No        | Integer | Maximum number of records that can be queried. The default value is <b>100000</b> .                      |

## Response Parameters

**Table 5-524** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| status       | String | Returned job status after the query is successful. Possible values: <ul style="list-style-type: none"> <li><b>pending</b></li> <li><b>running</b></li> <li><b>success</b></li> <li><b>failed</b></li> </ul> This parameter is left blank when the query fails. |
| data         | Object | Algorithm execution result. This parameter is left blank when the query fails.   |

**Table 5-525** data parameter description

| Parameter        | Type    | Description  |
|------------------|---------|--|
| vertices         | List    | Vertex-associated algorithm result                         |
| edges            | List    | Edge-associated algorithm result                           |
| outputs          | Object  | Other results  |
| data_return_size | Integer | Number of records returned after a query                   |
| data_offset      | Integer | Result offset of a query                                   |
| data_total_size  | Integer | Total amount of result data generated by asynchronous jobs |

## Example Request

Query the execution status of a job. The query offset is **0**, and the maximum number of returned results is **2**.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=0&limit=2
```

### NOTE

**SERVER\_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

## Example Response

**Status code: 200**

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "data_return_size": 2,
      "vertices": [
        {
          "id": "Sarah",
          "label": "user",
          "properties": {
            "Occupation": [
              "other or not specified"
            ],
            "Name": [
              "Sarah"
            ],
            "Zip-code": [
              "55105"
            ],
            "Gender": [
              "F"
            ],
            "Age": [
              "18-24"
            ]
          }
        }
      ]
    }
  }
}
```

```

    },
    {
      "id": "Sidney",
      "label": "user",
      "properties": {
        "Occupation": [
          "writer"
        ],
        "Name": [
          "Sidney"
        ],
        "Zip-code": [
          "85296"
        ],
        "Gender": [
          "M"
        ],
        "Age": [
          "18-24"
        ]
      }
    }
  ],
  "data_offset": 0,
  "data_total_size": 19
}
},
"status": "success"
}

```

**Status code: 400**

Example response for a failed request

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}

```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

**Error Codes**

See [Error Codes](#).

**5.2.13 Cypher Operation APIs**

### 5.2.13.1 Executing Cypher Queries

#### Function

Cypher is a widely used declarative graph database query language. It can be used to query data in GES and returns results. Graph statistics are used in Cypher implementation. Currently, the label-based vertex and edge indexes are used during Cypher query and compilation. To use Cypher normally, create indexes by referring to [Cypher Prerequisites](#).

#### URI

POST /ges/v1.0/{project\_id}/graphs/{graph\_name}/action?action\_id=execute-cypher-query

**Table 5-526** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_name | Yes       | String | Graph name   |

#### Example Request

Execute a Cypher query. The Cypher statement is **match (n) return n limit 1**. The returned results are in the format that each element corresponds to a field in the row.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "parameters": {},
    "resultDataContents": ["row"],
    "includeStats": false
  }]
}
```

## Request Parameters

**Table 5-527** Request body parameter

| Parameter  | Mandatory | Type | Description   |
|------------|-----------|------|---|
| statements | Yes       | List | Statement group that contains one or more statements. The <a href="#">statements parameters</a> table describes the format of each element. |

**Table 5-528** statements parameters

| Parameter              | Mandatory | Type           | Description  |
|------------------------|-----------|----------------|--|
| statement              | Yes       | String         | Cypher statement   |
| parameters             | Yes       | Object         | Cypher statement parameters, which are used for parameterized queries. By default, this field is left blank.<br>For details, see <a href="#">parameterized queries</a> .   |
| resultDataContents     | No        | String or List | Format of the returned result. You can set one or more formats. Available values are <b>row</b> , <b>graph</b> , and <b>raw</b> (added in version 2.2.27).   |
| includeStats           | No        | Boolean        | Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.   |
| runtime                | No        | String         | Executor type. The value can be <b>map</b> or <b>slotted</b> . The default value is <b>map</b> . The slotted executor is supported since version 2.3.15.   |
| executionMode (2.2.23) | No        | String         | Execution mode. Set this parameter to <b>sync</b> for synchronous execution and to <b>async</b> for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For details about how to obtain the query result in asynchronous mode, see <a href="#">Querying Job Status on the Service Plane</a> . |

| Parameter      | Mandatory | Type | Description   |
|----------------|-----------|------|---|
| limit (2.2.23) | No        | Int  | Maximum number of results of the asynchronous query. This parameter is valid only when <b>executionMode</b> is <b>sync</b> . The default value is <b>100000</b> . |
| transactional  | No        | Bool | Whether the Cypher request is transactional. The default value is <b>false</b> . For details about Cypher transactions, see <a href="#">Cypher transactions</a> . |

 NOTE

- You can add the **explain** or **profile** prefix before a statement to display the query plan. The **explain** prefix displays only the query plan but does not execute the statement. The **profile** prefix displays the query plan and executes the statement.
- Description of the **runtime** field: Compared with the map executor, the slotted executor completes more statement data flow analysis in the plan generation phase of statements. In most cases, it executes faster while requiring less memory.
- In asynchronous mode (**executionMode** is **async**), cypher query results can be exported to CSV files (GES 2.3.4 or later supports this function). For details, see [Exporting Job Execution Results to Files](#). Currently, the following values can be returned:
  - Vertex and edge single-value properties, vertex and edge IDs, and group counts.
  - The current version does not support exporting object types. Objects are converted to null values in the CSV file.
- Cypher transactions (database edition only):
  - For database edition graphs, Cypher transactions are supported. You can set **transactional** to **true** to enable the function to ensure the atomicity of a single Cypher statement. Transactions supporting multiple Cypher statements are not available. Transactions in GES use a serializability isolation level.
  - The transaction time window is limited to 5s in the underlying storage engine of GES. Cypher transactions cannot last exceeding 5s. For complex queries, such as multi-hop queries, the running time may exceed 5s. The transaction times out and the submission fails.

In this case, you can use the **dbms.killQuery** program of Cypher to terminate a Cypher transaction (for details, see [Cypher API-Functions and Procedures](#)) and restore all changes caused by the Cypher request.

## Response Parameters

Table 5-529 Response body parameters

| Parameter | Type | Description  |
|-----------|------|--|
| results   | List | Each element of the list is the return result of a Cypher statement.               |
| errors    | List | Each element in the list contains the code and message information in string form. |



**Table 5-530** Elements of the results parameter

| Parameter       | Type    | Description  |
|-----------------|---------|--|
| columns         | List    | Name of a returned field   |
| data            | List    | Returned data value. Each element indicates a record.  |
| stats           | Object  | Addition, deletion, and modification statistics  |
| plan            | Object  | If the Cypher statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed. The profile feature is supported since version 2.3.12. |
| jobId(2.3.10)   | String  | Asynchronous job ID if the request is executed asynchronously  |
| jobType(2.3.10) | Integer | Type of the asynchronous job if the request is executed asynchronously   |

**Table 5-531** Elements of the data parameter

| Parameter   | Type   | Description  |
|-------------|--------|--|
| row         | List   | Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when <b>resultDataContents</b> is empty or contains <b>row</b> . |
| meta        | List   | Type of each field in a row. This parameter is displayed only when <b>resultDataContents</b> is empty or contains <b>row</b> .   |
| graph       | Object | Information returned in graph format. This parameter is displayed only when <b>resultDataContents</b> contains <b>graph</b> .  |
| raw(2.2.27) | List   | Information returned in raw format. This parameter is displayed only when <b>resultDataContents</b> contains <b>raw</b> .  |

**Table 5-532** stats elements in a response

| Parameter        | Type    | Description                               |
|------------------|---------|---|
| contains_updates | Boolean | Whether data is modified during the query |

| Parameter        | Type    | Description                             |
|------------------|---------|---|
| edges_created    | Integer | Number of created edges                 |
| edges_deleted    | Int     | Number of deleted edges                 |
| labels_set       | Integer | Number of labels that have been set     |
| properties_set   | Integer | Number of properties that have been set |
| vertices_created | Integer | Number of created vertices              |
| vertices_deleted | Integer | Number of deleted vertices              |

## Example Response

**Status code: 200**

Example response for a successful request (synchronous call)

Http Status Code: 200

```
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ],
  "stats": {
    "contains_updates": false,
    "edges_created": 0,
    "edges_deleted": 0,
    "labels_set": 0,
    "properties_set": 0,
    "vertices_created": 0,
    "vertices_deleted": 0
  }
}
```

```
],
"errors": []
}
```

**Status code: 200**

Example response for a successful request (asynchronous call)

```
Http Status Code: 200
{
  "results": [
    {
      "columns": [
        "jobId",
        "jobType"
      ],
      "jobId": "b64a5846-e306-4f87-b0f1-d595ee2a9910",
      "jobType": 1,
      "data": [
        {
          "row": [
            "b64a5846-e306-4f87-b0f1-d595ee2a9910",
            1
          ],
          "meta": [
            null,
            null
          ]
        }
      ]
    }
  ],
  "errors": []
}
```

**Status code: 400**

Example response (failed request)

```
Http Status Code: 400
{
  "results": [],
  "errors": [
    {
      "code": "GES.8904",
      "message": "Label index in vertices is not found."
    }
  ]
}
```

**Status Codes**

| Return Value              | Description               |
|---------------------------|---------------------------|
| 400 Bad Request           | Request error.            |
| 401 Unauthorized          | Authorization failed.     |
| 403 Forbidden             | No operation permissions. |
| 404 Not Found             | No resources found.       |
| 500 Internal Server Error | Internal server error.    |
| 503 Service Unavailable   | Service unavailable.      |

## Error Codes

See [Error Codes](#).

### 5.2.13.2 Basic Operations and Compatibility

#### Basic Operations

| Operation                                  | Cypher Statement   |
|--|--|
| Querying vertices                          | match (n) return n   |
| Querying edges                             | match (n)-[r]->(m) return n, r, m  |
| Querying paths                             | match (n:user)-[r]->(m:movie)-->(s:series) return n,r,m,s                |
| Querying vertices based on filter criteria | match(n:user) where n.userid>=5 return n                                 |
| Grouping and aggregating                   | match(n:movie) return n.genres, count(*)                                 |
| Deduplicating                              | match(n:movie) return distinct n.genres                                  |
| Sorting                                    | match(n:movie) return n order by n.movieid                               |
| Creating a vertex                          | create (n:user{userid:1}) return n                                       |
| Creating an edge                           | match (n:user{userid:15}),(m:movie{movieid:10}) create (n)-[r:rate]->(m) |
| Deleting a vertex                          | match (n:user{userid:1}) delete n  |
| Modifying labels                           | match (n:user{userid:1}) set n:movie return n                            |
| Modifying properties                       | match (n:user{userid:1}) set n.userid=2 return n                         |

#### Compatibility to Cypher

1. Cypher clauses

Cypher implements a couple of clauses. You can combine clauses to implement different query semantics, including vertex and edge filtering, multi-hop query, sorting and deduplication, and grouping and aggregation.

Currently, GES supports the Cypher clauses listed in the following table.

**Table 5-533** Supported Cypher clauses

| Clause | Support             | Example                  |
|--------|---------------------|--------------------------|
| match  | Partially supported | match (n:movie) return n |

| Clause          | Support             | Example  |
|-----------------|---------------------|--|
| optional match  | Partially supported | optional match (n)-->(m) where id(n)='1'<br>return m   |
| return          | Supported           | return [1,2,3] as p  |
| with            | Supported           | match (n) with labels(n) as label, count(*) as count<br>where count > 10 return *  |
| where           | Supported           | match (n:movie) where n.movieid > 10 return n  |
| order by        | Supported           | match (n:movie) return n order by n.genres   |
| skip            | Supported           | match (n:movie) return n order by n.genres<br>skip 5   |
| limit           | Supported           | match (n:movie) return n order by n.genres<br>skip 5 limit 10  |
| create          | Supported           | create (n:user{_ID_: 'Jack' }) return n  |
| delete          | Supported           | match (n:movie)<-[r]-(m:user) delete r   |
| set             | Supported           | match (n:user{userid:0}) set n.gender='M'<br>return n  |
| call procedures | Supported           | call db.schema()   |
| unwind          | Supported           | unwind [1, 2, 3] as p return p   |
| union           | Supported           | match (n:movie) return id(n) union match<br>(n:user) return id(n)<br><b>NOTE</b><br>Union is available for graphs smaller than 10 billion<br>edges only. |

 **NOTE**

1. Currently, merge and foreach operations are not supported. Cypher statements cannot add or delete indexes.
  2. GES metadata is not schema-free, and the vertex and edge label properties are strictly restricted. Therefore, the remove operation is not supported.
  3. The order by clause does not support sorting of the list type. When Cardinality of the property value is not single, the sorting result is unknown.
- Available items for the match clause

| Item | Description | Example Clauses | Earliest Version Required |
|------|-------------|-----------------|---------------------------|
|      |             |                 |                           |

|                              |  |   |        |
|------------------------------|--|---|--------|
| Vertex pattern               | Patterns for matching vertex with specified labels, properties, and IDs.   | match (n:movie{title:'hello'})<br>match (n) where id(n)='xx'  | 2.2.16 |
| Edge pattern                 | Patterns for matching directional and non-directional edges with specified labels and properties. Specified IDs of both start and end vertices are supported.                            | match (n)-[r] -> (m)<br>match (n)-[r]- (m)<br>match (n)-[r:rate{Rating:1}] - (m)<br>match (n)-[r]- (m) where id(n)='x'and id(m)='y' | 2.2.16 |
| Path                         | Anonymous paths  | match (n)-[r]->(m)-->(s)  | 2.2.16 |
|                              | Named paths  | match p=(n)-[r]->(m)-->(s)  | 2.2.19 |
| Multiple patterns            | You can enter multiple patterns after <b>match</b> and separate them with commas (,).<br>match (n)-[r]->(m), (m)-->(s)   |   | 2.2.16 |
| Multi-match                  | You can enter multiple match clauses. You can use <b>with</b> to connect multiple clauses.<br>match (n)-[r]->(m) with m match (m)-->(s)  |   | 2.2.16 |
| Variable-length path pattern | Patterns for matching variable-length paths starting with a specified vertex.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' return p<br>match p=(n{title:'name'})-[r*1..3]->(m) return p |   | 2.2.19 |
|                              | Traversal conditions for matching variable-length paths.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx'and all (x in nodes(p) where x.prop='value1') return p                             |   | 2.2.28 |
|                              | Both start vertex and end vertex of a variable-length path can be specified.<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='y' return p  |   | 2.3.9  |
|                              | Deduplication by end vertex is not supported:<br>match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='yy' return distinct m   |   | No     |

## 2. Parameterized queries

Cypher supports parameterized queries. Numeric and string values in a query statement are extracted and converted to parameters for faster compilation, improving the query speed.

There are some examples of parameterized queries:

### - Example 1

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user) where n.occupation = $occupation return n",
    "parameters": {
      "occupation" : "artist"
    },
    "resultDataContents": ["row"]
  }]
}
```

### - Example 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user {`Zip-code`:'98133'}) set n = $props return n",
    "parameters": {
      "props": {
        "gender": "M",
        "age": "56+"
      }
    },
    "resultDataContents": ["row"]
  }]
}
```

### NOTE

There are some scenarios where parameterized queries are not supported. The following syntax is not valid:

1. Using **\$param** to search by property key and value. For example, **match (n) where n.\$param = 'something'**
2. Using **\$code** for vertex and edge labels. For example, **match (n:user) set n:\$code**

## 3. Supported data types

Currently, GES supports 10 data types: char, char\_array, float, double, Boolean, long, Integer, date, enum, and string. Both Boolean and numeric types are supported in the Cypher syntax. The following table lists the mapping between other types and Cypher data types.

**Table 5-534** Mapping between data types of GES and Cypher

| GES        | Cypher | Description |
|------------|--------|-------------|
| char       | String | -           |
| char_array | String | -           |
| string     | String | -           |

| GES  | Cypher   | Description   |
|------|----------|---|
| date | Temporal | Currently, Cypher dates can be converted into GES dates, but Cypher date functions cannot be used for inputting a date. |

**Table 5-535** Special types supported by Cypher

| Type              | Supported | Example   |
|-------------------|-----------|---|
| Node              | Yes       | match (n) return n limit 10                         |
| Relations<br>hip  | Yes       | match (n)-[r]->(m) return r limit 10                |
| List              | Yes       | return [1,2,3] as li                                |
| Map               | Yes       | match (n)-->(m) return {start:id(n), end:id(m)}     |
| Path              | Yes       | match p=(n1)-[:friends*1..2]-(n2) return p limit 10 |
| Point,<br>Spatial | No        | -   |

 **NOTE**

For the special types listed above, only the List type can be used to match multi-value properties in GES. Other types cannot be used in a set statement for setting the value of a property.

4. Vertex ID compatibility

- Cypher does not provide the syntax for setting the ID when a vertex is added. In GES, however, an ID of the string type is required to uniquely identify a vertex. To use the Cypher syntax in GES, add `_ID_` to specify the ID of a vertex in the create statement. For example, the **`create(n{ _ID_:'123456'})`** statement creates a vertex whose ID is 123456.
- If the ID is not specified, a random ID is generated for the vertex.

 **NOTE**

The `_ID_` identifier is supported only in the create statement. The match and set clauses do not support the `_ID_` identifier. In the match clause, you can use the `id()` function to obtain the vertex ID.

- Creating a multi-label vertex

GES graphs of the database edition supports the creation of vertices with multiple labels by running Cypher statements, for example, **`create (n:user:student {userid:10, studentName:'Bob'})`**.

The attributes of all labels are enclosed in a single set of curly braces. The creation process automatically matches labels with their corresponding attributes based on the schema, and the order of the attributes has no impact on their correspondence with the labels.



