



Graph Engine Service

API Reference

Date 2022-05-30

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Constraints and Limitations on Using GES.....	1
1.3.1 Constraints of Using Service Plane APIs.....	2
1.3.2 OBS Object Name Restrictions.....	2
1.4 Concepts.....	3
1.5 Selecting an API Type or Version.....	3
1.6 Hundred-Billion-Edge Graph.....	3
2 Calling APIs.....	6
2.1 Making an API Request.....	6
2.1.1 Making a Management Plane API Request.....	6
2.1.2 Making a Service Plane API Request.....	9
2.2 Response.....	10
3 Management Plane APIs.....	12
3.1 System Management APIs.....	12
3.1.1 Viewing Quotas	12
3.2 Graph Management APIs.....	14
3.2.1 Querying the Graph List.....	14
3.2.2 Querying Graph Details.....	21
3.2.3 Creating a Graph.....	27
3.2.4 Stopping a Graph.....	34
3.2.5 Starting a Graph.....	36
3.2.6 Deleting a Graph.....	38
3.2.7 Incrementally Importing Data to Graphs.....	41
3.2.8 Exporting a Graph.....	45
3.2.9 Clearing a Graph.....	48
3.2.10 Upgrading a Graph.....	50
3.2.11 Binding an EIP.....	52
3.2.12 Unbinding an EIP.....	54
3.2.13 Resizing a Graph.....	56
3.2.14 Restarting a Graph.....	59

3.2.15 Expanding a Graph.....	61
3.3 Backup Management APIs.....	63
3.3.1 Viewing the List of All Backups.....	63
3.3.2 Viewing the Backup List of a Graph.....	67
3.3.3 Adding a Backup.....	71
3.3.4 Deleting a Backup	73
3.4 Metadata Management APIs.....	74
3.4.1 Constraints.....	74
3.4.2 Querying the Metadata List	77
3.4.3 Querying Metadata	79
3.4.4 Adding Metadata	82
3.4.5 Deleting Metadata	86
3.5 Task Center APIs.....	87
3.5.1 Querying Job Status on the Management Plane	87
3.5.2 Querying Job Details in the Job Center.....	92
4 Service Plane APIs.....	98
4.1 Vertex Operation APIs.....	98
4.1.1 Querying Vertices That Meet Filter Criteria.....	98
4.1.2 Querying Vertex Details.....	102
4.1.3 Adding a Vertex.....	104
4.1.4 Deleting a Vertex.....	106
4.1.5 Updating Vertex Properties.....	108
4.1.6 Querying Vertex Data in Batches.....	111
4.1.7 Adding Vertices in Batches.....	113
4.1.8 Deleting Vertices in Batches.....	116
4.1.9 Updating Vertex Properties in Batches.....	117
4.1.10 Adding a Vertex Label.....	121
4.1.11 Deleting a Vertex Label.....	122
4.1.12 Exporting Filtered Vertices.....	124
4.1.13 Deleting Filtered Vertices.....	127
4.2 Edge Operation APIs.....	129
4.2.1 Querying Edges That Meet Filter Criteria.....	129
4.2.2 Querying Edge Details.....	132
4.2.3 Adding an Edge.....	135
4.2.4 Deleting an Edge.....	138
4.2.5 Updating Edge Properties.....	141
4.2.6 Querying Edge Data in Batches.....	144
4.2.7 Adding Edges in Batches.....	147
4.2.8 Deleting Edges in Batches	151
4.2.9 Updating Edge Properties in Batches	154
4.2.10 Exporting Filtered Edges	157
4.2.11 Deleting Filtered Edges	160