### 5.2.13.3 Supported Expressions, Functions, and Procedures

#### Expression

Cypher queries support multiple expressions and can be used in combination to form various filter criteria. Currently, the following expressions are supported:

| Operation Type         | Expression   | Example  |
|------------------------|--|--|
| Logical operations     | and  | match (n:user) where n.age='Under 18' and n.gender='F' return n  |
|                        | or   | match(n:user) where n.`Zip-code`='22181' or n.userid=6 return n  |
|                        | not  | match(n:movie) where not n.genres contains 'Drama' return n  |
| Null value judgment    | is null  | match (n) where n.userid is null return n  |
|                        | is not null  | match (n) where n.userid is not null return n  |
| Comparison calculation | >, >=, <, <=, =, <>                                      | match(n:user) where n.userid >= 5 return n   |
| Arithmetic operations  | +, -, *, /, %, ^   | return (1+3)%3   |
| String comparisons     | starts with  | match(n:movie) where n.genres starts with 'Comedy' return n  |
|                        | ends with  | match(n:movie) where n.genres ends with 'Drama' return n   |
|                        | contains   | match(n:movie) where n.genres contains 'Drama' return n  |
| List-related operation | in   | match(n:student) where 'math' in n.courses return n  |
|                        | []   | match(n:user) return n['userid']<br>with [1, 2, 3, 4] as list return list[0]<br>with [1, 2, 3, 4] as list return list[0..1]<br>match p=(n)-->(m) return [x in nodes(p) where x.gender='F' id(x)]       |
| Date expressions       | .year, .month, .day, .hour, .minute, .second, .dayOfWeek | Year, month, and day of a specific date: with '2000-12-27 23:44:41' as strVal with datetime(strVal) as d2 return d2.year, d2.month, d2.day, d2.hour, d2.minute, d2.second, d2.dayOfWeek, d2.ordinalDay |

 **NOTE**

The where clause in Cypher queries does not support regular expressions.

## Function

Cypher supports the following functions for grouping, aggregation, and vertex and edge operations:

1. Aggregate

| Function | Earliest Version Supported | Description                          | Example   |
|----------|----------------------------|--------------------------------------|---|
| count    | 2.2.17                     | Returns the total number of results. | match (n) return count(*)<br>match (n) return count(n.userid) |
| collect  | 2.2.17                     | Collects results into a list.        | match (n:movie) return n.genres, collect(n) as movieList      |
| sum      | 2.3.3                      | Returns the sum of values.           | unwind [1, 2.0, 3] as p return sum(p)                         |
| avg      | 2.3.3                      | Returns the average of values.       | unwind [1, 2.0, 3] as p return avg(p)                         |
| min      | 2.3.3                      | Returns the minimum value.           | unwind [1, 2.0, 3] as p return min(p)                         |
| max      | 2.3.3                      | Returns the maximum value.           | unwind [1, 2.0, 3] as p return max(p)                         |

2. Regular functions

Based on the types of input parameters, regular functions are classified into vertex and edge functions, path functions, list functions, and value functions.

**Table 5-536** Vertex and edge functions

| Function | Earliest Version Supported | Description                 | Example                    |
|----------|----------------------------|-----------------------------|----------------------------|
| id       | 2.2.16                     | Obtains the ID of a vertex. | match (n) return id(n)     |
| labels   | 2.2.16                     | Obtains labels of a vertex. | match (n) return labels(n) |

| Function  | Earliest Version Supported | Description                          | Example   |
|-----------|----------------------------|--------------------------------------|---|
| type      | 2.2.16                     | Obtains the label of an edge.        | match(n)-[r]->(m) return type(r)                |
| degree    | 2.2.26                     | Obtains the degree of a vertex.      | match (n) where id='Vivian' return degree(n)    |
| inDegree  | 2.2.26                     | Obtains the indegree of a vertex.    | match (n) where id='Vivian' return inDegree(n)  |
| outDegree | 2.2.26                     | Obtains the outdegree of a vertex.   | match (n) where id='Vivian' return outDegree(n) |
| startNode | 2.3.10                     | Obtains the start vertex of an edge. | match (n)-[r]->(m) return startNode(r)          |
| endNode   | 2.3.10                     | Obtains the end vertex of an edge.   | match (n)-[r]->(m) return endNode(r)            |

**Table 5-537** Path functions (2.2.19)

| Function | Earliest Version Supported | Description              | Example   |
|----------|----------------------------|--------------------------|---|
| length   | 2.2.19                     | Obtains the path length. | match p=(n)-[:friends*1..2]->(m) return length(p) |

**Table 5-538** List functions

| Function | Earliest Version Supported | Description                          | Example                                  |
|----------|----------------------------|--------------------------------------|--|
| head     | 2.3.10                     | Obtains the first element of a list. | with [1,2,3,4] as list return head(list) |

| Function       | Earliest Version Supported | Description                         | Example                                  |
|----------------|----------------------------|-------------------------------------|--|
| last           | 2.3.10                     | Obtains the last element of a list. | with [1,2,3,4] as list return last(list) |
| size           | 2.3.10                     | Obtains the list length.            | with [1,2,3,4] as list return size(list) |
| range(2.3.10 ) | 2.3.10                     | Generates a list.                   | return range(1,5), range(1,5,2)          |

**Table 5-539** Value functions

| Function  | Earliest Version Supported | Description                               | Example  |
|-----------|----------------------------|---|--|
| toString  | 2.2.21                     | Converts a value to a string.             | match (n) where toString(labels(n)) contains 'movi' return n |
| toUpper   | 2.2.26                     | Converts a string into uppercase letters. | match (n:movie) return toUpper(n.title)                      |
| toLower   | 2.2.26                     | Converts a string into lowercase letters. | match (n:movie) return toLower(n.title)                      |
| toInteger | 2.2.29                     | Converts a string to an int number.       | with '123' as p return toInteger(p)                          |
| toLong    | 2.2.29                     | Converts a string to a long number.       | with '123' as p return toLong(p)                             |
| toFloat   | 2.2.29                     | Converts a string to a float number.      | with '123.4' as p return toFloat(p)                          |
| toDouble  | 2.2.29                     | Converts a string to a double number.     | with '123.4' as p return toDouble(p)                         |

|           |        |  |  |
|-----------|--------|--|--|
| toBoolean | 2.2.29 | Converts a string to a bool value.                           | with 'true' as p return toBoolean(p)                   |
| size      | 2.2.29 | Obtains the string length.                                   | with 'GES' as p return size(p)                         |
| subString | 2.3.10 | Truncates a part of a string.                                | return subString('abc', 1),<br>subString('abcde', 1,2) |
| coalesce  | 2.3.10 | Obtains the first non-null value of the parameters.          | return coalesce(null, '123')                           |
| trim      | 2.3.11 | Removes whitespace characters on both sides of a string.     | return trim(' hello ')                                 |
| lTrim     | 2.3.11 | Removes whitespace characters on the left side of a string.  | return lTrim(' hello')                                 |
| rTrim     | 2.3.11 | Removes whitespace characters on the right side of a string. | return rTrim('hello ')                                 |
| reverse   | 2.3.11 | Returns a string with the characters in reverse order.       | return reverse('hello')                                |
| left      | 2.3.11 | Obtains several characters from the left side of a string.   | with 'hello' as p return left(p, 3)                    |
| right     | 2.3.11 | Obtains several characters from the right side of a string.  | with 'hello' as p return right(p, 3)                   |
| replace   | 2.3.11 | Replaces a string.   | with 'hello' as p return replace(p, 'll', 'o')         |

|       |        |                  |  |
|-------|--------|------------------|--|
| split | 2.3.11 | Splits a string. | with 'hello' as p return split(p, 'e') |
|-------|--------|------------------|--|

**Table 5-540** Mathematical functions

| Function | Earliest Version Supported | Description                                  | Example                       |
|----------|----------------------------|--|-------------------------------|
| floor    | 2.3.10                     | Rounds a number down to the nearest integer. | return floor(4.1)             |
| ceil     | 2.3.10                     | Rounds a number up to the nearest integer.   | return ceil(4.1)              |
| round    | 2.3.14                     | Round  | return round(3.4), round(3.5) |
| abs      | 2.3.14                     | Absolute value function                      | return abs(-3),abs(-3.5)      |
| sin      | 2.3.14                     | Sine function                                | return sin(pi()/2)            |
| cos      | 2.3.14                     | Cosine function                              | return cos(0),cos(pi()/2)     |
| tan      | 2.3.14                     | Tangent function                             | return tan(pi()/4)            |
| acos     | 2.3.14                     | Inverse cosine function                      | return acos(1)                |
| asin     | 2.3.14                     | Inverse sine function                        | return asin(0)                |
| atan     | 2.3.14                     | Inverse tangent function                     | return atan(1)                |
| cot      | 2.3.14                     | Cotangent function                           | return cot(pi()/4)            |
| radians  | 2.3.14                     | Converts degree to radian.                   | return radians(180)           |
| degrees  | 2.3.14                     | Converts radian to degree.                   | return degrees(pi())          |

| Function | Earliest Version Supported | Description                                    | Example     |
|----------|----------------------------|--|-------------|
| pi       | 2.3.14                     | Returns the approximate value of Pi ( $\pi$ ). | return pi() |

**Table 5-541** Date and time functions

| Function       | Earliest Version Supported | Description   | Example                                 |
|----------------|----------------------------|---|---|
| datetime(val)  | 2.3.10                     | Returns the time based on the timestamp.                        | return datetime(1688696395)             |
| datetime()     | 2.3.14                     | Obtains the current time (valid only for read statements).      | return datetime()                       |
| timestamp(val) | 2.3.10                     | Returns the timestamp based on the time string.                 | return timestamp('2023-07-07 02:20:42') |
| timestamp()    | 2.3.14                     | Obtains the current timestamp (valid only for read statements). | return timestamp()                      |
| localDatetime  | 2.3.14                     | Converts a time or timestamp to a local time string.            | return localDatetime(timestamp())       |

**Table 5-542** Predicate functions

| Function | Earliest Version Supported | Description  | Example                   |
|----------|----------------------------|--|---------------------------|
| all      | 2.2.19                     | If all elements meet the expression, <b>true</b> is returned.        | all (x in p where x>1)    |
| any      | 2.2.19                     | If any element meets the expression, <b>true</b> is returned.        | any (x in p where x>1)    |
| none     | 2.2.19                     | If all elements cannot meet the expression, <b>true</b> is returned. | none (x in p where x>1)   |
| single   | 2.2.19                     | If only one element meets the expression, <b>true</b> is returned.   | single (x in p where x>1) |

**Table 5-543** Algorithm expressions

| Function         | Earliest Version Supported | Description                                      | Example   |
|------------------|----------------------------|--|---|
| shortestPath     | 2.3.2                      | Returns the shortest path between two vertices.  | The following statement returns the shortest path between the given vertices <b>n</b> and <b>m</b> . The direction is m to n, and the edge label is <b>rate</b> :<br>with n,m, shortestPath((n)-[:rate*]-(m)) as p return p |
| allShortestPaths | 2.3.2                      | Returns all shortest paths between two vertices. | The following statement returns all shortest paths between the given vertices <b>n</b> and <b>m</b> :<br>with n,m, allShortestPaths((n)-[*]-(m)) as p return p  |



 **NOTE**

- Aggregate functions, such as **sum()**, **avg()**, **max()**, and **min()**, are not available for database edition graphs.
- Degree functions, path functions, and algorithm expressions are not available for 10-billion-edge graphs and database edition graphs. The **toUpper** and **toLower** functions are not available for the database edition.

## Procedure

Currently, GES supports the following procedures.

| Procedure  | Statement   |
|--|---|
| Obtaining graph pattern information                                      | call db.schema()  |
| Obtaining vertex labels  | call db.labels()  |
| Querying the Cypher statements that are being executed                   | call dbms.listQueries()                                 |
| Terminating a Cypher statement based on <b>queryId</b>                   | call dbms.killQuery('queryId')                          |
| Querying indexes   | call db.indexes()                                       |
| Full-text indexing for querying vertices that meet the search conditions | call db.index.fulltext.queryNodes()                     |
| Full-text indexing for querying edges that meet the conditions           | call db.index.fulltext.queryRelationships()             |
| Merging nodes  | call apoc.refactor.mergeNodes(nodeList, refactorConfig) |

 **NOTE**

- Full-text indexes support six types of queries: prefix, wildcard, regexp, fuzzy, match, and combine. To use full-text indexes, you need to call the API for creating a full-text index.
- Function and procedure names are case sensitive and must be in lower camel case.

- **Example of a full-text index query request**

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
  "statements": [
    {
      "statement": "call db.index.fulltext.queryNodes('combine', {title:'1977'}) yield node, score return node, score skip 1 limit 10",
      "resultDataContents": [
        "row"
      ],
      "parameters": {}
    }
  ]
}
```

- Parallel edge processing policy  
When using Cypher to add edges, you can add duplicate edges. Duplicate edges are two edges with the same source vertex and target vertex.
- How to add an edge without a label  
When you use a Cypher statement to add an edge, set the label of the edge to the default value `__DEFAULT__`. For example, `create ()-[r:__DEFAULT__]->() return r`.

## Querying the Schema Structure Using Cypher

- Function  
You can call the `db.schema ()` function using Cypher to query the structure of a generated schema (obtained from OBS).
- Query statement
  - Name: Schema structure query
  - Statement: `call db.schema()`
  - Note:  
If you did not call the API for generating the schema structure, the returned schema file contains all labels.  
If you have called the API for generating the schema structure, this API returns the labels as the vertices and the relationships between the labels as edges.

# 6 Application Examples

## 6.1 Analyzing Graphs Using HyG API Algorithms

GES offers a range of fundamental graph algorithms, graph analysis algorithms, and graph metric algorithms. These algorithms can be used for analyzing relationships within a graph.

### Prerequisites

The **HyG computing engine** slider was toggled on when creating a graph of the database edition.

**Figure 6-1** HyG computing engine



### Procedure

1. Create a HyG graph.
  - a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}`. *project\_id* indicates the project ID, and *graph\_name* indicates the name of a graph created in the graph database.
  - b. Add **X-Auth-Token** to the request header.
  - c. Specify the following parameters in the request body:

```
{
  "inEdge": true // Whether the graph contains the incoming edge
}
```
  - d. Check the response. The request is successful if the following response is displayed:

```
{
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

2. Synchronize data from the graph database to the HyG computing engine.
  - a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}/sync`. *project\_id* indicates the project ID, and *graph\_name* indicates the graph name.
  - b. Add **X-Auth-Token** to the request header.
  - c. Specify the following parameters in the request body:

```
{
  "vertex": [], // Vertex property list
  "edge": [
    {
      "property": [
        "Rating"
      ],
      "label": "rate"
    }
  ] // Edge property list
}
```

- d. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

With this API, you can synchronize any changes made to the graph database, whether it be adding, deleting, or modifying data, to the HyG computing engine. During the initial data synchronization, the **vertex** and **edge** parameters specified in the request body will be applied. For subsequent synchronizations, these parameters will default to the values specified during the first synchronization.

3. Check details about the HyG graph.
  - a. Send **GET** `/ges/v1.0/{project_id}/hyg/{graph_name}/summary`. *project\_id* indicates the project ID, and *graph\_name* indicates the graph name.
  - b. Add **X-Auth-Token** to the request header.
  - c. Check the response. The request is successful if the following response is displayed:

```
{
  "data": {
    "inEdge": true,
    "idIndex": true,
    "policy": "oec",
    "updateTime": "2024-01-25 10:55:31",
    "vertex": [],
    "edge": [
      {
        "label": "rate",
        "property": [
          "Rating"
        ]
      }
    ],
    "vertexNum": 146,
    "edgeNum": 1659
  },
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

4. Execute the algorithm.

- a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm`. *project\_id* indicates the project ID, and *graph\_name* indicates the graph name.
- b. Add **X-Auth-Token** to the request header.
- c. Specify the following parameters in the request body (using PageRank as an example):

```
{
  "algorithmName": "pagerank",
  "parameters": {
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "directed": true
  }
}
```

- d. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "data": {
    "outputs": {
      "data_offset": 0,
      "data_return_size": 147,
      "data_total_size": 147,
      "pagerank": [
        {
          "38": 0.02115960730038959
        },
        {
          "13": 0.018535705068819635
        },
        {
          "7": 0.0166381431701182
        },
        ... ..
      ],
      "runtime": 0.022
    }
  },
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

5. Delete the HyG graph.

- a. Send **DELETE** `/ges/v1.0/{project_id}/hyg/{graph_name}`. *project\_id* indicates the project ID, and *graph\_name* indicates the graph name.
- b. Add **X-Auth-Token** to the request header.
- c. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

# 7 Monitoring Metrics

## Function

This chapter describes information on the metrics reported by GES, including their namespaces, lists, and dimensions. You can use the provided APIs to query the information.

## Namespace

SYS.GES

## Metrics

**Table 7-1** GES metrics

| Metric ID                  | Name                  | Description  | Value Range                      | Monitored Object |
|----------------------------|-----------------------|--|----------------------------------|------------------|
| ges001_vertex_util         | Vertex Capacity Usage | Capacity usage of vertices in a graph instance. The value is the ratio of the number of used vertices to the total vertex capacity.<br>Unit: % | 0 to 100<br>Value type: Float    | GES instance     |
| ges002_edge_util           | Edge Capacity Usage   | Capacity usage of edges in a graph instance. The value is the ratio of the number of used edges to the total edge capacity.<br>Unit: %         | 0 to 100<br>Value type: Float    | GES instance     |
| ges003_average_import_rate | Average Import Rate   | Average rate of importing vertices or edges to a graph instance<br>Unit: count/s   | 0 to 400000<br>Value type: Float | GES instance     |

| Metric ID                              | Name   | Description  | Value Range                | Monitored Object |
|--|--|--|----------------------------|------------------|
| ges004_request_count                   | Request Quantity                             | Number of requests received by a graph instance<br>Unit: count   | ≥ 0<br>Value type: Integer | GES instance     |
| ges005_average_response_time           | Average Response Time                        | Average response time of requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer | GES instance     |
| ges006_min_response_time               | Minimum Response Time                        | Minimum response time of requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer | GES instance     |
| ges007_max_response_time               | Maximum Response Time                        | Maximum response time of requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer | GES instance     |
| ges008_read_task_pending_queue_size    | Length of the Waiting Queue for Read Tasks   | Length of the waiting queue for read requests received by a graph instance. This metric is used to view the number of read requests waiting in the queue.<br>Unit: count | ≥ 0<br>Value type: Integer | GES instance     |
| ges009_read_task_pending_max_time      | Maximum Waiting Duration of Read Tasks       | Maximum waiting duration of read requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer | GES instance     |
| ges010_pending_max_time_read_task_type | Type of the Read Task That Waits the Longest | Type of the read request that waits the longest in a graph instance. You can find the corresponding task name in GES documents.  | ≥ 1<br>Value type: Integer | GES instance     |
| ges011_read_task_running_queue_size    | Length of the Running Queue for Read Tasks   | Length of the running queue for read requests received by a graph instance. This metric is used to view the number of running read requests.<br>Unit: count              | ≥ 0<br>Value type: Integer | GES instance     |

| Metric ID                               | Name  | Description  | Value Range                | Monitored Object |
|---|---|--|----------------------------|------------------|
| ges012_read_task_running_max_time       | Maximum Running Duration of Read Tasks        | Maximum running duration of read requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer | GES instance     |
| ges013_running_max_time_read_task_type  | Type of the Read Task That Runs the Longest   | Type of the read request that runs the longest in a graph instance. You can find the corresponding task name in GES documents.   | ≥ 1<br>Value type: Integer | GES instance     |
| ges014_write_task_pending_queue_size    | Length of the Waiting Queue for Write Tasks   | Length of the waiting queue for write requests received by a graph instance. This metric is used to view the number of write requests waiting in the queue.<br>Unit: count | ≥ 0<br>Value type: Integer | GES instance     |
| ges015_write_task_pending_max_time      | Maximum Waiting Duration of Write Tasks       | Maximum waiting duration of write requests received by a graph instance<br>Unit: ms  | ≥ 0<br>Value type: Integer | GES instance     |
| ges016_pending_max_time_write_task_type | Type of the Write Task That Waits the Longest | Type of the write request that waits the longest in a graph instance. You can find the corresponding task name in GES documents.   | ≥ 1<br>Value type: Integer | GES instance     |
| ges017_write_task_running_queue_size    | Length of the Running Queue for Write Tasks   | Length of the running queue for write requests received by a graph instance. This metric is used to view the number of running write requests.<br>Unit: count              | ≥ 0<br>Value type: Integer | GES instance     |



| Metric ID                               | Name   | Description   | Value Range                   | Monitored Object |
|---|--|---|-------------------------------|------------------|
| ges018_write_task_running_max_time      | Maximum Running Duration of Write Tasks      | Maximum running duration of write requests received by a graph instance<br>Unit: ms   | ≥ 0<br>Value type: Integer    | GES instance     |
| ges019_running_max_time_write_task_type | Type of the Write Task That Runs the Longest | Type of the write request that runs the longest in a graph instance. You can find the corresponding task name in GES documents. | ≥ 1<br>Value type: Integer    | GES instance     |
| ges020_computer_resource_usage          | Computing Resource Usage                     | Computing resource usage of each graph instance<br>Unit: %  | 0 to 100<br>Value type: Float | GES instance     |
| ges021_memory_usage                     | Memory Usage                                 | Memory usage of each graph instance<br>Unit: %  | 0 to 100<br>Value type: Float | GES instance     |
| ges022_iops                             | IOPS   | Number of I/O requests processed by each graph instance per second<br>Unit: count/s   | ≥ 0<br>Value type: Integer    | GES instance     |
| ges023_bytes_in                         | Network Input Throughput                     | Data input to each graph instance per second over the network<br>Unit: byte/s   | ≥ 0<br>Value type: Float      | GES instance     |
| ges024_bytes_out                        | Network Output Throughput                    | Data sent to the network per second from each graph instance<br>Unit: byte/s  | ≥ 0<br>Value type: Float      | GES instance     |
| ges025_disk_usage                       | Disk Usage                                   | Disk usage of each graph instance<br>Unit: %  | 0 to 100<br>Value type: Float | GES instance     |
| ges026_disk_total_size                  | Total Disk Size                              | Total data disk space of each graph instance<br>Unit: GB  | ≥ 0<br>Value type: Float      | GES instance     |

| Metric ID                     | Name                        | Description  | Value Range                | Monitored Object |
|-------------------------------|-----------------------------|--|----------------------------|------------------|
| ges027_disk_used_size         | Disk Space Used             | Used data disk space of each graph instance<br>Unit: GB  | ≥ 0<br>Value type: Float   | GES instance     |
| ges028_disk_read_throughput   | Disk Read Throughput        | Data volume read from the disk in a graph instance per second<br>Unit: byte/s                    | ≥ 0<br>Value type: Float   | GES instance     |
| ges029_disk_write_throughput  | Disk Write Throughput       | Data volume written to the disk in a graph instance per second<br>Unit: byte/s                   | ≥ 0<br>Value type: Float   | GES instance     |
| ges030_avg_disk_sec_per_read  | Average Time per Disk Read  | Average time used each time when the disk of a graph instance reads data<br>Unit: second         | ≥ 0<br>Value type: Float   | GES instance     |
| ges031_avg_disk_sec_per_write | Average Time per Disk Write | Average time used each time when data is written to the disk of a graph instance<br>Unit: second | ≥ 0<br>Value type: Float   | GES instance     |
| ges032_avg_disk_queue_length  | Average Disk Queue Length   | Average I/O queue length of the disk in a graph instance<br>Unit: count                          | ≥ 0<br>Value type: Integer | GES instance     |

## Dimensions

Table 7-2 Dimensions

| Key         | Value        |
|-------------|--------------|
| instance_id | GES instance |

## Mapping Between Task Types and Names

**Table 7-3** Task types and corresponding task names

| Type | Name                                       |
|------|--|
| 100  | Querying a vertex                          |
| 101  | Creating a vertex                          |
| 102  | Deleting a vertex                          |
| 103  | Modifying a vertex property                |
| 104  | Adding a vertex label                      |
| 105  | Deleting a vertex label                    |
| 200  | Querying an edge                           |
| 201  | Creating an edge                           |
| 202  | Deleting an edge                           |
| 203  | Modifying an edge property                 |
| 300  | Querying schema details                    |
| 301  | Adding a Label                             |
| 302  | Modifying a Label                          |
| 303  | Querying a Label                           |
| 304  | Modifying a property                       |
| 400  | Querying graph details                     |
| 401  | Clearing a graph                           |
| 402  | Incrementally importing graph data online  |
| 403  | Creating a graph                           |
| 405  | Deleting a graph                           |
| 406  | Exporting a graph                          |
| 407  | filtered_khop                              |
| 408  | Querying path details                      |
| 409  | Incrementally importing graph data offline |
| 500  | Creating a graph backup                    |
| 501  | Restoring a graph from a backup            |
| 601  | Creating an index.                         |

| Type | Name                          |
|------|-------------------------------|
| 602  | Querying an index             |
| 603  | Updating an index             |
| 604  | Deleting an index             |
| 700  | Running the algorithm         |
| 800  | Querying an asynchronous task |

# 8 Out-of-Date APIs

---

## 8.1 Management Plane APIs (V1)

### 8.1.1 System Management APIs

#### 8.1.1.1 Querying Quotas

##### Function

This API is used to query tenant quotas.

##### URI

GET /v1.0/{project\_id}/graphs/quotas

**Table 8-1** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 8-2** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-3** Response body parameter

| Parameter    | Type                                | Description  |
|--------------|-------------------------------------|--|
| errorMessage | String                              | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String                              | System prompt.<br><ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| quotas       | <a href="#">GesQuotaResp</a> object | Resource type list. This field is left blank when the request fails.   |

**Table 8-4** GesQuotaResp

| Parameter | Type                                   | Description             |
|-----------|--|-------------------------|
| resources | Array of <a href="#">Quota</a> objects | GES resource quota list |

**Table 8-5** Quota

| Parameter  | Type    | Description  |
|------------|---------|--|
| type       | String  | Quota type. Available values are as follows: <ul style="list-style-type: none"> <li>• "graph"</li> <li>• "backup"</li> <li>• "metadata"</li> </ul> |
| available  | Integer | Number of available graphs   |
| edgeVolume | Integer | Number of available edges. This parameter is available only when <b>type</b> is <b>graph</b> .   |

### Example Request

GET https://Endpoint/v1.0/{project\_id}/graphs/quotas

### Example Response

Status code: 200

OK

```
{
  "quotas" : {
    "resources" : [ {
      "type" : "graph",
      "available" : 1,
      "edgeVolume" : 178800
    }, {
      "type" : "backup",
      "available" : 7
    }, {
      "type" : "metadata",
      "available" : 13
    }
  ]
}
```

### Status Codes

| Status Code | Description               |
|-------------|---------------------------|
| 200         | Request sent.             |
| 400         | Request error.            |
| 401         | Authorization failed.     |
| 403         | No operation permissions. |
| 404         | No resources found.       |
| 500         | Internal server error.    |
| 503         | Service unavailable.      |

## 8.1.2 Graph Management APIs

### 8.1.2.1 Querying the Graph List

#### Function

This API is used to query all graphs owned by the current tenant.

#### URI

GET /v1.0/{project\_id}/graphs

**Table 8-6** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 8-7** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |

#### Request Parameters

**Table 8-8** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |



## Response Parameters

Status code: 200

**Table 8-9** Response body parameter

| Parameter    | Type                                     | Description  |
|--------------|--|--|
| graphCount   | Integer                                  | Total number of graphs. This parameter is left blank when the request fails.   |
| graphs       | Array of <a href="#">graph_1</a> objects | Graph list. This parameter is left blank when the request fails.   |
| errorMessage | String                                   | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message. |
| errorCode    | String                                   | System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.    |

**Table 8-10** graph\_1

| Parameter           | Type   | Description                            |
|---------------------|--|--|
| id                  | String   | Graph ID                               |
| name                | String   | Graph name                             |
| createdBy           | String   | IAM username                           |
| isMultiAz           | String   | Whether to enable cross-AZ HA          |
| regionCode          | String   | Region code                            |
| azCode              | String   | AZ code                                |
| schemaPath          | Array of <a href="#">schemaPath_1</a> objects  | Path for storing the metadata file     |
| edgesetPath         | Array of <a href="#">edgesetPath_1</a> objects | OBS path for storing the edge data set |
| edgesetFormat       | String   | Format of the edge data file           |
| edgesetDefaultLabel | String   | Default label of the edge data file    |

| Parameter             | Type   | Description   |
|-----------------------|--|---|
| vertexsetPath         | Array of <a href="#">vertexsetPath_1</a> objects | OBS path for storing the vertex data set  |
| vertexsetFormat       | String   | Format of the vertex data file  |
| vertexsetDefaultLabel | String   | Default label of the vertex data file   |
| dataStoreVersion      | String   | Graph version   |
| sys_tags              | Array of strings                                 | Enterprise project information. If this parameter is not specified, this function is disabled (default).  |
| status                | String   | Status code of a graph: <ul style="list-style-type: none"> <li>● <b>100</b>: Indicates that a graph is being prepared.</li> <li>● <b>200</b>: indicates that a graph is running.</li> <li>● <b>201</b>: indicates that a graph is upgrading.</li> <li>● <b>202</b>: indicates that a graph is being imported.</li> <li>● <b>203</b>: indicates that a graph is being rolled back.</li> <li>● <b>204</b>: indicates that a graph is being exported.</li> <li>● <b>205</b>: indicates that a graph is being cleared.</li> <li>● <b>206</b>: indicates that the system is preparing for resize.</li> <li>● <b>207</b>: indicates that the resize is in progress.</li> <li>● <b>208</b>: Indicates that the resize is being rolled back.</li> <li>● <b>210</b>: Preparing for expansion</li> <li>● <b>211</b>: Expanding</li> <li>● <b>300</b>: indicates that a graph is faulty.</li> <li>● <b>303</b>: indicates that a graph fails to be created.</li> <li>● <b>400</b>: indicates that a graph is deleted.</li> <li>● <b>900</b>: indicates that a graph is stopped.</li> <li>● <b>901</b>: indicates that a graph is being stopped.</li> <li>● <b>920</b>: indicates that a graph is being started.</li> </ul> |
| actionProgress        | String   | Progress of graph creation in percentage  |

| Parameter          | Type             | Description  |
|--------------------|------------------|--|
| graphSizeTypeIndex | String           | Graph size type index: <ul style="list-style-type: none"> <li>● <b>0</b>: indicates 10 thousand edges.</li> <li>● <b>1</b>: indicates 1 million edges.</li> <li>● <b>2</b>: indicates 10 million edges.</li> <li>● <b>3</b>: indicates 100 million edges.</li> <li>● <b>4</b>: indicates 1 billion edges.</li> <li>● <b>5</b>: indicates 10 billion edges.</li> <li>● <b>401</b>: indicates 1 billion enhanced edges.</li> </ul> |
| vpclId             | String           | VPC ID   |
| subnetId           | String           | Subnet ID in the VPC   |
| securityGroupId    | String           | Security group ID  |
| replication        | Integer          | Number of replicas. The default value is <b>1</b> .  |
| created            | String           | Time when a graph is created   |
| updated            | String           | Time when a graph is updated   |
| privateIp          | String           | Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.   |
| trafficIpList      | Array of strings | Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.  |
| cryptAlgorithm     | String           | Graph instance cryptography algorithm. Available values are as follows: <ul style="list-style-type: none"> <li>● <b>generalCipher</b>: Chinese cryptographic algorithm</li> <li>● <b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul>   |
| enableHttps        | Boolean          | Whether to enable the security mode. This mode may damage GES performance greatly.   |
| tags               | Array of objects | Tag list. Each tag is in <key,value> format.   |

**Table 8-11** schemaPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

**Table 8-12** edgesetPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

**Table 8-13** vertexsetPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

## Example Request

GET https://Endpoint/v1.0/{project\_id}/graphs?offset=0&limit=10

## Example Response

Status code: 200

OK

```
{
  "graphCount": 2,
  "graphs": [
    {
      "id": "f1529b88-c958-493e-8452-fccfe932cde1",
      "name": "demo",
      "regionCode": "XXX",
      "azCode": "XXX",
      "schemaPath": [
        {
          "path": "ges-graphs/demo_movie/schema.xml",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "edgesetPath": [
        {
          "path": "ges-graphs/demo_movie/edge.csv",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "vertexsetPath": [
        {
          "path": "",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "status": "200",
      "graphSizeTypeIndex": "1",
      "vpclId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
      "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
      "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
      "created": "2018-07-23T04:09:44",
      "updated": "2018-07-23T04:09:44",
      "privateIp": "192.168.0.4",

      "dataStoreVersion": "1.0.5",
      "arch": "x86_64",
    },
    {
      "id": "53205529-026b-455a-9e07-228fae4b12b9",
      "name": "ges_c5de",
      "regionCode": "XXX",
      "azCode": "XXX",

      "schemaPath": [
        {
          "path": "ges-graphs/demo_movie/schema.xml",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "edgesetPath": [
        {
          "path": "ges-graphs/demo_movie/edge.csv",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "vertexsetPath": [
        {
          "path": "",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "status": "200",
    }
  ]
}
```

```

    "graphSizeTypeIndex": "2",
    "vpcId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
    "created": "2018-07-18T13:30:16",
    "updated": "2018-07-18T13:30:16",
    "privateIp": "192.168.0.168",
    "dataStoreVersion": "1.0.5",
    "arch": "aarch64",
  }
]
}

```

## Status Codes

| Status Code | Description               |
|-------------|---------------------------|
| 200         | Request sent.             |
| 400         | Request error.            |
| 401         | Authorization failed.     |
| 403         | No operation permissions. |
| 404         | No resources found.       |
| 500         | Internal server error.    |
| 503         | Service unavailable.      |

### 8.1.2.2 Querying Graph Details

#### Function

This API is used to query the details about a graph based on the graph ID.

#### URI

GET /v1.0/{project\_id}/graphs/{graph\_id}

**Table 8-14** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 8-15** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-16** Response body parameter

| Parameter    | Type                           | Description   |
|--------------|--------------------------------|---|
| graph        | <a href="#">graph_1</a> object | Graph object. If the request fails, this parameter is left empty.   |
| errorMessage | String                         | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String                         | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |

**Table 8-17** graph\_1

| Parameter | Type   | Description  |
|-----------|--------|--------------|
| id        | String | Graph ID     |
| name      | String | Graph name   |
| createdBy | String | IAM username |

| Parameter             | Type   | Description  |
|-----------------------|--|--|
| isMultiAz             | String   | Whether to enable cross-AZ HA  |
| regionCode            | String   | Region code  |
| azCode                | String   | AZ code  |
| schemaPath            | Array of <a href="#">schemaPath_1</a> objects    | Path for storing the metadata file   |
| edgesetPath           | Array of <a href="#">edgesetPath_1</a> objects   | OBS path for storing the edge data set   |
| edgesetFormat         | String   | Format of the edge data file   |
| edgesetDefaultLabel   | String   | Default label of the edge data file  |
| vertexsetPath         | Array of <a href="#">vertexsetPath_1</a> objects | OBS path for storing the vertex data set   |
| vertexsetFormat       | String   | Format of the vertex data file   |
| vertexsetDefaultLabel | String   | Default label of the vertex data file  |
| dataStoreVersion      | String   | Graph version  |
| sys_tags              | Array of strings                                 | Enterprise project information. If this parameter is not specified, this function is disabled (default). |



| Parameter          | Type   | Description  |
|--------------------|--------|--|
| status             | String | <p>Status code of a graph:</p> <ul style="list-style-type: none"> <li>● <b>100</b>: Indicates that a graph is being prepared.</li> <li>● <b>200</b>: indicates that a graph is running.</li> <li>● <b>201</b>: indicates that a graph is upgrading.</li> <li>● <b>202</b>: indicates that a graph is being imported.</li> <li>● <b>203</b>: indicates that a graph is being rolled back.</li> <li>● <b>204</b>: indicates that a graph is being exported.</li> <li>● <b>205</b>: indicates that a graph is being cleared.</li> <li>● <b>206</b>: indicates that the system is preparing for resize.</li> <li>● <b>207</b>: indicates that the resize is in progress.</li> <li>● <b>208</b>: Indicates that the resize is being rolled back.</li> <li>● <b>210</b>: Preparing for expansion</li> <li>● <b>211</b>: Expanding</li> <li>● <b>300</b>: indicates that a graph is faulty.</li> <li>● <b>303</b>: indicates that a graph fails to be created.</li> <li>● <b>400</b>: indicates that a graph is deleted.</li> <li>● <b>900</b>: indicates that a graph is stopped.</li> <li>● <b>901</b>: indicates that a graph is being stopped.</li> <li>● <b>920</b>: indicates that a graph is being started.</li> </ul> |
| actionProgress     | String | Progress of graph creation in percentage   |
| graphSizeTypeIndex | String | <p>Graph size type index:</p> <ul style="list-style-type: none"> <li>● <b>0</b>: indicates 10 thousand edges.</li> <li>● <b>1</b>: indicates 1 million edges.</li> <li>● <b>2</b>: indicates 10 million edges.</li> <li>● <b>3</b>: indicates 100 million edges.</li> <li>● <b>4</b>: indicates 1 billion edges.</li> <li>● <b>5</b>: indicates 10 billion edges.</li> <li>● <b>401</b>: indicates 1 billion enhanced edges.</li> </ul>  |
| vpcId              | String | VPC ID   |
| subnetId           | String | Subnet ID in the VPC   |
| securityGroupId    | String | Security group ID  |

| Parameter      | Type             | Description  |
|----------------|------------------|--|
| replication    | Integer          | Number of replicas. The default value is <b>1</b> .  |
| created        | String           | Time when a graph is created   |
| updated        | String           | Time when a graph is updated   |
| privatelp      | String           | Private network access address of a graph instance. Users can access the instance using the IP address through the ECS deployed on the private network.  |
| trafficIpList  | Array of strings | Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.  |
| cryptAlgorithm | String           | Graph instance cryptography algorithm. Available values are as follows: <ul style="list-style-type: none"> <li>● <b>generalCipher</b>: Chinese cryptographic algorithm</li> <li>● <b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul> |
| enableHttps    | Boolean          | Whether to enable the security mode. This mode may damage GES performance greatly.   |
| tags           | Array of strings | Tag list. Each tag is in <key,value> format.   |

**Table 8-18** schemaPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

**Table 8-19** edgesetPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

**Table 8-20** vertexsetPath\_1

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | Job ID corresponding to OBS file import  |
| path      | String | OBS storage path, excluding OBS endpoint   |
| status    | String | OBS file import status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>partiallyFailed</b>: Partially failed.</li> <li>● <b>failed</b>: Failed to import the file.</li> </ul> |

## Example Request

GET https://Endpoint/v1.0/{project\_id}/graphs/{graph\_id}

## Example Response

Status code: 200

OK