4.3 Metadata Operation APIs.....	162
4.3.1 Adding a Label.....	162
4.3.2 Updating a Label.....	166
4.3.3 Query Labels (for 100-billion-edge graphs only).....	169
4.3.4 Querying Graph Metadata Details.....	171
4.3.5 Changing Property Names in Batches.....	175
4.3.6 Deleting a Label.....	177
4.3.7 Adding Labels in Batches.....	179
4.3.8 Adding Properties (for 100-billion-edge graphs only).....	183
4.3.9 Query Properties (for 100-billion-edge graphs only).....	185
4.3.10 Querying the Property List (for 100-billion-edge graphs only).....	187
4.3.11 Querying Schema Structure.....	189
4.3.12 Generating a Schema Structure.....	192
4.4 Index Operation APIs.....	193
4.4.1 Creating an Index.....	193
4.4.2 Deleting an Index.....	197
4.4.3 Querying Indexes.....	199
4.5 Gremlin Operation APIs.....	201
4.5.1 Executing Gremlin Queries	201
4.6 Algorithm APIs.....	204
4.6.1 Running Algorithms.....	204
4.6.2 Algorithm API Parameter References.....	206
4.6.2.1 Common Algorithm Parameters.....	206
4.6.2.2 PageRank.....	212
4.6.2.3 PersonalRank.....	213
4.6.2.4 K-core.....	214
4.6.2.5 K-hop	214
4.6.2.6 Shortest Path.....	215
4.6.2.7 All Shortest Paths	217
4.6.2.8 Filtered Shortest Path.....	218
4.6.2.9 SSSP.....	221
4.6.2.10 Shortest Path of Vertex Sets.....	221
4.6.2.11 n-Paths.....	223
4.6.2.12 Closeness Centrality.....	224
4.6.2.13 Label Propagation.....	224
4.6.2.14 Louvain.....	226
4.6.2.15 Link Prediction.....	227
4.6.2.16 Node2vec.....	227
4.6.2.17 Real-time Recommendation.....	229
4.6.2.18 Common Neighbors.....	231
4.6.2.19 Connected Component.....	232
4.6.2.20 Degree Correlation.....	232

4.6.2.21 Triangle Count.....	232
4.6.2.22 Cluster Coefficient.....	233
4.6.2.23 Common Neighbors of Vertex Sets.....	233
4.6.2.24 All Shortest Paths of Vertex Sets.....	235
4.6.2.25 Filtered Circle Detection (2.2.15).....	236
4.6.2.26 Subgraph Matching (2.2.16).....	239
4.6.2.27 Filtered All Pairs Shortest Paths (2.2.17).....	240
4.6.2.28 Filtered All Shortest Paths (2.2.17).....	242
4.6.2.29 Topicrank (2.2.20).....	244
4.6.2.30 Filtered n-Paths (2.2.22).....	245
4.7 Path APIs.....	248
4.7.1 Querying Path Details.....	249
4.8 Graph Statistics APIs.....	252
4.8.1 Querying General Information About a Graph.....	252
4.8.2 Querying the Graph Version.....	257
4.9 Subgraph Operation APIs.....	259
4.9.1 Querying a Subgraph.....	259
4.9.2 Executing an Algorithm on a Subgraph.....	261
4.10 Job Management APIs.....	264
4.10.1 Querying Job Status on the Service Plane.....	265
4.10.2 Canceling a Job.....	268
4.10.3 Exporting Job Execution Results to Files.....	270
4.10.4 Querying the Job List.....	273
4.11 Querying K Hop Vertices or Edges Using a Filter.....	276
4.12 Updating Specified Properties of Vertices and Edges by Importing a File.....	288
4.13 Deleting Vertices and Edges by Files.....	293
4.14 Performing Cypher Queries.....	296
4.15 Granular Permission Control APIs.....	311
4.15.1 Authorization.....	311
4.15.2 Canceling Authorization.....	315
4.15.3 Querying Authorization.....	317
5 GES Metrics.....	320
6 Appendix.....	327
6.1 Status Codes.....	327
6.2 Error Codes.....	331
6.2.1 Error Codes for Management Plane APIs.....	331
6.2.2 Error Codes for Service Plane APIs.....	338
6.3 Obtaining a Project ID.....	347
6.4 Obtaining an Account ID.....	348

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 Constraints of Using Service Plane APIs

Two methods are available for you to access the service plane APIs.