```
{
  "graph": {
    "id": "f1529b88-c958-493e-8452-fccfe932cde1",
    "name": "demo",
    "regionCode": "XXX",
    "azCode": "XXX",
    "schemaPath": [
      {
        "path": "ges-graphs/demo_movie/schema.xml",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ],
    "edgesetPath": [
      {
        "path": "ges-graphs/demo_movie/edge.csv",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ]
  },
}
```

```

    "vertexsetPath": [
      {
        "path": "",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ],
    "status": "200",
    "graphSizeTypeIndex": "1",
    "vpclId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
    "created": "2018-07-23T04:09:44",
    "privateIp": "192.168.0.4",

    "dataStoreVersion": "1.0.5",
    "arch": "x86_64"
  }
}

```

## Status Codes

| Status Code | Description               |
|-------------|---------------------------|
| 200         | Request sent.             |
| 400         | Request error.            |
| 401         | Authorization failed.     |
| 403         | No operation permissions. |
| 404         | No resources found.       |
| 500         | Internal server error.    |
| 503         | Service unavailable.      |

### 8.1.2.3 Creating a Graph

#### Function

This API is used to create a graph.

#### URI

POST /v1.0/{project\_id}/graphs

**Table 8-21** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 8-22** Request header parameter

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-23** Request body parameter

| Parameter | Mandatory | Type                         | Description |
|-----------|-----------|------------------------------|-------------|
| graph     | Yes       | <a href="#">graph</a> object | Graph type  |

**Table 8-24** graph

| Parameter           | Mandatory | Type   | Description  |
|---------------------|-----------|--------|--|
| name                | Yes       | String | A graph name must start with a letter and contains 4 to 50 characters consisting of letters, digits, hyphens (-), and underscores (_). It cannot contain special characters.   |
| graphSizeType Index | Yes       | String | Graph size type index: <ul style="list-style-type: none"> <li>● <b>0</b>: indicates 10 thousand edges.</li> <li>● <b>1</b>: indicates 1 million edges.</li> <li>● <b>2</b>: indicates 10 million edges.</li> <li>● <b>3</b>: indicates 100 million edges.</li> <li>● <b>4</b>: indicates 1 billion edges.</li> <li>● <b>5</b>: indicates 10 billion edges.</li> <li>● <b>401</b>: indicates 1 billion enhanced edges.</li> </ul> |

| Parameter         | Mandatory | Type                               | Description  |
|-------------------|-----------|------------------------------------|--|
| arch              | No        | String                             | Graph instance's CPU architecture type. The value can be <b>x86_64</b> or <b>aarch64</b> . The default value is <b>x86_64</b> . <ul style="list-style-type: none"> <li>• <b>x86_64</b>: x86 64-bit architecture</li> <li>• <b>aarch64</b>: Arm 64-bit architecture</li> </ul>                    |
| vpclId            | Yes       | String                             | VPC ID   |
| subnetId          | Yes       | String                             | Subnet ID in the VPC   |
| securityGroupId   | Yes       | String                             | Security group ID  |
| publicIp          | No        | <b>publicIp</b> object             | Public IP address. If the parameter is not specified, public connection is not used by default.  |
| enableMultiAZ     | No        | Boolean                            | Whether the created graph supports the cross-AZ mode. The default value is <b>false</b> . If the value is <b>true</b> , the system will create the ECSs in the graph in two AZs.<br><br>If this parameter is not specified when you create a graph, all ECSs in the graph are created in one AZ. |
| encryption        | No        | <b>encryptionReq</b> object        | Whether to encrypt the graph instance. The graph instance is not encrypted by default.   |
| ltsOperationTrace | No        | object                             | Whether to enable audit logs. This function is disabled by default.  |
| sys_tags          | No        | Array of <b>SysTagsRes</b> objects | Enterprise project information. If this parameter is not specified, this function is disabled (default).   |
| tags              | No        | Array of <b>SysTagsRes</b> objects | TMS tags for expenses. This function is disabled by default.   |

| Parameter      | Mandatory | Type      | Description   |
|----------------|-----------|-----------|---|
| enableRBAC     | No        | Boolean   | Whether to enable granular permission control for the created graph. The default value is <b>false</b> , indicating that granular permission control is disabled. If this parameter is set to <b>true</b> , no user has the permission to access the graph. To access the graph, you need to call the granular permission control API of the service plane to set the required permissions. |
| cryptAlgorithm | Yes       | String    | Graph instance cryptography algorithm. Available values are as follows: <ul style="list-style-type: none"> <li>• <b>generalCipher</b>: Chinese cryptographic algorithm</li> <li>• <b>SMcompatible</b>: Commercial cryptography algorithm (compatible with international ones)</li> </ul>  |
| enableHttps    | Yes       | Boolean   | Whether to enable the security mode. This mode may damage GES performance greatly.  |
| tags           | No        | JsonArray | Tag list. Each tag is in <key,value> format.  |

**Table 8-25** parameters

| Parameter     | Mandatory | Type   | Description   |
|---------------|-----------|--------|---|
| schemaPath    | Yes       | String | OBS path for storing the metadata file. Only files are supported.   |
| edgesetPath   | Yes       | String | OBS path for storing the edge file. Only files are supported.   |
| edgesetFormat | No        | String | Format of the edge data set. Currently, only the CSV format is supported.<br>The CSV format is used by default. |

| Parameter             | Mandatory | Type                                | Description  |
|-----------------------|-----------|-------------------------------------|--|
| edgesetDefaultLabel   | No        | String                              | Default label of an edge data set. This parameter is left blank by default.  |
| vertexsetPath         | No        | String                              | OBS path for storing the vertex file. Only files are supported.  |
| vertexsetFormat       | No        | String                              | Format of the vertex data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.            |
| vertexsetDefaultLabel | No        | String                              | Default label of a vertex data set. This parameter is left blank by default.   |
| logDir                | No        | String                              | OBS log storage directory. This directory stores the data that fails to be imported during graph creation and detailed logs. |
| parallelEdge          | No        | <a href="#">parallelEdge</a> object | How to process repetitive edges.   |

 NOTE

- For details about the value validity of the **schemaPath**, **edgesetPath**, **vertexsetPath**, and **logDir** character strings, see the [OBS Object Name Restrictions](#).

**Table 8-26** parallelEdge

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| action    | No        | String | Processing mode of repetitive edges. The value can be <b>allow</b> , <b>ignore</b> , or <b>override</b> . The default value is <b>allow</b> . <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul> |



| Parameter   | Mandatory | Type    | Description  |
|-------------|-----------|---------|--|
| ignoreLabel | No        | Boolean | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: Indicates that the repetitive edge definition contains the label. That is, the &lt;source vertex, target vertex, label&gt; indicates an edge.</li> </ul> |

**Table 8-27** publicIp

| Parameter      | Mandatory | Type   | Description  |
|----------------|-----------|--------|--|
| publicBindType | No        | String | <p>Binding type of an EIP. The value can be either of the following:</p> <ul style="list-style-type: none"> <li>• <b>auto_assign</b></li> <li>• <b>bind_existing</b></li> </ul>  |
| eipId          | No        | String | <p>EIP ID</p> <ul style="list-style-type: none"> <li>• If <b>publicBindType</b> is <b>bind_existing</b>, the value is the ID of a created EIP that has not been bound.</li> <li>• If <b>publicBindType</b> is <b>auto_assign</b>, leave this parameter blank.</li> </ul> |

**Table 8-28** encryptionReq

| Parameter   | Mandatory | Type    | Description   |
|-------------|-----------|---------|---|
| enable      | No        | Boolean | <p>Whether to enable the encryption feature. The value can be <b>true</b> or <b>false</b>. The default value is <b>false</b>.</p> |
| masterKeyId | No        | String  | <p>ID of the customer master key created by DEW in the project corresponding to the graph creation</p>                            |

**Table 8-29** SysTagsRes

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| key       | No        | String | Key of the enterprise project. Set this parameter to <b>_sys_enterprise_project_id</b> . |
| value     | No        | String | Enterprise project ID. You can obtain it from the enterprise project.                    |

## Response Parameters

Status code: 200

**Table 8-30** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| id           | String | Graph ID  |
| name         | String | Graph name  |
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

```
POST https://Endpoint/v1.0/{project_id}/graphs
{
  "graph":{
    "name":"demo",
    "graphSizeTypeIndex": "1",
    "arch":"x86_64",
    "vpcId":"2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId":"dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId":"11d27338-8649-4076-8579-5ebc1a60f79e",
    "publicIp":{
      "publicBindType":"bind_existing",
      "eipId":"30ef2d58-08a9-4481-b526-b2cbe67d020d"
    },
    "enableMultiAz":false,
    "encryption":{
      "enable":true,
      "masterKeyId":"b00b9356-73fb-4d49-8f79-f0a5da5354d1"
    }
  }
}
```

```

    },
    "cryptAlgorithm": "generalCipher",
    "enableHttps": "false"
  }
}

```

## Example Responses

**Status code: 200**

OK

```

{
  "id" : "f1529b88-c958-493e-8452-fccfe932cde1",
  "name" : "demo"
}

```

## Status Codes

| Status Code | Description               |
|-------------|---------------------------|
| 200         | Request sent.             |
| 400         | Request error.            |
| 401         | Authorization failed.     |
| 403         | No operation permissions. |
| 404         | No resources found.       |
| 500         | Internal server error.    |
| 503         | Service unavailable.      |

### 8.1.2.4 Closing a Graph

#### Function

This API is used to stop a graph. After the graph is created, you can disable it if it is not used temporarily.

#### URI

POST [https://Endpoint/v1.0/{project\\_id}/graphs/{graph\\_id}/action?action\\_id=stop](https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=stop)

**Table 8-31** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-32** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>The value can be: <ul style="list-style-type: none"> <li>• <b>stop</b></li> </ul> |

## Request Parameters

**Table 8-33** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-34** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---|
| jobId     | String | ID of the graph stopping job. This parameter is left blank when the request fails.<br><br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |

## Example Request

POST https://Endpoint//v1.0/{project\_id}/graphs/{graph\_id}/action?action\_id=stop

## Example Response

**Status code: 200**

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Bad Request

```
{
  "errorCode" : "GES.7001",
  "errorMessage" : "The graph is not running."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

## 8.1.2.5 Starting a Graph

### Function

This API is used to start a graph. You can disable a graph if it is not used temporarily.

### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-35** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-36** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to the enumeration value <b>start</b> , which means the graph will be started. |

### Request Parameters

**Table 8-37** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-38** Request body parameters

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| graph_backup_id | No        | String | Backup ID associated during graph startup. If this parameter is configured, the graph starts from the backup. If this parameter is left blank, the graph starts from the status when it was closed last time. For details about how to back up a graph, see section <a href="#">Adding a Backup</a> . |

## Response Parameters

Status code: 200

**Table 8-39** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>  |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>   |
| jobId        | String | ID of the graph startup job. This parameter is left blank when the request fails. <p><b>NOTE</b><br/>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a>.</p> |

## Example Request

```
https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=start
{
  "graph_backup_id" : "08a898ae-3ff8-40e8-a7ed-03afe05aedbb"
}
```

## Example Response

**Status code: 200**

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Bad Request

```
{
  "errorCode" : "GES.7015",
  "errorMessage" : "The graph is not running or stopped."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.6 Deleting a Graph

#### Function

This API is used to delete a graph.

#### URI

DELETE /v1.0/{project\_id}/graphs/{graph\_id}

**Table 8-40** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |



**Table 8-41** Query parameters

| Parameter  | Mandatory | Type    | Description   |
|------------|-----------|---------|---|
| keepBackup | No        | Boolean | Whether to retain the backups of a graph after it is deleted. By default, one automatic backup and two manual backups are retained. If this parameter is left empty, no backups are retained. |

## Request Parameters

**Table 8-42** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-43** Response body parameter

| Parameter    | Type   | Description   |
|--------------|--------|---|
| jobId        | String | ID of the graph deletion job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |

| Parameter | Type   | Description  |
|-----------|--------|--|
| errorCode | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

### Example Request

```
DELETE https://Endpoint/v1.0/{project_id}/graphs/{graph_id}
```

### Example Response

Example response with status code **200**:

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Bad Request

```
{
  "errorCode" : "GES.7000",
  "errorMessage" : "The graph does not exist or has been deleted."
}
```

### Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

## 8.1.2.7 Incrementally Importing Data to a Graph

### Function

This API is used to import data to graphs incrementally.

 NOTE

To prevent failures in restoring the imported graph data during system restarting, do not delete the data stored on OBS when the graph is in use.

## URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-44** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-45** Query parameters

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to the enumeration value <b>import-graph</b> , which means the incremental data will be imported to the target graph. |

## Request Parameters

**Table 8-46** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-47** Request body parameters

| Parameter       | Mandatory | Type   | Description   |
|-----------------|-----------|--------|---|
| edgesetPath     | No        | String | Edge file directory or name   |
| edgesetFormat   | No        | String | Format of the edge data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.   |
| vertexsetPath   | No        | String | Vertex file directory or name   |
| vertexsetFormat | No        | String | Format of the vertex data set. Currently, only the CSV format is supported.<br>The CSV format is used by default.   |
| schemaPath      | No        | String | Path for storing the metadata file of the new data.   |
| logDir          | No        | String | Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.  |
| parallelEdge    | No        | Object | Repetitive edge processing  |
| action          | No        | String | Processing mode of repetitive edges. The value can be <b>allow</b> , <b>ignore</b> , or <b>override</b> . The default value is <b>allow</b> . <ul style="list-style-type: none"> <li>• <b>allow</b> indicates that repetitive edges are allowed.</li> <li>• <b>ignore</b> indicates that subsequent repetitive edges are ignored.</li> <li>• <b>override</b> indicates that the previous repetitive edges are overwritten.</li> </ul> |

| Parameter   | Mandatory | Type    | Description  |
|-------------|-----------|---------|--|
| ignoreLabel | No        | Boolean | <p>Whether to ignore labels on repetitive edges. The value is <b>true</b> or <b>false</b>, and the default value is <b>true</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Indicates that the repetitive edge definition does not contain the label. That is, the &lt;source vertex, target vertex&gt; indicates an edge, excluding the label information.</li> <li>• <b>false</b>: Indicates that the repetitive edge definition contains the label. That is, the &lt;source vertex, target vertex, label&gt; indicates an edge.</li> </ul> |
| delimiter   | No        | String  | <p>Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).</p>  |
| trimQuote   | No        | String  | <p>Field quote character in a CSV file. The default value is double quotation marks ("). It is used to enclose a field if the field contains separators or line breaks.</p>  |
| offline     | No        | Boolean | <p>Whether offline import is selected. The value is <b>true</b> or <b>false</b>, and the default value is <b>false</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import.</li> <li>• <b>false</b>: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.</li> </ul>   |

 NOTE

- For details about the value validity of the **edgesetPath**, **vertexsetPath**, **schemaPath**, and **logDir** character strings, see the [OBS Object Name Restrictions](#).

## Response Parameters

Status code: 200

**Table 8-48** Response body parameter

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobId        | String | Indicates the ID of an asynchronous job.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .    |

## Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true
  },
  "delimiter": ",",
  "trimQuote": "\"\"",
  "offline": true
}
```

## Example Response

Status code: 200

OK

```
{
  "jobId" : "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

**Status code: 400**

Bad Request

```
{
  "errorMessage" : "parameter format error",
  "errorCode" : "GES.8013"
}
```

**Status Code**

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

**8.1.2.8 Exporting a Graph**

**Function**

This API is used to export a graph.

**URI**

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-49** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-50** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>The value can be: <ul style="list-style-type: none"> <li><b>export-graph</b></li> </ul> |

## Request Parameters

**Table 8-51** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-52** Request body parameters

| Parameter       | Mandatory | Type   | Description                           |
|-----------------|-----------|--------|---------------------------------------|
| graphExportPath | Yes       | String | OBS path to which a graph is exported |
| edgeSetName     | Yes       | String | Exported edge file name               |
| vertexSetName   | Yes       | String | Exported vertex file name             |
| schemaName      | Yes       | String | Name of the exported metadata file    |

 **NOTE**

- For details about the value validity of the **graphExportPath** character strings, see the [OBS Object Name Restrictions](#).

## Response Parameters

**Status code: 200**



**Table 8-53** Response body parameter

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| jobId        | String | ID of an asynchronous job<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> .                   |

 **NOTE**

- For details about the value validity of the **graphExportPath** character strings, see the [OBS Object Name Restrictions](#).

### Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=export-graph
{
  "graphExportPath" : "demo_movie/",
  "edgeSetName" : "set_edge.csv",
  "vertexSetName" : "set_vertex.csv",
  "schemaName" : "set_schema.xml"
}
```

### Example Response

**Status code: 200**

OK

```
{
  "jobId" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

**Status code: 400**

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8011"
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.9 Clearing a Graph

#### Function

This API is used to clear all data in a graph.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-54** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-55** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to the enumeration value <b>clear-graph</b> , which means that all data in the target graph will be cleared. |

| Parameter      | Mandatory | Type    | Description   |
|----------------|-----------|---------|---|
| clear-metadata | No        | Boolean | Whether to clear graph metadata. Set this parameter to <b>true</b> . The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> . <ul style="list-style-type: none"> <li>• <b>true</b>: The metadata will be cleared.</li> <li>• <b>false</b>: The metadata will not be cleared.</li> </ul> |

## Request Parameters

**Table 8-56** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-57** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |

| Parameter | Type   | Description  |
|-----------|--------|--|
| jobId     | String | ID of an asynchronous job.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

## Example Request

POST http://Endpoint/v1.0/{project\_id}/graphs/{graph\_id}/action?action\_id=clear-graph&clear-metadata=true

## Example Response

**Status code: 200**

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8012"
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.10 Upgrading a Graph

#### Function

This API is used to upgrade a graph. The GES version is periodically upgraded. You can upgrade your graphs as required.

## URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-58** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-59** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to the enumeration value <b>upgrade</b> , which means that the graph will be upgraded. |

## Request Parameters

**Table 8-60** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-61** Request body parameters

| Parameter      | Mandatory | Type   | Description  |
|----------------|-----------|--------|--|
| upgradeVersion | Yes       | String | Target version, which must be later than the current version |

| Parameter    | Mandatory | Type    | Description   |
|--------------|-----------|---------|---|
| forceUpgrade | No        | Boolean | <p>Whether to upgrade forcibly. The value is <b>true</b> or <b>false</b>, and the default value is <b>false</b>.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: forcible upgrades, which will interrupt running tasks, such as long algorithm execution tasks. As a result, a small number of requests may fail.</li> <li>• <b>false</b>: non-forcible upgrades, which will wait for running services to complete. The upgrade process may be slow.</li> </ul> |

## Response Parameters

Status code: 200

Table 8-62 Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | <p>System prompt.</p> <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | <p>System prompt code.</p> <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | <p>ID of an asynchronous job</p> <p><b>NOTE</b><br/>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a>.</p>                 |

## Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=upgrade
{
  "upgradeVersion" : "1.1.8",
```

```
"forceUpgrade" : false  
}
```

## Example Response

**Status code: 200**

OK

```
{  
  "jobId" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"  
}
```

**Status code: 400**

Bad Request

```
{  
  "errorMessage" : "graph [demo] is not found",  
  "errorCode" : "GES.8011"  
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.11 Binding an EIP

#### Function

This API enables you to access GES by binding an elastic IP (EIP).

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-63** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-64** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to <b>bindEip</b> , which means that an EIP will be bound to the target graph. |

## Request Parameters

**Table 8-65** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-66** Request body parameters

| Parameter | Mandatory | Type   | Description                  |
|-----------|-----------|--------|------------------------------|
| eipId     | Yes       | String | ID of the elastic IP address |

## Response Parameters

Status code: 200



**Table 8-67** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |

### Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=bindEip
{
  "eipId" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

### Example Response

**Status code: 200**

OK

```
{}
```

**Status code: 400**

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8011"
}
```

### Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |

| Status Code | Description          |
|-------------|----------------------|
| 503         | Service unavailable. |

### 8.1.2.12 Unbinding an EIP

#### Function

If you do not need to use the EIP, this API enabled you to unbind the EIP to release network resources.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-68** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-69** Query parameters

| Parameter | Mandatory | Type   | Description   |
|-----------|-----------|--------|---|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to <b>unbindEip</b> , which means that the EIP will be unbound from the target graph. |

## Request Parameters

**Table 8-70** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-71** Request body parameters

| Parameter | Mandatory | Type   | Description                   |
|-----------|-----------|--------|-------------------------------|
| eipId     | Yes       | String | ID of the elastic IP address. |

## Response Parameters

**Status code: 200**

**Table 8-72** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/action?action_id=unbindEip
{
```

```
"eipId" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"  
}
```

## Example Responses

**Status code: 200**

OK

```
{ }
```

**Status code: 400**

Bad Request

```
{  
  "errorMessage" : "graph [demo] is not found",  
  "errorCode" : "GES.8011"  
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.13 Resizing a Graph

#### Function

This API is used to resize a graph instance.

 **NOTE**

After the graph is resized, you need to re-create all indexes.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/resize

**Table 8-73** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 8-74** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-75** Request body parameters

| Parameter | Mandatory | Type   | Description                                      |
|-----------|-----------|--|--|
| resize    | Yes       | <a href="#">GraphSizeTypeIndexReq</a> object | Graph specifications after the graph is resized. |

**Table 8-76** GraphSizeTypeIndexReq

| Parameter          | Mandatory | Type   | Description   |
|--------------------|-----------|--------|---|
| graphSizeTypeIndex | Yes       | String | Graph flavor. Currently, the value can be <b>2</b> , <b>3</b> , <b>4</b> , or <b>5</b> , indicating <b>10-million-edge</b> , <b>100-million-edge</b> , <b>1-billion-edge</b> , or <b>10-billion-edge</b> , respectively.<br>(graph_size_type_index) |

## Response Parameters

Status code: 200

Table 8-77 Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. <ul style="list-style-type: none"><li>• If execution succeeds, this parameter may be left blank.</li><li>• If execution fails, this parameter is used to display the error message.</li></ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"><li>• If execution succeeds, this parameter may be left blank.</li><li>• If execution fails, this parameter is used to display the error code.</li></ul>   |
| jobId        | String | Indicates the ID of the resize job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Job Management APIs</a> . |

## Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/resize
{
  "resize" : {
    "graphSizeTypeIndex" : "2"
  }
}
```

## Example Response

Status code: 200

OK

```
{}
```

Status code: 400

Bad Request

```
{
  "errorCode" : "GES.7001",
  "errorMessage" : "The graph is not running."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.14 Restarting a Graph

#### Function

This API is used to forcibly start a graph in the importing, exporting, running, or clearing state. If a graph is forcibly restarted, asynchronous tasks of the graph are failed state and the graph is stopped and started.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/action

**Table 8-78** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-79** Query parameters

| Parameter | Mandatory | Type   | Description  |
|-----------|-----------|--------|--|
| action_id | Yes       | String | Graph action ID<br>Set this parameter to <b>restart</b> , which means that the target graph will be restarted. |

## Request Parameters

**Table 8-80** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-81** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>                                    |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>                                       |
| jobId        | String | ID of a forcible restart job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |

## Example Request

POST https://Endpoint/v1.0/{project\_id}/graphs/{graph\_id}/action?action\_id=restart

## Example Response

Status code: 200



OK

```
{ }
```

**Status code: 400**

Bad Request

```
{
  "errorMessage" : "The request is invalid.",
  "errorCode" : "GES.7016"
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.2.15 Expanding a Graph

#### Function

This API is used to expand multiple secondary nodes dynamically. The expanded secondary nodes can process read requests, improving read performance.

#### NOTE

1. This API is not supported by graphs of the 10,000-edge and 10-billion-edge types.
2. Graphs cannot be resized after expansion.
3. If you want to resize and expand the graph, resize the graph before you expand it.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/expand

**Table 8-82** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

## Request Parameters

**Table 8-83** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-84** Request body parameters

| Parameter | Mandatory | Type                                  | Description           |
|-----------|-----------|---------------------------------------|-----------------------|
| expand    | Yes       | <a href="#">ReplicationReq</a> object | Expansion information |

**Table 8-85** ReplicationReq

| Parameter   | Mandatory | Type   | Description                   |
|-------------|-----------|--------|-------------------------------|
| replication | Yes       | String | Number of new nodes to expand |

## Response Parameters

Status code: 200

**Table 8-86** Response body parameters

| Parameter    | Type   | Description  |
|--------------|--------|--|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>                                |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>                              |
| jobId        | String | ID of the expansion job. This parameter is left blank when the request fails.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> . |

### Example Request

```
POST http://Endpoint/v1.0/{project_id}/graphs/{graph_id}/expand
{
  "expand": {
    "replication" : "1"
  }
}
```

### Example Responses

**Status code: 200**

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

**Status code: 400**

Bad Request

```
{
  "errorCode" : "GES.7015",
  "errorMessage" : "The graph is not running or stopped."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

## 8.1.3 Backup Management APIs

### 8.1.3.1 Viewing the List of All Backups

#### Function

This API is used to query the list of all backups.

#### URI

GET /v1.0/{project\_id}/graphs/backups

**Table 8-87** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 8-88** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |

## Request Parameters

**Table 8-89** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-90** Response body parameter

| Parameter    | Type                                    | Description  |
|--------------|---|--|
| errorMessage | String                                  | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String                                  | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul>    |
| backupCount  | Integer                                 | Total number of backups. This parameter is left blank if the request fails.  |
| backupList   | Array of <a href="#">backup</a> objects | List of all backups under the current project ID. This parameter is left blank if the request fails.   |

**Table 8-91** backup

| Parameter | Type   | Description |
|-----------|--------|-------------|
| id        | String | Backup ID   |

| Parameter          | Type   | Description   |
|--------------------|--------|---|
| name               | String | Backup name   |
| backupMethod       | String | Backup method. The value can be <b>auto</b> or <b>manual</b> .  |
| graphId            | String | ID of the graph associated with the backup  |
| graphName          | String | Name of the graph associated with the backup  |
| graphStatus        | String | Status of the graph associated with the backup  |
| graphSizeTypeIndex | String | Size of the graph associated with the backup  |
| dataStoreVersion   | String | Version of the graph associated with the backup   |
| arch               | String | CPU architecture of the graph node associated with the backup   |
| status             | String | Backup status: <ul style="list-style-type: none"> <li>● <b>backing_up</b>: indicates that a graph is being backed up.</li> <li>● <b>success</b>: indicates that a graph is successfully backed up.</li> <li>● <b>failed</b>: indicates that a graph fails to be backed up.</li> </ul> |
| startTimestamp     | Long   | Start timestamp of a backup job   |
| startTime          | String | Backup start time   |
| endTimestamp       | Long   | End timestamp of a backup job   |
| endTime            | String | Backup end time   |
| size               | Long   | Backup file size (MB)   |
| duration           | Long   | Backup duration (seconds)   |

## Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/backups?offset=0&limit=2
```

## Example Response

Example response with status code **200**:

OK

```
{
  "backupCount": 3,
```

```
"backupList": [
  {
    "id": "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name": "ges060803_nodelete-20210608135513",
    "backupMethod": "manual",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graphName": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.21",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623160513000,
    "startTime": "2021-06-08T13:55:13",
    "endTimestamp": 1623160568000,
    "endTime": "2021-06-08T13:56:08",
    "size": 1,
    "duration": 54,
    "encrypted": false
  },
  {
    "id": "7ed3f51d-816d-4651-9129-fe21b64b5c91",
    "name": "ges060803_nodelete_20210609203323_auto",
    "backupMethod": "auto",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graphName": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.21",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623242004000,
    "startTime": "2021-06-09T12:33:24",
    "endTimestamp": 1623242004000,
    "endTime": "2021-06-09T12:33:24",
    "size": 1,
    "duration": 0,
    "encrypted": false
  },
  {
    "id": "604bfb46-04dd-45fc-a9ae-df24a0705b9d",
    "name": "ges060802_nodelete-20210608135523",
    "backupMethod": "manual",
    "graphId": "9b9a05c2-0cdb-41ac-b55f-93caffb0519a",
    "graphName": "ges060802_nodelete",
    "graphStatus": "400",
    "graphSizeTypeIndex": "0",
    "dataStoreVersion": "2.2.23",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623160524000,
    "startTime": "2021-06-08T13:55:24",
    "endTimestamp": 1623160577000,
    "endTime": "2021-06-08T13:56:17",
    "size": 1,
    "duration": 53,
    "encrypted": false
  }
]
```

**Status code: 400**

Bad Request

```
{
  "errorCode": "GES.7006",
  "errorMessage": "The underlying graph engine has internal error."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.3.2 Viewing the Backup List of a Graph

#### Function

This API is used to query the backup list of a graph.

#### URI

GET /v1.0/{project\_id}/graphs/{graph\_id}/backups

**Table 8-92** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

**Table 8-93** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> .                           |



## Request Parameters

**Table 8-94** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-95** Response body parameters

| Parameter    | Type                                    | Description   |
|--------------|---|---|
| errorMessage | String                                  | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String                                  | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| backupCount  | Integer                                 | Total number of backups. This parameter is left blank if the request fails.   |
| backupList   | Array of <a href="#">backup</a> objects | List of backups of a specified graph under the current project. This parameter is left blank if the request fails.  |

**Table 8-96** backup

| Parameter | Type   | Description |
|-----------|--------|-------------|
| id        | String | Backup ID   |

| Parameter          | Type   | Description   |
|--------------------|--------|---|
| name               | String | Backup name   |
| backupMethod       | String | Backup method. The value can be <b>auto</b> or <b>manual</b> .  |
| graphId            | String | ID of the graph associated with the backup  |
| graphName          | String | Name of the graph associated with the backup  |
| graphStatus        | String | Status of the graph associated with the backup  |
| graphSizeTypeIndex | String | Size of the graph associated with the backup  |
| dataStoreVersion   | String | Version of the graph associated with the backup   |
| arch               | String | CPU architecture of the graph node associated with the backup   |
| status             | String | Backup status: <ul style="list-style-type: none"> <li>● <b>backing_up</b>: indicates that a graph is being backed up.</li> <li>● <b>success</b>: indicates that a graph is successfully backed up.</li> <li>● <b>failed</b>: indicates that a graph fails to be backed up.</li> </ul> |
| startTimestamp     | Long   | Start timestamp of a backup job   |
| startTime          | String | Backup start time   |
| endTimestamp       | Long   | End timestamp of a backup job   |
| endTime            | String | Backup end time   |
| size               | Long   | Backup file size (MB)   |
| duration           | Long   | Backup duration (seconds)   |

## Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/backups?offset=0&limit=2
```

## Example Response

Example response with status code **200**:

OK

```
{
  "backupCount": 2,
```

```

"backupList": [
  {
    "id": "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name": "ges060803_nodelete-20210608135513",
    "backupMethod": "manual",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graphName": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.22",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623160513000,
    "startTime": "2021-06-08T13:55:13",
    "endTimestamp": 1623160568000,
    "endTime": "2021-06-08T13:56:08",
    "size": 1,
    "duration": 54,
    "encrypted": false
  },
  {
    "id": "7ed3f51d-816d-4651-9129-fe21b64b5c91",
    "name": "ges060803_nodelete_20210609203323_auto",
    "backupMethod": "auto",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graphName": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.21",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623242004000,
    "startTime": "2021-06-09T12:33:24",
    "endTimestamp": 1623242004000,
    "endTime": "2021-06-09T12:33:24",
    "size": 1,
    "duration": 0,
    "encrypted": false
  }
]

```

**Status code: 400**

Bad Request

```

{
  "errorCode": "GES.7000",
  "errorMessage": "The graph does not exist or has been deleted."
}

```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |

| Status Code | Description          |
|-------------|----------------------|
| 503         | Service unavailable. |

### 8.1.3.3 Adding a Backup

#### Function

This API is used to add a backup. If data in the current graph is incorrect or faulty, you can start the backup graph to restore the data.

#### URI

POST /v1.0/{project\_id}/graphs/{graph\_id}/backups

**Table 8-97** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |

#### Request Parameters

**Table 8-98** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

#### Response Parameters

Status code: 200

**Table 8-99** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String | ID of the graph backup job.<br><b>NOTE</b><br>You can view the job execution status and obtain the return result by querying the job ID. For details, see <a href="#">Task Center APIs</a> .                      |

### Example Request

POST https://Endpoint/v1.0/{project\_id}/graphs/{graph\_id}/backups

### Example Response

Example response with status code **200**:

OK

```
{
  "jobId": "2c9080c48e55a6ba018e59e7d63600fb",
  "backupId": "105788ca-05da-454e-8cb0-9b95d2eec3a6"
}
```

**Status code: 400**

Bad Request

```
{
  "errorCode": "GES.7000",
  "errorMessage": "The graph does not exist or has been deleted."
}
```

### Status Code

| Status Code | Description            |
|-------------|------------------------|
| 200         | Request sent.          |
| 400         | Request error.         |
| 401         | Authentication failed. |

| Status Code | Description              |
|-------------|--------------------------|
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.3.4 Deleting a Backup

#### Function

This API is used to delete a backup.

#### URI

DELETE /v1.0/{project\_id}/graphs/{graph\_id}/backups/{backup\_id}

**Table 8-100** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| backup_id  | Yes       | String | Graph backup ID  |
| graph_id   | Yes       | String | Graph ID   |

#### Request Parameters

**Table 8-101** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-102** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

```
DELETE https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/backups/{backupId}
```

## Example Response

Status code: 200

OK

```
{}
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "Parameter error!",
  "errorCode" : "GES.0001"
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |

| Status Code | Description             |
|-------------|-------------------------|
| 500         | Internal service error. |
| 503         | Service unavailable.    |

## 8.1.4 Metadata Management APIs

### 8.1.4.1 Constraints

[Table 8-103](#) and [Table 8-104](#) list the metadata types.

**Table 8-103** Metadata property constraints

| Data Type  | Constraints   |
|------------|---|
| char       | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| char array | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| float      | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |



| Data Type | Constraints   |
|-----------|---|
| double    | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| bool      | <ul style="list-style-type: none"> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> </ul>   |
| long      | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| int       | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| date      | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |
| enum      | <ul style="list-style-type: none"> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> </ul>   |

| Data Type | Constraints   |
|-----------|---|
| string    | <ul style="list-style-type: none"> <li>• Less than (&lt;)</li> <li>• Greater than (&gt;)</li> <li>• Equal to (=)</li> <li>• Not equal to (!=)</li> <li>• In range (range)</li> <li>• Greater than or equal to (&gt;=)</li> <li>• Less than or equal to (&lt;=)</li> </ul> |

**Table 8-104** Property-level constraints

| Property Level                   | Constraints | Description                     |
|----------------------------------|-------------|---------------------------------|
| Single value/<br>Multiple values | has         | This property is contained.     |
| Single value/<br>Multiple values | hasNot      | This property is not contained. |

### 8.1.4.2 Querying the Metadata List

#### Function

This API is used to query the metadata list.