- 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 OBS Object Name Restrictions

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

Letters and digits	0-9, a-z, A-Z
Special characters	! - _ . * ' ()
Other characters	\u4e00-\u9fa5

The following characters are not supported:

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

1.4 Concepts

- **User**
A user is created in 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 Selecting an API Type or Version

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

1.6 Hundred-Billion-Edge Graph

API

The following APIs are supported for graphs of the hundred-billion-edge type:

1. [Querying Vertex Details](#)
2. [Querying Edge Details](#)
3. [Querying Vertex Data in Batches](#)
4. [Querying Edge Data in Batches](#)
5. [Adding Vertices in Batches](#)
6. [Adding Edges in Batches](#)

7. [Updating Vertex Properties in Batches](#)
8. [Updating Edge Properties in Batches](#)
9. [Deleting Vertices in Batches](#)
10. [Deleting Edges in Batches](#)
11. [Querying Job Status on the Service Plane](#)
12. [Querying the Job List](#)
13. [Querying Graph Metadata Details](#)
14. [Adding Properties](#)
15. [Querying a Property](#)
16. [Querying the Property List](#)
17. [Adding a Label](#)
18. [Updating a Label](#)
19. [Querying Labels](#)
20. [Creating an Index](#)
21. [Deleting an Index](#)
22. [Querying Indexes](#)
23. [Querying General Information About a Graph](#)
24. [Querying the Graph Version](#)
25. [Shortest Path](#)
26. [Cypher Query](#)

Supported Data Types

Type	Description
char	Character
float	Float type (32-bit float)
double	Double floating point type (64-bit float point)
bool	Boolean type. Available values are 0/1 and true/false .
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"> • YYYY-MM-DD HH:MM:SS • YYYY-MM-DD <p>NOTE The value of MM or DD must consist of two digits. If the day or month number contains only one digit, add 0 before it, for example, 05/01.</p>
string	Variable-length string

Data Import Restrictions

For details about how to import data, see .

The restrictions on importing hundred-billion-edge graphs are as follows:

- **Importing Data Concurrently**
Multiple data files can be imported at the same time. To improve the import speed, split a large file into 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 hundred-billion-edge 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.

2 Calling APIs

2.1 Making an API Request

2.1.1 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 2-1 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 application/json . 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 https requests is port 443 .	code.test.com or 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 supporting token-based authentication, cloud APIs also support authentication using access key ID/secret access key (AK/SK). 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.

2.1.2 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 [Constraints of 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 2-2 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 application/json . 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 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 *xxxxxxxxxxxxxxxxxxx* (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.

2.2 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 2-3](#) list the response header parameters.

Table 2-3 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.

3 Management Plane APIs

3.1 System Management APIs

3.1.1 Viewing Quotas

Function

This API is used to query tenant quotas.

URI

GET /v1.0/{project_id}/graphs/quotas

Table 3-1 URI parameters

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

Request Parameters

Table 3-2 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-3 Response body parameter

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.
quotas	GesQuotaResp object	Resource type list. This field is left blank when the request fails.

Table 3-4 GesQuotaResp

Parameter	Type	Description
resources	Array of Quota objects	GES resource quota list

Table 3-5 Quota

Parameter	Type	Description
type	String	Resource type. Available values are as follows: <ul style="list-style-type: none"> • "graph" • "backup" • "metadata"
available	Integer	Number of available graphs
edgeVolume	Integer	Number of available edges. The parameter value is valid only when type is "graph".

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	OK

3.2 Graph Management APIs

3.2.1 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 3-6 URI parameters

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

Table 3-7 Query parameters

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

Request Parameters

Table 3-8 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-9 Response body parameter

Parameter	Type	Description
graphCount	Integer	Total number of graphs. This parameter is left blank when the request fails.
graphs	Array of graph_1 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 3-10 graph_1

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
createdBy	String	Account of the graph creator
isMultiAz	String	Whether to enable cross-AZ HA
regionCode	String	Region code
azCode	String	AZ code
schemaPath	Array of schemaPath_1 objects	Path for storing the metadata file
edgesetPath	Array of edgesetPath_1 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 vertexsetPath_1 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 SysTagsRes objects	Enterprise project information. If this parameter is not specified, this function is disabled (default).