#### URI

GET /v1.0/{project\_id}/graphs/metadatas

**Table 8-105** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 8-106** Query parameters

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| limit     | No        | Integer | Maximum number of resources displayed on a single page. The default value is <b>10</b> . |

| Parameter | Mandatory | Type    | Description  |
|-----------|-----------|---------|--|
| offset    | No        | Integer | Start position of the request. The default value is <b>0</b> . |

## Request Parameters

**Table 8-107** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

**Status code: 200**

**Table 8-108** Response body parameters

| Parameter    | Type                                      | Description   |
|--------------|---|---|
| errorMessage | String                                    | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String                                    | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| schemaCount  | Integer                                   | Number of returned metadata files. This parameter is left blank if the request fails.   |
| schemaList   | Array of <a href="#">metadata</a> objects | List of all metadata files under the current project ID. This parameter is left blank if the request fails.   |

**Table 8-109** metadata

| Parameter      | Type   | Description                       |
|----------------|--------|-----------------------------------|
| id             | String | Metadata ID                       |
| name           | String | Metadata name                     |
| description    | String | Metadata description              |
| status         | String | Whether the metadata is available |
| metadataPath   | String | Metadata path                     |
| startTime      | String | Metadata creation time            |
| lastUpdateTime | String | Last update time of the metadata  |

### Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/metadatas?offset=10&limit=100
```

### Example Response

**Status code: 200**

OK

```
{
  "schemaCount": 1,
  "schemaList": [ {
    "id": "ff7dddc4-6402-43d7-9aed-c5ec677b47fa",
    "name": "schema_demo",
    "description": "",
    "status": "200",
    "metadataPath": "ges-graphs/demo_movie/schema.xml",
    "startTime": "2018-07-23T02:59:41",
    "lastUpdateTime": "2018-07-23T02:59:41"
  } ]
}
```

**Status code: 500**

Internal Server Error

```
{
  "errorCode": "GES.7006",
  "errorMessage": "The underlying graph engine has internal error."
}
```

### Status Code

| Status Code | Description            |
|-------------|------------------------|
| 200         | Request sent.          |
| 400         | Request error.         |
| 401         | Authentication failed. |

| Status Code | Description              |
|-------------|--------------------------|
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.4.3 Querying Metadata

#### Function

This API is used to query the metadata of a graph.

#### URI

GET /v1.0/{project\_id}/graphs/metadatas/{metadata\_id}

**Table 8-110** URI parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| project_id  | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| metadata_id | Yes       | String | Metadata ID  |

#### Request Parameters

**Table 8-111** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-112** Response body parameters

| Parameter    | Type                          | Description  |
|--------------|-------------------------------|--|
| errorMessage | String                        | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul> |
| errorCode    | String                        | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul>    |
| gesMetadata  | <b>GesMetaDat</b><br>a object | Object for storing metadata message information.   |

**Table 8-113** GesMetaData

| Parameter | Type                          | Description              |
|-----------|-------------------------------|--------------------------|
| labels    | Array of <b>Label</b> objects | Label data structure set |

**Table 8-114** Label

| Parameter  | Type   | Description  |
|------------|--------|--|
| name       | String | Label name   |
| properties | Object | Maps of metadata properties. The key-value pairs are those in the imported metadata. |

## Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/metadatas/{metadata_id}
```

## Example Response

Example response with status code **200**:

OK

```
{
  "gesMetadata": {
```

```
"labels": [
  {
    "name": "friends",
    "properties": null
  },
  {
    "name": "movie",
    "properties": [
      {
        "dataType": "string",
        "name": "Title",
        "cardinality": "single"
      },
      {
        "dataType": "int",
        "name": "Year",
        "cardinality": "single"
      },
      {
        "dataType": "string",
        "name": "Genres",
        "cardinality": "set"
      }
    ]
  },
  {
    "name": "user",
    "properties": [
      {
        "dataType": "string",
        "name": "Title",
        "cardinality": "single"
      },
      {
        "dataType": "int",
        "name": "Year",
        "cardinality": "single"
      },
      {
        "dataType": "string",
        "name": "Genres",
        "cardinality": "set"
      }
    ]
  },
  {
    "typeName1": "F",
    "typeName2": "M",
    "typeNameCount": "2",
    "dataType": "enum",
    "name": "Gender",
    "cardinality": "single"
  },
  {
    "typeName1": "Under 18",
    "typeName2": "18-24",
    "typeName3": "25-34",
    "typeName4": "35-44",
    "typeNameCount": "7",
    "dataType": "enum",
    "name": "Age",
    "typeName5": "45-49",
    "typeName6": "50-55",
    "cardinality": "single",
    "typeName7": "56+"
  },
  {
    "dataType": "string",
    "name": "Occupation",
    "cardinality": "single"
  },
  {
```

```
        "dataType": "char array",
        "name": "Zip-code",
        "maxDataSize": "12",
        "cardinality": "single"
      }
    ]
  },
  {
    "name": "rate",
    "properties": [
      {
        "dataType": "int",
        "name": "Score",
        "cardinality": "single"
      },
      {
        "dataType": "date",
        "name": "Datetime",
        "cardinality": "single"
      }
    ]
  }
]
}
```

**Status code: 500**

Internal Server Error

```
{
  "errorCode": "GES.7006",
  "errorMessage": "The underlying graph engine has internal error."
}
```

### Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.4.4 Adding Metadata

#### Function

This API is used to add the metadata.



## URI

POST /v1.0/{project\_id}/graphs/metadatas

**Table 8-115** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 8-116** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-117** Request body parameters

| Parameter    | Mandatory | Type    | Description  |
|--------------|-----------|---------|--|
| metadataPath | Yes       | String  | Metadata storage address   |
| name         | Yes       | String  | Metadata name, which contains 1 to 64 characters consisting of only letters, digits, and underscores (_)   |
| description  | Yes       | String  | Metadata description   |
| isOverwrite  | Yes       | Boolean | Whether to overwrite existing files The value can be <b>true</b> or <b>false</b> . The default value is <b>false</b> . <ul style="list-style-type: none"> <li><b>true</b>: Existing file will be overwritten.</li> <li><b>false</b>: Existing file will not be overwritten.</li> </ul> |

| Parameter   | Mandatory | Type   | Description                                      |
|-------------|-----------|--------|--|
| gesMetadata | Yes       | Object | Object for storing metadata message information. |

## Response Parameters

Status code: 200

Table 8-118 Response body parameter

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| id           | String | Metadata ID   |
| name         | String | Metadata name   |

## Example Request

```
POST https://Endpoint/v1.0/{project_id}/graphs/metadatas
{
  "metadataPath": "gesdata/demo_movie/schema.xml",
  "name": "movie_schema",
  "description": "xxxxx",
  "isOverwrite": "true",
  "encryption": {
    "enable": true,
    "masterKeyId": "2fc79d04-7010-4f63-9534-d8de74ab67e0"
  },
  "gesMetadata": {
    "labels": [
      {
        "name": "friends",
        "properties": null
      },
      {
        "name": "movie",
        "properties": [
          {
            "dataType": "string",
            "name": "Title",
            "cardinality": "single"
          }
        ]
      }
    ]
  }
}
```

```
{
  {
    "dataType": "int",
    "name": "Year",
    "cardinality": "single"
  },
  {
    "dataType": "string",
    "name": "Genres",
    "cardinality": "set"
  }
]
},
{
  "name": "user",
  "properties": [
    {
      "dataType": "string",
      "name": "Name",
      "cardinality": "single"
    },
    {
      "typeName1": "F",
      "typeName2": "M",
      "typeNameCount": "2",
      "dataType": "enum",
      "name": "Gender",
      "cardinality": "single"
    },
    {
      "typeName1": "Under 18",
      "typeName2": "18-24",
      "typeName3": "25-34",
      "typeName4": "35-44",
      "typeNameCount": "7",
      "dataType": "enum",
      "name": "Age",
      "typeName5": "45-49",
      "typeName6": "50-55",
      "cardinality": "single",
      "typeName7": "56+"
    },
    {
      "dataType": "string",
      "name": "Occupation",
      "cardinality": "single"
    },
    {
      "dataType": "char array",
      "name": "Zip-code",
      "maxDataSize": "12",
      "cardinality": "single"
    }
  ]
},
{
  "name": "rate",
  "properties": [
    {
      "dataType": "int",
      "name": "Score",
      "cardinality": "single"
    },
    {
      "dataType": "date",
      "name": "Datetime",
      "cardinality": "single"
    }
  ]
}
}
```

```
    ]  
  }  
}
```

## Example Response

Example response with status code **200**:

OK

```
{  
  "id" : "ff8080815f9a3c84015f9a438ff70001",  
  "name" : "movie_schema"  
}
```

**Status code: 500**

Internal Server Error

```
{  
  "errorCode" : "GES.2067",  
  "errorMessage" : "name: 1 to 64 characters, only letters, digits, and underscores(_) are allowed."  
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.4.5 Deleting Metadata

#### Function

This API is used to delete the metadata.

#### URI

DELETE /v1.0/{project\_id}/graphs/metadatas/{metadata\_id}

**Table 8-119** URI parameters

| Parameter   | Mandatory | Type   | Description  |
|-------------|-----------|--------|--|
| project_id  | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| metadata_id | Yes       | String | Metadata ID  |

## Request Parameters

**Table 8-120** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-121** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |

## Example Request

```
DELETE https://Endpoint/v1.0/{project_id}/graphs/metadatas/{metadata_id}
```

## Example Response

**Status code: 200**

OK

```
{ }
```

**Status code: 400**

Bad Request

```
{
  "errorCode" : "GES.7024",
  "errorMessage" : "The metadata is not exist or has been deleted."
}
```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.4.6 Importing Metadata from OBS

#### Function

This API is used to import metadata from OBS.

#### URI

POST /v1.0/{project\_id}/graphs/metadata/upload\_from\_obs

**Table 8-122** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

## Request Parameters

**Table 8-123** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

**Table 8-124** Request body parameters

| Parameter    | Mandatory | Type   | Description                   |
|--------------|-----------|--------|-------------------------------|
| metadataPath | Yes       | String | Path for storing the metadata |
| name         | Yes       | String | Metadata name                 |
| description  | No        | String | Metadata description          |

## Response Parameters

**Status code: 200**

**Table 8-125** Response body parameters

| Parameter    | Type   | Description   |
|--------------|--------|---|
| errorMessage | String | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |

| Parameter | Type   | Description   |
|-----------|--------|---------------|
| id        | String | Metadata ID   |
| name      | String | Metadata name |

### Example Request

```
POST https://Endpoint/v1.0/{project_id}/graphs/metadata/upload_from_obs
{
  "metadataPath": "devdata/unionsdk/schema.xml",
  "name": "test_schema",
  "description": "",
  "encryption": {}
}
```

### Example Response

Example response with status code **200**:

OK

```
{
  "id" : "d30d2e94-f2ee-4344-af49-eb27fd002eea",
  "name" : "test_schema"
}
```

#### Status code: **404**

Internal Server Error

```
{
  "errorCode": "GES.0016",
  "errorMessage": "Resource not found"
}
```

### Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

## 8.1.5 Task Center APIs



### 8.1.5.1 Querying Job Status on the Management Plane

#### Function

This API is used to query the execution status of a job. Asynchronous APIs that are used to create, stop, start, delete, and import graphs will return job IDs after commands are sent. You can query the job execution status according to the job IDs.

#### URI

GET /v1.0/{project\_id}/graphs/{graph\_id}/jobs/{job\_id}/status

**Table 8-126** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |
| graph_id   | Yes       | String | Graph ID   |
| job_id     | Yes       | String | ID of the asynchronous job   |

#### Request Parameters

**Table 8-127** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

#### Response Parameters

Status code: 200

**Table 8-128** Response body parameters

| Parameter    | Type                    | Description   |
|--------------|-------------------------|---|
| errorMessage | String                  | System prompt. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String                  | System prompt code. <ul style="list-style-type: none"> <li>• If execution succeeds, this parameter may be left blank.</li> <li>• If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobId        | String                  | Job ID  |
| status       | String                  | Job status: <ul style="list-style-type: none"> <li>• <b>pending</b></li> <li>• <b>running</b></li> <li>• <b>success</b></li> <li>• <b>failed</b></li> </ul>   |
| jobType      | Integer                 | Job type  |
| jobName      | String                  | Job name  |
| relatedGraph | String                  | Associated graph name   |
| beginTime    | String                  | Job start time (UTC). The format is (yyyy-MM-dd HH:mm:ss).  |
| endTime      | String                  | Job end time (UTC). The format is (yyyy-MM-dd HH:mm:ss).  |
| jobDetail    | <b>JobDetail</b> object | This parameter is returned only when <b>jobName</b> is set to <b>ImportGraph</b> and is used to display graph import details.   |
| failReason   | String                  | Job failure cause   |
| jobProgress  | Double                  | Job execution progress. It is a reserved field, and not used currently.   |

**Table 8-129** JobDetail

| Parameter  | Type                               | Description               |
|------------|------------------------------------|---------------------------|
| schemaPath | Array of <b>schemaPath</b> objects | Path for storing metadata |

| Parameter     | Type   | Description                          |
|---------------|--|--------------------------------------|
| edgesetPath   | Array of <a href="#">edgesetPath</a> objects   | Path for storing the edge data set   |
| vertexsetPath | Array of <a href="#">vertexsetPath</a> objects | Path for storing the vertex data set |

**Table 8-130** schemaPath

| Parameter | Type   | Description  |
|-----------|--------|--|
| path      | String | OBS storage path   |
| log       | String | Import log   |
| status    | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |
| cause     | String | Import failure cause   |

**Table 8-131** edgesetPath

| Parameter   | Type   | Description  |
|-------------|--------|--|
| path        | String | OBS storage path   |
| log         | String | Import log   |
| status      | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |
| cause       | String | Import failure cause   |
| totalLines  | Long   | The total number of imported lines, which is subject to your quota and the specifications of the created graph. The value <b>-1</b> indicates that this field is not returned in the current version.            |
| failedLines | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |

| Parameter       | Type | Description  |
|-----------------|------|--|
| successfulLines | Long | Lines imported successfully. The value <b>-1</b> indicates that this field is not returned in the current version. |

**Table 8-132** vertexsetPath

| Parameter       | Type   | Description  |
|-----------------|--------|--|
| path            | String | OBS storage path   |
| log             | String | Import log   |
| status          | String | OBS file status: <ul style="list-style-type: none"> <li>• <b>success</b>: Imported successfully.</li> <li>• <b>Failed</b>: Failed to import the file.</li> <li>• <b>partFailed</b>: Partially failed.</li> </ul> |
| cause           | String | Import failure cause   |
| totalLines      | Long   | Total number of imported lines. The value is subject to your quota and specifications of the created graph. If the value is <b>-1</b> , this parameter is not supported by the current version.                  |
| failedLines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successfulLines | Long   | Lines imported successfully. The value <b>-1</b> indicates that this field is not returned in the current version.   |

## Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/jobs/{job_id}/status
```

## Example Response

**Status code: 200**

OK

```
Http Status Code: 200
{
  "jobId": "ff80808167f09aaa0167f19b35ec0305",
  "status": "success",
  "jobType": "GraphManagement",
  "jobName": "ImportGraph",
  "relatedGraph": "GES_UI_AUTO",
  "beginTime": "2018-11-27T21:39:00",
  "endTime": "2018-11-27T21:39:56",
  "jobDetail": {
```

```

"vertexsetPath": [
  {
    "path": "ges-ui/auDatas/list_set_vertex.csv",
    "log": null,
    "cause": null,
    "status": "success"
  }
],
"edgesetPath": [
  {
    "path": "ges-ui/auDatas/list_set_edge.csv",
    "log": null,
    "cause": null,
    "status": "success"
  }
],
"schemaPath": [
  {
    "path": "ges-ui/auDatas/list_set_schema.xml",
    "log": null,
    "cause": null,
    "status": "success"
  }
]
},
"jobProgress": 0
}

```

**Status code: 400**

Bad Request

```

{
  "errorMessage" : "can not find job, jobId is ff808081646e81d40164c5fb414b2b1a1",
  "errorCode" : "GES.8301"
}

```

## Status Code

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

### 8.1.5.2 Querying Job Details in the Job Center

#### Function

This API is used to query asynchronous job details in the job center on the management plane. Asynchronous jobs include creating, closing, starting, deleting, adding, importing, exporting, and upgrading graphs, as well as adding backups.

## URI

GET /v1.0/{project\_id}/graphs/jobs

**Table 8-133** URI parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| project_id | Yes       | String | Project ID. For details about how to obtain the project ID, see <a href="#">Obtaining a Project ID</a> . |

**Table 8-134** Query parameters

| Parameter  | Mandatory | Type   | Description  |
|------------|-----------|--------|--|
| endTime    | No        | String | Job end date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.                                  |
| graph_name | No        | String | Associated graph name  |
| limit      | No        | String | Maximum number of resources displayed on a single page. The default value is <b>10</b> .   |
| offset     | No        | String | Start position of the request. The default value is <b>0</b> .   |
| startTime  | No        | String | Job start date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.                                |
| status     | No        | String | Job status. Possible values: <ul style="list-style-type: none"> <li>● running</li> <li>● waiting</li> <li>● success</li> <li>● failed</li> </ul> |

## Request Parameters

**Table 8-135** Parameters in the request header

| Parameter    | Mandatory | Type   | Description   |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes       | String | User token.<br>Used to obtain the permission to use APIs. For details about how to obtain the token, see <a href="#">Authentication of Management Plane APIs</a> . The value of <b>X-Subject-Token</b> in the response header is the token. |

## Response Parameters

Status code: 200

**Table 8-136** Response body parameter

| Parameter    | Type                        | Description   |
|--------------|-----------------------------|---|
| errorMessage | String                      | System prompt. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error message.</li> </ul>   |
| errorCode    | String                      | System prompt code. <ul style="list-style-type: none"> <li>If execution succeeds, this parameter may be left blank.</li> <li>If execution fails, this parameter is used to display the error code.</li> </ul> |
| jobCount     | Integer                     | Total number of jobs  |
| jobList      | Array of <b>Job</b> objects | Task list   |

**Table 8-137** Job

| Parameter | Type   | Description |
|-----------|--------|-------------|
| jobId     | String | Job ID      |

| Parameter    | Type                    | Description   |
|--------------|-------------------------|---|
| status       | String                  | Job status. <ul style="list-style-type: none"> <li>• <b>pending</b></li> <li>• <b>running</b></li> <li>• <b>success</b></li> <li>• <b>failed</b></li> </ul> |
| jobType      | Integer                 | Task type   |
| jobName      | String                  | Task name   |
| relatedGraph | String                  | Associated graph name   |
| beginTime    | String                  | Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.  |
| endTime      | String                  | Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.  |
| jobDetail    | <b>JobDetail</b> object | This parameter is returned only when <b>jobName</b> is set to <b>ImportGraph</b> and is used to display graph import details.                               |
| failReason   | String                  | Job failure cause   |
| jobProgress  | Double                  | Job execution progress. It is a reserved field, and not used currently.   |

**Table 8-138** JobDetail

| Parameter     | Type                                  | Description                          |
|---------------|---------------------------------------|--------------------------------------|
| schemaPath    | Array of <b>schemaPath</b> objects    | Path for storing metadata            |
| edgesetPath   | Array of <b>edgesetPath</b> objects   | Path for storing the edge data set   |
| vertexsetPath | Array of <b>vertexsetPath</b> objects | Path for storing the vertex data set |