Parameter	Type	Description
status	String	Status code of a graph: <ul style="list-style-type: none"> ● 100: Indicates that a graph is being prepared. ● 200: indicates that a graph is running. ● 201: indicates that a graph is upgrading. ● 202: indicates that a graph is being imported. ● 203: indicates that a graph is being rolled back. ● 204: indicates that a graph is being exported. ● 205: indicates that a graph is being cleared. ● 206: indicates that the system is preparing for resize. ● 207: indicates that the resize is in progress. ● 208: Indicates that the resize is being rolled back. ● 300: indicates that a graph is faulty. ● 303: indicates that a graph fails to be created. ● 400: indicates that a graph is deleted. ● 900: indicates that a graph is stopped. ● 901: indicates that a graph is being stopped. ● 920: indicates that a graph is being started.
actionProgress	String	Progress of graph creation in percentage
graphSizeTypeIndex	String	Graph size type index: <ul style="list-style-type: none"> ● 0: indicates 10 thousand edges. ● 1: indicates 1 million edges. ● 2: indicates 10 million edges. ● 3: indicates 100 million edges. ● 4: indicates 1 billion edges. ● 5: indicates 10 billion edges. ● 6: indicates 100 billion edges. ● 401: indicates 1 billion enhanced edges.
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 1 .
created	String	Time when a graph is created
updated	String	Time when a graph is updated
privatelp	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	List<String>	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"> • generalCipher: International algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enableHttps	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.
tags	JsonArray	Tag list. Each tag is in <key,value> format.

Table 3-11 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"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-12 edgesetPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint

Parameter	Type	Description
status	String	OBS file import status: <ul style="list-style-type: none"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-13 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"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-14 SysTagsRes

Parameter	Type	Description
key	String	Key of the enterprise project. The value is _sys_enterprise_project_id .
value	String	Enterprise project ID. You can obtain it from the enterprise project.

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": "az1-dc1",
      "azCode": "az1-dc1a",
      "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": "az1-dc1",
    "azCode": "az1-dc1a",

    "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",
    "vpclId": "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 Code

Status Code	Description
200	OK

3.2.2 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 3-15 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 3-16 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-17 Response body parameter

Parameter	Type	Description
graph	graph_1 object	Graph object. If the request fails, this parameter is left empty.
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 3-18 graph_1

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
createdBy	String	Account of the graph creator
isMultiAz	String	Whether to enable cross-AZ HA
regionCode	String	Region code
azCode	String	AZ code
schemaPath	Array of schemaPath_1 objects	Path for storing the metadata file
edgesetPath	Array of edgesetPath_1 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 vertexsetPath_1 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

Parameter	Type	Description
dataStoreVersion	String	Graph version
sys_tags	Array of SysTagsRes objects	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"> ● 100: Indicates that a graph is being prepared. ● 200: indicates that a graph is running. ● 201: indicates that a graph is upgrading. ● 202: indicates that a graph is being imported. ● 203: indicates that a graph is being rolled back. ● 204: indicates that a graph is being exported. ● 205: indicates that a graph is being cleared. ● 206: indicates that the system is preparing for resize. ● 207: indicates that the resize is in progress. ● 208: Indicates that the resize is being rolled back. ● 300: indicates that a graph is faulty. ● 303: indicates that a graph fails to be created. ● 400: indicates that a graph is deleted. ● 900: indicates that a graph is stopped. ● 901: indicates that a graph is being stopped. ● 920: indicates that a graph is being started.
actionProgress	String	Progress of graph creation in percentage
graphSizeTypeIndex	String	Graph size type index: <ul style="list-style-type: none"> ● 0: indicates 10 thousand edges. ● 1: indicates 1 million edges. ● 2: indicates 10 million edges. ● 3: indicates 100 million edges. ● 4: indicates 1 billion edges. ● 5: indicates 10 billion edges. ● 401: indicates 1 billion enhanced edges.
vpcId	String	VPC ID

Parameter	Type	Description
subnetId	String	Subnet ID in the VPC
securityGroupId	String	Security group ID
replication	Integer	Number of replicas. The default value is 1 .
created	String	Time when a graph is created
updated	String	Time when a graph is updated
privateIp	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	List<String>	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"> • generalCipher: International algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enableHttps	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.
tags	JsonArray	Tag list. Each tag is in <key,value> format.

Table 3-19 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"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-20 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"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-21 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"> • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 3-22 SysTagsRes

Parameter	Type	Description
key	String	Key of the enterprise project. The value is _sys_enterprise_project_id .
value	String	Enterprise project ID. You can obtain it from the enterprise project.

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": "az1-dc1",
  }
}
```

```

    "azCode": "az1-dc1a",
    "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",
    "vpcId": "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"
  }
}
{
  "graph": {
    "id": "f1529b88-c958-493e-8452-fccfe932cde1",
    "name": "demo",
    "regionCode": "az1-dc1",
    "azCode": "az1-dc1a",
    "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",
    "vpcId": "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",
    "publicIp": "49.4.81.183",
    "dataStoreVersion": "1.0.5"
  }
}

```

Status Code

Status Code	Description
200	OK

3.2.3 Creating a Graph

Function

This API is used to create a graph.

URI

POST /v1.0/{project_id}/graphs

Table 3-23 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Request Parameters

Table 3-24 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-25 Request body parameters

Parameter	Mandatory	Type	Description
graph	Yes	graph object	Graph type

Table 3-26 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"> ● 0: indicates 10 thousand edges. ● 1: indicates 1 million edges. ● 2: indicates 10 million edges. ● 3: indicates 100 million edges. ● 4: indicates 1 billion edges. ● 5: indicates 10 billion edges. ● 6: indicates 100 billion edges. ● 401: indicates 1 billion enhanced edges.
arch	No	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 . The default value is x86_64 . <ul style="list-style-type: none"> ● x86_64: X64 64-bit architecture ● aarch64: Arm 64-bit architecture
dataSource	No	dataSource object	Data source. This field is not set when you create an empty graph.
vpcId	Yes	String	VPC ID
subnetId	Yes	String	Subnet ID in the VPC
securityGroupId	Yes	String	Security group ID
publicIp	No	publicIp object	Public IP address. If the parameter is not specified, public connection is not used by default.

Parameter	Mandatory	Type	Description
enableMultiAZ	No	Boolean	Whether the created graph supports the cross-AZ mode. The default value is false . If the value is true , the system will create the ECSs in the graph in two AZs. If this parameter is not specified when you create a graph, all ECSs in the graph are created in one AZ.
encryption	No	encryptionReq 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 SysTagsRes objects	Enterprise project information. If this parameter is not specified, this function is disabled (default).
tags	No	Array of SysTagsRes objects	TMS tags for expenses. This function is disabled by default.
enableRBAC	No	Boolean	Whether to enable granular permission control for the created graph. The default value is false , indicating that granular permission control is disabled. If this parameter is set to true , 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"> • generalCipher: International algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)

Parameter	Mandatory	Type	Description
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 3-27 dataSource

Parameter	Mandatory	Type	Description
type	Yes	String	Data source type. Currently, only OBS is supported.
parameters	Yes	object	Data source parameters.

Table 3-28 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. The CSV format is used by default.
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. The CSV format is used by default.

Parameter	Mandatory	Type	Description
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	parallelEdge object	Repetitive edge processing

 NOTE

- For details about the value validity of the **schemaPath**, **edgesetPath**, **vertexsetPath**, and **logDir** character strings, see the [OBS Object Name Restrictions](#).

Table 3-29 parallelEdge

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

Table 3-30 publicIp

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

Table 3-31 encryptionReq

Parameter	Mandatory	Type	Description
enable	No	Boolean	Whether to enable the encryption feature. The value can be true or false . The default value is false .
masterKeyId	No	String	ID of the user master key created by the Data Encryption Workshop (DEW) in the project where the graph is created.

Table 3-32 SysTagsRes

Parameter	Mandatory	Type	Description
key	No	String	Key of the enterprise project. Enter _sys_enterprise_project_id .
value	No	String	Enterprise project ID. You can obtain it from the enterprise project.

Response Parameters

Status code: 200

Table 3-33 Response body parameters

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
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.

Example Request