**Table 8-139** schemaPath

| Parameter | Type   | Description      |
|-----------|--------|------------------|
| path      | String | OBS storage path |



| Parameter | Type   | Description  |
|-----------|--------|--|
| log       | String | Import log   |
| status    | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause     | String | Import failure cause   |

**Table 8-140** edgesetPath

| Parameter       | Type   | Description  |
|-----------------|--------|--|
| path            | String | OBS storage path   |
| log             | String | Import log   |
| status          | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause           | String | Import failure cause   |
| totalLines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failedLines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successfulLines | Long   | Lines imported successfully. The value <b>-1</b> indicates that this field is not returned in the current version.   |

**Table 8-141** vertexsetPath

| Parameter | Type   | Description      |
|-----------|--------|------------------|
| path      | String | OBS storage path |
| log       | String | Import log       |

| Parameter       | Type   | Description  |
|-----------------|--------|--|
| status          | String | OBS file status: <ul style="list-style-type: none"> <li>● <b>success</b>: Imported successfully.</li> <li>● <b>Failed</b>: Failed to import the file.</li> <li>● <b>partFailed</b>: Partially failed.</li> </ul> |
| cause           | String | Import failure cause   |
| totalLines      | Long   | Total number of imported lines. The value <b>-1</b> indicates that this field is not returned in the current version.  |
| failedLines     | Long   | Lines failed to be imported. The value <b>-1</b> indicates that this field is not returned in the current version.   |
| successfulLines | Long   | Lines imported successfully. The value <b>-1</b> indicates that this field is not returned in the current version.   |

## Example Request

```
GET https://Endpoint/v1.0/{project_id}/graphs/jobs?offset=0&limit=100
```

## Example Response

Example response with status code **200**:

OK

```
Http Status Code: 200
{
  "jobCount": 136,
  "jobList": [
    {
      "jobId": "ff80808167bb90340167bc3c7b5b026a",
      "status": "success",
      "jobType": "GraphManagement",
      "jobName": "ImportGraph",
      "relatedGraph": "test1217",
      "beginTime": "2018-12-17T12:55:40",
      "endTime": "2018-12-17T12:56:32",
      "jobDetail": {
        "vertexsetPath": null,
        "edgesetPath": [
          {
            "path": "hkmovie/edge.csv",
            "log": null,
            "cause": null,
            "status": "success"
          }
        ],
        "schemaPath": [
          {
            "path": "hkmovie/schema.xml",
            "log": null,
            "cause": null,
            "status": "success"
          }
        ]
      }
    }
  ]
}
```

```

    ]
  },
  "jobProgress": 0
},
{
  "jobId": "ff80808167bb90340167bc5d0b1d0358",
  "status": "success",
  "jobType": "GraphManagement",
  "jobName": "DeleteGraph",
  "relatedGraph": "test1218",
  "beginTime": "2018-12-17T13:31:14",
  "endTime": "2018-12-17T13:34:48",
  "jobProgress": 0
}
]
}

```

**Status code: 400**

Bad Request

```

{
  "errorMessage": "failed",
  "errorCode": "GES.9999"
}

```

**Status Code**

| Status Code | Description              |
|-------------|--------------------------|
| 200         | Request sent.            |
| 400         | Request error.           |
| 401         | Authentication failed.   |
| 403         | No operation permission. |
| 404         | No resources found.      |
| 500         | Internal service error.  |
| 503         | Service unavailable.     |

# 9 Appendix

## 9.1 Status Codes

[Table 9-1](#) describes status codes.

**Table 9-1** Status codes

| Status Code | Message                       | Description   |
|-------------|-------------------------------|---|
| 100         | Continue                      | The client should continue with its request. This interim response is used to inform the client that part of the request has been received and has not yet been rejected by the server. |
| 101         | Switching Protocols           | The protocol should be switched. The protocol can only be switched to a newer protocol. For example, the current HTTP protocol is switched to a later version of HTTP.                  |
| 201         | Created                       | The request has been fulfilled and a new resource has been created.   |
| 202         | Accepted                      | The request has been received but has not been processed.   |
| 203         | Non-Authoritative Information | The server has successfully processed the request, but is returning information that may be from another source.  |
| 204         | NoContent                     | The request has been fulfilled, but the HTTP response does not contain a response body. The status code is returned in response to an HTTP OPTIONS request.                             |

| Stat<br>us<br>Cod<br>e | Message           | Description   |
|------------------------|-------------------|---|
| 205                    | Reset Content     | The server has successfully processed the request, but does not return any content.   |
| 206                    | Partial Content   | The server has successfully processed the partial GET request.  |
| 300                    | Multiple Choices  | There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which a user terminal (such as a browser) can choose the most appropriate one. |
| 301                    | Moved Permanently | The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.   |
| 302                    | Found             | The requested resource resides temporarily under a different URI.   |
| 303                    | See Other         | The response to the request can be found under a different URI.<br>The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.  |
| 304                    | Not Modified      | The requested resource has not been modified. In such a case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.  |
| 305                    | Use Proxy         | The requested resource is available only through a proxy.   |
| 306                    | Unused            | The HTTP status code is no longer used.   |
| 400                    | BadRequest        | Invalid request.<br>The client should modify the request instead of re-initiating it.   |
| 401                    | Unauthorized      | This status code is returned after the client provides the authentication information, indicating that the authentication information is incorrect or invalid.  |
| 402                    | Payment Required  | This status code is reserved for future use.  |

| Stat<br>us<br>Cod<br>e | Message                       | Description  |
|------------------------|-------------------------------|--|
| 403                    | Forbidden                     | The server has received the request and understood it, but the server is refusing to respond to it.<br>The server has received and understood the request; yet it refused to respond, because the request is set to deny access. Do not retry the request before modification. |
| 404                    | NotFound                      | The requested resource could not be found.<br>The client should modify the request instead of re-initiating it.  |
| 405                    | MethodNotAllowed              | The method specified in the request is not supported by the requested resource.<br>The client should modify the request instead of re-initiating it.   |
| 406                    | Not Acceptable                | The server could not fulfill the request according to the content characteristics of the request.  |
| 407                    | Proxy Authentication Required | This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.  |
| 408                    | Request Time-out              | The server timed out waiting for the request.<br>The client may repeat the request without modifications at any time later.  |
| 409                    | Conflict                      | The request could not be processed due to a conflict in the request.<br>This status code indicates that the resource that the client is attempting to create already exists, or that the request has failed to be processed because of the update of the conflict request.     |
| 410                    | Gone                          | The requested resource cannot be found.<br>The status code indicates that the requested resource has been deleted permanently.   |
| 411                    | Length Required               | The server is refusing to process the request without a defined <b>Content-Length</b> .  |
| 412                    | Precondition Failed           | The server does not meet one of the preconditions that the requester puts on the request.  |

| Stat<br>us<br>Cod<br>e | Message                         | Description  |
|------------------------|---------------------------------|--|
| 413                    | Request Entity Too Large        | The server is refusing to process a request because the request entity is too large for the server to process. The server may disable the connection to prevent the client from sending requests consecutively. If the server is only temporarily unable to process the request, the response will contain a <b>Retry-After</b> header field.  |
| 414                    | Request-URI Too Large           | The Request-URI is too long for the server to process.   |
| 415                    | Unsupported Media Type          | The server does not support the media type in the request.   |
| 416                    | Requested range not satisfiable | The requested range is invalid.  |
| 417                    | Expectation Failed              | The server has failed to meet the requirements of the <b>Expect</b> request-header field.  |
| 422                    | UnprocessableEntity             | The request was well-formed but was unable to be followed due to semantic errors.  |
| 429                    | TooManyRequests                 | The client has sent excessive number of requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server has received an excessive number of requests within a given time (beyond its processing capability). In this case, the client should resend the request after the time specified in the <b>Retry-After</b> header of the response has elapsed. |
| 500                    | InternalServerError             | The server is able to receive the request but unable to understand it.   |
| 501                    | Not Implemented                 | The server does not support the requested function.  |
| 502                    | Bad Gateway                     | The server was acting as a gateway or proxy and received an invalid request from the remote server.  |
| 503                    | ServiceUnavailable              | The requested service is invalid.<br>The client should modify the request instead of re-initiating it.   |
| 504                    | ServerTimeout                   | The request cannot be fulfilled within a given time. This status code is returned to the client only when the <b>Timeout</b> parameter is specified in the request.  |
| 505                    | HTTP Version not supported      | The server does not support the HTTP protocol version used in the request.   |

## 9.2 Error Codes

### 9.2.1 Error Codes for Management Plane APIs

If an error occurs in API calling, no result is returned. Identify the cause of error based on the error codes of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you fail to locate the cause of the error, contact technical support and provide the error code for troubleshooting.

**Table 9-2** Error codes

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message                                       | Description   | Solution   |
|--------------------|---------------|--|---|--|
| 400                | GES.0<br>001  | Incorrect<br>parameter.                                | Incorrect<br>parameter.                                   | <ol style="list-style-type: none"> <li>1. Check whether the project ID or graph ID in the URL is correct.</li> <li>2. Check whether the request header is correct, for example, whether <b>X-Auth-Token</b> is correct.</li> </ol>                           |
| 400                | GES.0<br>016  | Resource not<br>found                                  | Resource<br>not found.                                    | <ol style="list-style-type: none"> <li>1. Check whether the project ID in the URL is the same as the project ID of the token.</li> <li>2. Check whether the project ID in the URL is the same as the project ID of the graph.</li> </ol>                     |
| 400                | GES.7<br>000  | The graph<br>does not exist<br>or has been<br>deleted. | The graph<br>does not<br>exist or has<br>been<br>deleted. | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. Check whether the project ID or graph ID in the URL is correct.</li> </ol>   |
| 400                | GES.7<br>001  | The graph is<br>not running.                           | The graph is<br>not running.                              | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is <b>200</b>.</li> </ol> |



| Statu<br>s<br>Code | Error<br>Code | Error<br>Message  | Description   | Solution   |
|--------------------|---------------|---|---|--|
| 400                | GES.7<br>002  | The graph is being backed up.   | The graph is being backed up.   | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is <b>903</b>.</li> </ol>               |
| 400                | GES.7<br>003  | The graph is being stopped or is stopped.                               | The graph is being stopped or is stopped.                               | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is <b>900</b> or <b>901</b>.</li> </ol> |
| 400                | GES.7<br>004  | Components at the IaaS layer are faulty.                                | Components at the IaaS layer are faulty.                                | Check whether the components at the IaaS layer, such as VPC, ECS, and OBS, are faulty.   |
| 408                | GES.7<br>005  | The underlying service of the graph engine is unavailable.              | The underlying service of the graph engine is unavailable.              | Try again later or contact technical personnel.  |
| 400                | GES.7<br>006  | An internal error occurs in the underlying service of the graph engine. | An internal error occurs in the underlying service of the graph engine. | Try again later or contact technical support.  |
| 400                | GES.7<br>007  | The job does not exist.   | The job does not exist.   | Check whether the job ID in the URL is correct.  |
| 400                | GES.7<br>008  | The job is stopped.   | The job is stopped.   | Jobs cannot be stopped repeatedly.   |
| 400                | GES.7<br>009  | The job operation is not supported.                                     | The job operation is not supported.                                     | The job operation is not supported.  |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message   | Description  | Solution  |
|--------------------|---------------|--|--|---|
| 400                | GES.7<br>010  | Failed to verify the schema and data files.                        | Failed to verify the schema and data files.                        | Check whether the schema file matches the edge and vertex data files.   |
| 400                | GES.7<br>011  | The path or name of the schema or data file is invalid.            | The path or name of the schema or data file is invalid.            | Check whether the name of the scheme, vertex, or edge data file is valid. The name can contain only letters, digits, underscores (_), exclamation marks (!), hyphens (-), dot marks (.), asterisks (*), left brackets, right brackets, and slashes (/).                       |
| 400                | GES.7<br>012  | Failed to verify the graph name.                                   | Failed to verify the graph name.                                   | Check the graph name. The name contains 4 to 64 characters, starting with a letter. Only letters, digits, and underscores (_) are supported.  |
| 400                | GES.7<br>013  | The graph name already exists.                                     | The graph name already exists.                                     | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. Query the graph list returned in the preceding step and check whether the <b>name</b> field in the request body already exists.</li> </ol>                                |
| 400                | GES.7<br>014  | An error is reported when the metadata verification API is called. | An error is reported when the metadata verification API is called. | Check whether the value of <b>action_id</b> is <b>check-schema</b> .  |
| 400                | GES.7<br>015  | The graph is not running or is stopped.                            | The graph is not running or is stopped.                            | <ol style="list-style-type: none"> <li>1. Call the graph query API to query all graphs.</li> <li>2. View the graph list returned in the preceding step and check whether the graph corresponding to the graph ID in the URL exists or is in the <b>900</b> status.</li> </ol> |
| 400                | GES.7<br>016  | The request body or header is invalid.                             | The request body or header is invalid.                             | Check the API reference and ensure that every configuration item in the request body and header is correctly configured.  |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message   | Description  | Solution   |
|--------------------|---------------|--|--|--|
| 400                | GES.7<br>017  | The object does not exist. Check whether the bucket or object name is correct. | The object does not exist. Check whether the bucket or object name is correct. | Check whether the schema, vertex, and edge data files in the request body exist on OBS.  |
| 400                | GES.7<br>018  | The number of graphs or edges reaches the upper limit.                         | The number of graphs or edges reaches the upper limit.                         | Call the quota query API to check whether graphs have available quotas.  |
| 400                | GES.7<br>019  | The number of graph backups reaches the upper limit.                           | The number of graph backups reaches the upper limit.                           | Call the quota query API to check whether graph backups have available quotas.   |
| 400                | GES.7<br>020  | The VPC does not exist.  | The VPC does not exist.  | Check whether the VPC ID in the request body exists.   |
| 400                | GES.7<br>021  | The subnet cannot be found in the specified VPC.                               | The subnet cannot be found in the specified VPC.                               | Check whether the subnet ID in the request body exists or belongs to the preceding VPC.  |
| 400                | GES.7<br>022  | The security group does not exist.   | The security group does not exist.   | Check whether the security group ID in the request body exists.  |
| 400                | GES.7<br>023  | The graph size index is invalid.   | The graph size index is invalid.   | Check whether the graph size index in the request body is valid.   |
| 400                | GES.7<br>024  | The graph backup does not exist or has been deleted.                           | The graph backup does not exist or has been deleted.                           | <ol style="list-style-type: none"> <li>1. Call the backup query API to query all backups of a specified graph.</li> <li>2. Check whether the backup ID or graph ID in the URL is correct.</li> </ol> |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message  | Description   | Solution  |
|--------------------|---------------|---|---|---|
| 400                | GES.7<br>027  | Failed to<br>create an<br>agency.                                   | Failed to<br>create an<br>agency.                                   | <ol style="list-style-type: none"> <li>1. Assign the security administrator role to the user group to which the user belongs.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical support.</li> </ol>   |
| 400                | GES.7<br>028  | Failed to<br>authorize an<br>agency.                                | Failed to<br>authorize an<br>agency.                                | <ol style="list-style-type: none"> <li>1. Assign the security administrator role to the user group to which the user belongs.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.7<br>029  | The agency<br>resource<br>exceeds the<br>quota limit.               | The agency<br>resource<br>exceeds the<br>quota limit.               | Check whether the agency resource reaches the quota limit on the IAM page.  |
| 400                | GES.7<br>030  | Agency query<br>error.  | Agency<br>query error.  | Check the error message for detailed information.   |
| 400                | GES.7<br>031  | Invalid<br>binding type<br>of an EIP.                               | Invalid<br>binding type<br>of an EIP.                               | <p>Confirm the EIP binding type. The value can be either of the following:</p> <ul style="list-style-type: none"> <li>• <b>bind_existing</b></li> </ul>   |
| 400                | GES.7<br>032  | The EIP<br>resource<br>exceeds the<br>quota limit.                  | The EIP<br>resource<br>exceeds the<br>quota limit.                  | Check whether the EIP resource reaches the quota limit on the VPC page.   |
| 400                | GES.7<br>033  | Invalid EIP ID.   | Invalid EIP<br>ID.  | If the EIP binding type is set to <b>bind_existing</b> , ensure that the EIP ID exists.   |
| 400                | GES.7<br>035  | Invalid region<br>code.   | Invalid<br>region code.   | Enter the correct region code.  |
| 400                | GES.7<br>036  | The target<br>version is<br>earlier than<br>the current<br>version. | The target<br>version is<br>earlier than<br>the current<br>version. | A graph can only be upgraded to a later version.  |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message                                | Description                                     | Solution   |
|--------------------|---------------|---|---|--|
| 400                | GES.7<br>037  | The graph is not in the <b>Stopped</b> state.   | The graph is not in the <b>Stopped</b> state.   | Check whether the graph is in the <b>Stopped</b> state.  |
| 400                | GES.7<br>040  | Failed to back up a graph.                      | Failed to back up a graph.                      | Failed to restore a graph from the backup you select.  |
| 400                | GES.7<br>041  | Insufficient permission.                        | Insufficient permissions.                       | Insufficient permissions.  |
| 400                | GES.7<br>042  | The graph is being created.                     | The graph is being created.                     | The graph is being created.  |
| 400                | GES.7<br>048  | Invalid graph operation.                        | Invalid graph operation.                        | Check whether the value of <b>action_id</b> is <b>start</b> , <b>stop</b> , <b>import-graph</b> , <b>export-graph</b> , <b>clear-graph</b> , or <b>upgrade</b> .                     |
| 400                | GES.7<br>049  | The parameter does not exist.                   | The parameter does not exist.                   | Check whether the request body is consistent with that in the API reference. Mandatory parameters must be set.   |
| 400                | GES.7<br>050  | The parameter is empty.                         | The parameter is empty.                         | Check whether the request body is consistent with that in the API reference. Mandatory parameters must be set.   |
| 400                | GES.7<br>051  | Components at the IaaS layer are faulty.        | Components at the IaaS layer are faulty.        | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, obtain logs and analyze them for fault locating.</li> </ol> |
| 400                | GES.7<br>052  | Invalid CPU architecture of the graph instance. | Invalid CPU architecture of the graph instance. | Check whether the value of <b>arch</b> is set to <b>x86_64</b> or <b>aarch64</b> when the graph is created.  |
| 400                | GES.7<br>054  | The graph is being deleted or has been deleted. | The graph is being deleted or has been deleted. | This error occurs in concurrent deletion scenarios. Generally, a message indicating that the graph does not exist is displayed when you try again later.                             |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message  | Description   | Solution  |
|--------------------|---------------|---|---|---|
| 400                | GES.7<br>056  | The graph of the current flavor cannot be scaled out.                     | The graph of the current flavor cannot be scaled out.                     | Currently, ten-thousand-edge and ten-billion-edge graphs cannot be scaled out. Check whether the graph is a ten-thousand-edge or ten-billion-edge one.                                    |
| 400                | GES.7<br>057  | Invalid graph flavor for scale-out.                                       | Invalid graph flavor for resize.  | <b>graphSizeTypeIndex</b> in the resize request body can be set to <b>2, 3, 4, or 5</b> , indicating the ten-million-edge, hundred-million-edge, billion-edge, or ten-billion-edge graph. |
| 400                | GES.7<br>059  | The IaaS resources of the graph flavor to be scaled out are insufficient. | The IaaS resources of the graph flavor to be scaled out are insufficient. | Check whether the compute resources are sufficient. For details about the IaaS resources required by each GES flavor, see the LLD.  |
| 400                | GES.7<br>061  | Failed to create the ECS because the resources are insufficient.          | Failed to create the ECS because the resources are insufficient.          | Check whether the compute resources are sufficient. For details about the IaaS resources required by each GES flavor, see the LLD.  |
| 400                | GES.7<br>062  | Failed to create the data disk.   | Failed to create the data disk.   | Check the FusionStorage capacity or obtain the detailed error information from the returned <b>errorMessage</b> .   |
| 400                | GES.7<br>063  | Failed to create the system disk.   | Failed to create the system disk.   | Check the FusionStorage capacity or obtain the detailed error information from the returned <b>errorMessage</b> .   |
| 400                | GES.7<br>064  | Failed to create the ECS.   | Failed to create the ECS.   | See the returned <b>errorMessage</b> or download the microservice log on the management plane, search for the ID of failed task in the log, and contact IaaS O&M personnel.               |