```
POST https://Endpoint/v1.0/{project_id}/graphs
{
  "graph":{
    "name":"demo",
    "dataSource": {
      "type": "OBS",
      "parameters": {
        "schemaPath": "hkmovie/testcre/schema.xml",
        "edgesetPath": "hkmovie/testcre/edges/edge.csv",
        "edgesetFormat": "csv",
        "edgesetDefaultLabel": "",
        "vertexsetPath": "hkmovie/testcre/vetexs/vertex.csv",
        "vertexsetFormat": "csv",
        "vertexsetDefaultLabel": "",
        "logDir": "hkmovie/logdir",
        "parallelEdge": {
          "action":"override",
          "ignoreLabel":true
        }
      }
    },
    "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"
    }
  }
}
```

Example Responses

Status code: 200

OK

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

Returned Values

Status Code	Description
200	OK

3.2.4 Stopping 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 /v1.0/{project_id}/graphs/{graph_id}/action

Table 3-34 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-35 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> • stop

Request Parameters

Table 3-36 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-37 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 graph stopping job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

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	OK
400	Bad Request

3.2.5 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 3-38 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-39 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> • start

Request Parameters

Table 3-40 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-41 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 Adding a Backup .

Response Parameters

Status code: 200

Table 3-42 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.

Parameter	Type	Description
jobId	String	ID of the graph startup job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

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	OK
400	Bad Request

3.2.6 Deleting a Graph

Function

This API is used to delete a graph.

URI

DELETE /v1.0/{project_id}/graphs/{graph_id}

Table 3-43 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-44 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 3-45 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-46 Response body parameter

Parameter	Type	Description
jobId	String	ID of the graph deletion job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .
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.

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	OK
400	Bad Request

3.2.7 Incrementally Importing Data to Graphs

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 3-47 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-48 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> import-graph

Request Parameters

Table 3-49 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-50 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
vertexsetFormat	No	String	Format of the vertex data set. Currently, only the CSV format is supported. 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 This parameter is not supported for graphs of the 100-billion-edge type.
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . <ul style="list-style-type: none"> • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten. This parameter is not supported for graphs of the 100-billion-edge type.

Parameter	Mandatory	Type	Description
ignoreLabel	No	Boolean	<p>Whether to ignore labels on repetitive edges. The value is true or false, and the default value is true.</p> <ul style="list-style-type: none"> • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.
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 true or false, and the default value is false.</p> <ul style="list-style-type: none"> • true: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import. • false: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.

 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 3-51 Response body parameter

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	Indicates the ID of an asynchronous job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

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	OK
400	Bad Request

3.2.8 Exporting a Graph

Function

This API is used to export a graph.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 3-52 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-53 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> export-graph

Request Parameters

Table 3-54 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-55 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 3-56 Response body parameter

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.

Parameter	Type	Description
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

 **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	OK
400	Bad Request

3.2.9 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 3-57 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-58 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> clear-graph
clear-metadata	No	Boolean	Whether to clear the metadata associated with the graph. You are advised to clear it.

Request Parameters

Table 3-59 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-60 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 an asynchronous job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

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	OK
400	Bad Request

3.2.10 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 3-61 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-62 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> upgrade

Request Parameters

Table 3-63 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-64 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 true or false, and the default value is false.</p> <ul style="list-style-type: none"> • true: forcible upgrades, which will interrupt running tasks, such as long algorithm execution tasks. As a result, a small number of requests may fail. • false: non-forcible upgrades, which will wait for running services to complete. The upgrade process may be slow.

Response Parameters

Status code: 200

Table 3-65 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	<p>ID of an asynchronous job</p> <p>NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.</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	OK
400	Bad Request

3.2.11 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 3-66 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-67 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> • bindEip

Request Parameters

Table 3-68 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-69 Request body parameters

Parameter	Mandatory	Type	Description
eipId	Yes	String	ID of the elastic IP address For details about how to query the EIP ID, see .

Response Parameters

Status code: 200

Table 3-70 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.

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	OK
400	Bad Request

3.2.12 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 3-71 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-72 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none"> unbindEip

Request Parameters

Table 3-73 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-74 Request body parameters

Parameter	Mandatory	Type	Description
eipId	Yes	String	ID of the elastic IP address. For details about how to query the EIP ID, see .

Response Parameters

Status code: 200

Table 3-75 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.

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	OK
400	Bad Request

3.2.13 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 3-76 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_id	Yes	String	Graph ID

Request Parameters

Table 3-77 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-78 Request body parameters

Parameter	Mandatory	Type	Description
resize	Yes	GraphSizeTypeIndexReq object	Resize is an object.

Table 3-79 GraphSizeTypeIndexReq

Parameter	Mandatory	Type	Description
graphSizeTypeIndex	Yes	String	Graph flavor type. Currently, the value can be 2, 3, 4, or 5 , indicating that a graph can be scaled out to support 10 million, 100 million, 1 billion, or 10 billion edges, respectively. (graph_size_type_index)
graph_size_type_index	Yes	String	Graph flavor type. Currently, the value can be 2, 3, 4, or 5 , indicating that a graph can be scaled out to support 10 million, 100 million, 1 billion, or 10 billion edges, respectively.

Response Parameters

Status code: 200

Table 3-80 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	Indicates the ID of the resize job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

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	OK
400	Bad Request

3.2.14 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 3-81 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-82 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: <ul style="list-style-type: none">• restart

Request Parameters

Table 3-83 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-84 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 a forcible restart job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

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	OK
400	Bad Request

3.2.15 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 3-85 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 3-86 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-87 Request body parameters

Parameter	Mandatory	Type	Description
expand	Yes	ReplicationReq object	expand is an object.

Table 3-88 ReplicationReq

Parameter	Mandatory	Type	Description
replication	Yes	String	Number of new nodes to expand

Response Parameters

Status code: 200

Table 3-89 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.

Parameter	Type	Description
jobId	String	ID of the expansion job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

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	OK
400	Bad Request

3.3 Backup Management APIs

3.3.1 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 3-90 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Table 3-91 Query parameters

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

Request Parameters

Table 3-92 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-93 Response body parameter

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.

Parameter	Type	Description
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.
backupCount	Integer	Total number of backups. This parameter is left blank when the request fails.
backupList	Array of backup objects	List of all backups under the current project ID. This parameter is left blank when the request fails.

Table 3-94 backup

Parameter	Type	Description
id	String	Backup ID
name	String	Backup name
backupMethod	String	Backup method. The value can be auto or manual .
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"> ● backing_up: indicates that a graph is being backed up. ● success: indicates that a graph is successfully backed up. ● failed: indicates that a graph fails to be backed up.
startTimestamp	Long	Start timestamp of a backup job
startTime	String	Backup start time

Parameter	Type	Description
endTimeStamp	Long	End timestamp of a backup job
endTime	String	Backup end time
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)
encrypted	Boolean	Whether to encrypt backup data. The value true indicates that the backup data is encrypted. The default value false indicates that the backup data is not encrypted.

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	OK
400	Bad Request

3.3.2 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 3-95 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 3-96 Query parameters

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

Request Parameters

Table 3-97 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-98 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.

Parameter	Type	Description
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.
backupCount	Integer	Total number of backups. This field is left blank when the request fails.
backupList	Array of backup objects	List of backups of the specified graph under the current project. This parameter is left blank if the request fails.

Table 3-99 backup

Parameter	Type	Description
id	String	Backup ID
name	String	Backup name
backupMethod	String	Backup method. The value can be auto or manual .
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"> ● backing_up: indicates that a graph is being backed up. ● success: indicates that a graph is successfully backed up. ● failed: indicates that a graph fails to be backed up.
startTimestamp	Long	Start timestamp of a backup job
startTime	String	Backup start time

Parameter	Type	Description
endTimeStamp	Long	End timestamp of a backup job
endTime	String	Backup end time
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)
encrypted	Boolean	Whether to encrypt backup data. The value true indicates that the backup data is encrypted. The default value false indicates that the backup data is not encrypted.

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	OK
400	Bad Request

3.3.3 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 3-100 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 3-101 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-102 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 graph backup job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

POST https://Endpoint/v1.0/{project_id}/graphs/{graph_id}/backups

Example Response

Example response with status code 200:

OK