| Statu<br>s<br>Code | Error<br>Code | Error<br>Message  | Description   | Solution   |
|--------------------|---------------|---|---|--|
| 400                | GES.7<br>065  | Failed to query the image because the image does not exist. | Failed to query the image because the image does not exist. | Check whether the image ID configured on the GES management plane exists.  |
| 400                | GES.7<br>066  | Failed to query the flavor.                                 | Failed to query the flavor.                                 | Check whether the flavor ID configured on the GES management plane exists.   |
| 400                | GES.7<br>067  | Insufficient ECS quota.                                     | Insufficient ECS quota.                                     | Check whether the flavor ID configured on the GES management plane exists.   |
| 400                | GES.7<br>068  | Invalid request parameters.                                 | Invalid request parameters.                                 | During graph creation, parameters in the request for calling the IaaS API are invalid. Obtain detailed information based on the returned error message for analysis..  |
| 400                | GES.7<br>069  | The metadata file is too large.                             | The metadata file is too large.                             | The metadata file on OBS or that in the request body exceeds 10 MB.  |
| 400                | GES.7<br>070  | Failed to parse the metadata file.                          | Failed to parse the metadata file.                          | When creating metadata, the metadata file on OBS or in the request body does not comply with the metadata standards. Create a metadata file or request body correctly. |

## 9.2.2 Error Codes for Service Plane APIs

If an error occurs in API calling, no result is returned. Identify the cause of error based on the error codes of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you fail to locate the cause of the error, contact technical support and provide the error code for troubleshooting.

**Table 9-3** Error codes

| Statu<br>s<br>Code | Error<br>Code | Error Message  | Description   | Solution   |
|--------------------|---------------|--|---|--|
| 400                | GES.80<br>00  | Incorrect<br>parameter<br>format.                                    | Incorrect<br>parameter<br>format.                                     | Check whether the request<br>body is the same as that<br>described in the document.  |
| 400                | GES.80<br>01  | Failed to query<br>graph statistics.                                 | Failed to<br>query graph<br>statistics.                               | <ol style="list-style-type: none"> <li>1. If the network fluctuates,<br/>try again later.</li> <li>2. If the fault persists, report<br/>the error information in<br/><b>errorMessage</b> to technical<br/>personnel.</li> </ol>                                |
| 500                | GES.80<br>02  | Graph statistics<br>query error.                                     | Graph<br>statistics<br>query error.                                   | <ol style="list-style-type: none"> <li>1. Check whether the token<br/>has expired. If it is expired,<br/>obtain a new one.</li> <li>2. If the fault persists, report<br/>the error information in<br/><b>errorMessage</b> to technical<br/>support.</li> </ol> |
| 400                | GES.80<br>05  | Incorrect<br>parameter.  | Incorrect<br>parameter.   | <ol style="list-style-type: none"> <li>1. Check whether the project<br/>ID in the URL is correct.</li> <li>2. Check whether the request<br/>header is correct, for<br/>example, whether <b>X-Auth-<br/>Token</b> is correct.</li> </ol>                        |
| 400                | GES.80<br>06  | Invalid<br>resource<br>access.                                       | Invalid<br>resource<br>access.  | Check whether the project ID<br>in the URL is correct.   |
| 400                | GES.80<br>07  | Invalid token.   | Invalid<br>token.   | Check whether the token is<br>correct.   |
| 400                | GES.80<br>08  | An error occurs<br>in the<br>underlying<br>authentication<br>system. | An error<br>occurs in the<br>underlying<br>authenticati<br>on system. | Try again later or contact<br>technical support.   |
| 400                | GES.80<br>11  | Failed to<br>export a graph.   | Failed to<br>export a<br>graph.                                       | <ol style="list-style-type: none"> <li>1. Check whether the graph<br/>name is correct.</li> <li>2. Check whether the export<br/>path is correct.</li> <li>3. Check whether the<br/>account has the OBS write<br/>permission.</li> </ol>                        |



| Statu<br>s<br>Code | Error<br>Code | Error Message   | Description   | Solution  |
|--------------------|---------------|---|---|---|
| 400                | GES.80<br>12  | Failed to clear a graph.  | Failed to clear a graph.  | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical support.</li> </ol> |
| 400                | GES.80<br>13  | Failed to incrementally import data to the graph.                 | Failed to incrementally import data to the graph.                                       | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical support.</li> </ol> |
| 400                | GES.80<br>20  | The current user does not have permission.                        | The current user does not have the required permission for granular permission control. | Grant permissions as the Security Administrator.  |
| 400                | GES.81<br>01  | Invalid filter criteria for edge queries.                         | Invalid filter criteria for edge queries.   | Check whether format of the filter criteria for edge queries is correct.  |
| 400                | GES.81<br>02  | Invalid label for edge filtering queries.                         | Invalid label for edge filtering queries.   | Check whether the labels are in JSON format.  |
| 400                | GES.81<br>03  | Both the condition and label of edge filtering queries are empty. | Both the condition and label of edge filtering queries are empty.                       | The condition and label of edge filtering queries cannot be both empty.   |
| 400                | GES.81<br>04  | Invalid edge filtering query sequence.                            | Invalid edge filtering query sequence.  | Check whether the edge filtering query sequence is valid.   |

| Statu<br>s<br>Code | Error<br>Code | Error Message  | Description  | Solution  |
|--------------------|---------------|--|--|---|
| 400                | GES.81<br>05  | Failed to query edges that meet filter criteria.                 | Failed to query edges that meet filter criteria.                 | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.81<br>06  | The source vertex or target vertex in the edge details is empty. | The source vertex or target vertex in the edge details is empty. | Ensure that the source vertex or target vertex in the edge details cannot be empty.   |
| 400                | GES.81<br>07  | Failed to query edge details.                                    | Failed to query edge details.                                    | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 500                | GES.81<br>08  | Edge details query error.  | Edge details query error.  | Try again later or contact technical personnel.   |
| 400                | GES.81<br>09  | Invalid edge filtering query operator.                           | Invalid edge filtering query operator.                           | Ensure that the values of edge filtering query operators are <b>in</b> , <b>out</b> , <b>both</b> , and <b>edge</b> .   |
| 400                | GES.81<br>10  | Parameter edges cannot be left blank.                            | Parameter <b>edges</b> cannot be left blank.                     | Check whether the value of <b>edges</b> in the batch edge query request body is empty.  |
| 400                | GES.82<br>01  | Invalid label for vertex filtering queries.                      | Invalid label for vertex filtering queries.                      | Check whether the labels are in JSON format.  |

| Statu<br>s<br>Code | Error<br>Code | Error Message  | Description   | Solution  |
|--------------------|---------------|--|---|---|
| 400                | GES.82<br>02  | Invalid filter<br>criteria for<br>vertex queries.                                  | Invalid filter<br>criteria for<br>vertex<br>queries.                                  | <ol style="list-style-type: none"> <li>1. Check whether <b>propertyName</b> of the vertex query API is left blank.</li> <li>2. Check whether <b>values</b> of the vertex query API is left blank.</li> <li>3. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.82<br>03  | Both the<br>condition and<br>label of vertex<br>filtering<br>queries are<br>empty. | Both the<br>condition<br>and label of<br>vertex<br>filtering<br>queries are<br>empty. | Ensure that the condition and label of vertex filtering queries are not both empty.   |
| 400                | GES.82<br>04  | Failed to query<br>vertices that<br>meet filter<br>criteria.                       | Failed to<br>query<br>vertices that<br>meet filter<br>criteria.                       | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>   |
| 400                | GES.82<br>05  | Invalid vertex<br>filtering query<br>sequence.                                     | Invalid<br>vertex<br>filtering<br>query<br>sequence.                                  | In the vertex filtering query API, <b>orderValue</b> must be set to <b>incr</b> or <b>decr</b> .  |
| 400                | GES.82<br>06  | Both vertexid<br>and vertexids<br>exist.   | Both<br><b>vertexid</b> and<br><b>vertexids</b><br>exist.                             | <b>vertexid</b> and <b>vertexids</b> cannot coexist.  |
| 400                | GES.82<br>07  | Both vertexid<br>and vertexids<br>are empty.                                       | Both<br><b>vertexid</b> and<br><b>vertexids</b><br>are empty.                         | The <b>vertexid</b> or <b>vertexids</b> parameter is empty.   |
| 400                | GES.82<br>08  | Incorrect<br>vertexids<br>format.  | Incorrect<br><b>vertexids</b><br>format.  | Check whether <b>vertexids</b> is a JSON array.   |
| 400                | GES.82<br>09  | Failed to query<br>vertex details.   | Failed to<br>query vertex<br>details.   | Check whether the graph name exists.  |

| Statu<br>s<br>Code | Error<br>Code | Error Message                            | Description                                     | Solution  |
|--------------------|---------------|--|---|---|
| 500                | GES.82<br>10  | Vertex details query error.              | Vertex details query error.                     | Try again later or contact technical personnel.   |
| 400                | GES.82<br>11  | Invalid vertex filtering query operator. | Invalid vertex filtering query operator.        | Ensure that values of vertex filtering query operators are <b>inV</b> , <b>outV</b> , <b>bothV</b> , and <b>vertex</b> .  |
| 400                | GES.82<br>12  | Failed to delete the vertex label.       | Failed to delete the vertex label.              | Check whether the label exists.   |
| 400                | GES.82<br>13  | Failed to add the vertex label.          | Failed to add the vertex label.                 | Check whether the label exists.   |
| 400                | GES.82<br>14  | Parameter vertices cannot be left blank. | Parameter <b>vertices</b> cannot be left blank. | Check whether the value of vertices in the batch vertex query request body is empty.  |
| 400                | GES.82<br>20  | Failed to update the vertex properties.  | Failed to update the vertex properties.         | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.82<br>21  | Failed to update the edge properties.    | Failed to update the edge properties.           | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.83<br>01  | Failed to query a job.                   | Failed to query a job.                          | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 500                | GES.83<br>02  | Job query error.                         | Job query error.                                | Try again later or contact technical personnel.   |

| Statu<br>s<br>Code | Error<br>Code | Error Message                                  | Description                                    | Solution  |
|--------------------|---------------|--|--|---|
| 400                | GES.83<br>03  | Failed to terminate a job.                     | Failed to terminate a job.                     | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>   |
| 500                | GES.83<br>04  | Job termination error.                         | Job termination error.                         | Try again later or contact technical personnel.   |
| 400                | GES.84<br>01  | The algorithm or graph name cannot be empty.   | The algorithm or graph name cannot be empty.   | Ensure that the algorithm or graph name in not empty.   |
| 400                | GES.84<br>02  | Failed to run the algorithm.                   | Failed to run the algorithm.                   | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. Check whether the graph name in the algorithm running API is correct.</li> <li>3. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 500                | GES.84<br>03  | Algorithm running error.                       | Algorithm running error.                       | Try again later or contact technical personnel.   |
| 400                | GES.84<br>04  | Invalid algorithm running format.              | Invalid algorithm running format.              | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>   |
| 400                | GES.85<br>01  | The Gremlin command is not supported.          | The Gremlin command is not supported.          | Replace the unsupported Gremlin statements: tryNext, explain, and tree.   |
| 400                | GES.85<br>02  | Failed to find the Gremlin configuration file. | Failed to find the Gremlin configuration file. | Try again later or contact technical personnel.   |

| Statu<br>s<br>Code | Error<br>Code | Error Message   | Description   | Solution   |
|--------------------|---------------|---|---|--|
| 400                | GES.85<br>03  | Gremlin query failed.   | Gremlin query failed.   | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>                                |
| 500                | GES.85<br>04  | Gremlin query error.  | Gremlin query error.  | Try again later or contact technical personnel.  |
| 400                | GES.85<br>05  | The Gremlin query statement does not contain the command field.           | The Gremlin query statement does not contain the command field.           | Ensure that the Gremlin query statement does not contain the command field.  |
| 400                | GES.85<br>06  | The size of the Gremlin query request statements exceeds the upper limit. | The size of the Gremlin query request statements exceeds the upper limit. | The current limit is 64 MB.  |
| 500                | GES.86<br>01  | Gremlin service unavailable.  | Gremlin service unavailable.  | Try again later or contact technical personnel.  |
| 500                | GES.86<br>02  | Engine service unavailable.   | Engine service unavailable.   | Try again later or contact technical personnel.  |
| 400                | GES.86<br>03  | Failed to create an index   | Failed to create an index   | <ol style="list-style-type: none"> <li>1. Check whether the index name contains only letters, digits, hyphens (-), and underscores (_).</li> <li>2. Check whether the index parameter type complies with that specified by the API.</li> </ol> |

| Statu<br>s<br>Code | Error<br>Code | Error Message  | Description   | Solution   |
|--------------------|---------------|--|---|--|
| 400                | GES.86<br>04  | Failed to delete<br>an index   | Failed to<br>delete an<br>index   | <ol style="list-style-type: none"> <li>1. Check whether the graph name is correct.</li> <li>2. Check whether the index name is correct.</li> <li>3. Check whether <b>Method type</b> of the request is <b>delete</b>.</li> </ol>   |
| 400                | GES.86<br>05  | Failed to query<br>an index  | Failed to<br>query an<br>index  | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>  |
| 400                | GES.86<br>09  | The request<br>body for<br>querying path<br>details is<br>invalid.                             | The request<br>body for<br>querying<br>path details<br>is invalid.                                    | <ol style="list-style-type: none"> <li>1. Check whether the graph name is correct.</li> <li>2. Check whether the parameter format of the API for querying path details is correct.</li> <li>3. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>   |
| 400                | GES.86<br>10  | The path<br>parameter of<br>the request<br>body for<br>querying path<br>details is<br>invalid. | The <b>path</b><br>parameter of<br>the request<br>body for<br>querying<br>path details<br>is invalid. | <ol style="list-style-type: none"> <li>1. Check whether the parameter format of the API for querying path details is correct.</li> <li>2. Check whether the mandatory parameters of the API for querying path details are set.</li> <li>3. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol> |
| 400                | GES.86<br>11  | Failed to query<br>path details.   | Failed to<br>query path<br>details.   | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>  |

| Statu<br>s<br>Code | Error<br>Code | Error Message  | Description  | Solution   |
|--------------------|---------------|--|--|--|
| 400                | GES.86<br>12  | The operation of querying path details is not supported. | The operation of querying path details is not supported. | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>  |
| 400                | GES.88<br>01  | Failed to add a label to metadata.                       | Failed to add a label to metadata.                       | <ol style="list-style-type: none"> <li>1. Check whether the label to be added already exists.</li> <li>2. Check whether the format of the parameter for adding the label is correct.</li> <li>3. Check whether the mandatory parameters for adding the label are set.</li> </ol> |
| 400                | GES.88<br>03  | Failed to query the metadata.                            | Failed to query the metadata.                            | <ol style="list-style-type: none"> <li>1. Check whether the graph to be queried exists.</li> <li>2. Check whether the value of <b>graph_name</b> in the API for querying graph metadata is correct.</li> </ol>   |
| 500                | GES.88<br>04  | Metadata query error.                                    | Metadata query error.                                    | Try again later or contact technical personnel.  |
| 400                | GES.88<br>06  | K-Hop query with filter criteria failed.                 | K-Hop query with filter criteria failed.                 | <ol style="list-style-type: none"> <li>1. If the network fluctuates, try again later.</li> <li>2. If the fault persists, report the error information in <b>errorMessage</b> to technical personnel.</li> </ol>  |

## 9.3 Obtaining a Project ID

### Obtaining a Project ID by Calling an API

You can obtain a project ID by calling an API. The API for obtaining a project ID is **GET <https://{Endpoint}/v3/projects>**, where *{Endpoint}* indicates the IAM endpoint.

For details about API authentication, see [Authentication](#).

The following is an example response. The value of **id** under **projects** is the project ID. The following is an example response. For example, if GES is deployed in the *xxx* region, search for the **name** whose value is *xxx* in the response body, and the value of **id** under **projects** is the project ID.



```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

## Obtaining a Project ID from the Console

A project ID is required for some URIs when an API is called.

To obtain a project ID, perform the following operations:

1. Register an account and log in to the console.
2. In the upper right corner of the page, click the username and choose **My Credentials** from the drop-down list. The **My Credentials** page is displayed.
3. On the **API Credentials** page, view the project ID and name in the **Projects** area.

If there are multiple projects, unfold the target region and obtain the project ID from the **Project ID** column.

## 9.4 Obtaining the Account Name and Account ID

An account ID is required for some requests for calling APIs. To obtain an account ID, perform the following operations:

1. Log in to the management console.
2. Hover the cursor on the username and select **My Credentials** from the drop-down list.

On the **API Credentials** page, view the account name and ID.