```
{
  "jobId" : "ff8080815f9a3c84015f9a438ff70001"
}
```

Status code: 400

Bad Request

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

Status Code

Status Code	Description
200	OK
400	Bad Request

3.3.4 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 3-103 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
backup_id	Yes	String	Graph backup ID
graph_id	Yes	String	Graph ID

Request Parameters

Table 3-104 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-105 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.

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	OK
400	Bad Request

3.4 Metadata Management APIs

3.4.1 Constraints

[Table 3-106](#) and [Table 3-107](#) list the metadata types.

Table 3-106 Metadata property constraints

Data Type	Constraints
char	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
char array	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
float	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
double	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
bool	<ul style="list-style-type: none"> • Equal to (=) • Not equal to (!=)

Data Type	Constraints
long	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
int	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
date	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
enum	<ul style="list-style-type: none"> • Equal to (=) • Not equal to (!=)
string	<ul style="list-style-type: none"> • Less than (<) • Greater than (>) • Equal to (=) • Not equal to (!=) • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)

Table 3-107 Property-level constraints

Property Level	Constraints	Description
Single value/ Multiple values	has	This property is contained.

Property Level	Constraints	Description
Single value/ Multiple values	hasNot	This property is not contained.

3.4.2 Querying the Metadata List

Function

This API is used to query the metadata list.

URI

GET /v1.0/{project_id}/graphs/metadatas

Table 3-108 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Table 3-109 Query parameters

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

Request Parameters

Table 3-110 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-111 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.
schemaCount	Integer	Number of returned metadata files. This parameter is left blank if the request fails.
schemaList	Array of metadata objects	List of all metadata files under the current project ID. This parameter is left blank if the request fails.

Table 3-112 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.
createTimestamp	String	Metadata creation timestamp
lastUpdateTimestamp	String	Last timestamp when the metadata is upgraded

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	OK
500	Internal Server Error

3.4.3 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 3-113 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 3-114 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-115 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.
gesMetadata	GesMetaDat a object	Object for storing metadata message information.

Table 3-116 GesMetaData

Parameter	Type	Description
labels	Array of Label objects	Label data structure set

Table 3-117 Label

Parameter	Type	Description
name	String	Label name
properties	Object	Property map

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",

```


URI

POST /v1.0/{project_id}/graphs/metadatas

Table 3-118 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Request Parameters

Table 3-119 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Table 3-120 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
gesMetadata	Yes	Object	Object for storing metadata message information.

Response Parameters

Status code: 200

Table 3-121 Response body parameter

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.
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	OK
500	Internal Server Error

3.4.5 Deleting Metadata

Function

This API is used to delete the metadata.

URI

DELETE /v1.0/{project_id}/graphs/metadatas/{metadata_id}

Table 3-122 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 3-123 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Indicates the user token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-124 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.

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	OK
400	Bad Request

3.5 Task Center APIs

3.5.1 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 3-125 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID
job_id	Yes	String	Job ID

Request Parameters

Table 3-126 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-127 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	Job ID

Parameter	Type	Description
status	String	Job status: <ul style="list-style-type: none"> • pending • running • success • failed
jobType	String	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	JobDetail object	This parameter is returned only when jobName is set to ImportGraph 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 3-128 JobDetail

Parameter	Type	Description
schemaPath	Array of schemaPath objects	Path for storing metadata
edgesetPath	Array of edgesetPath objects	Path for storing the edge data set
vertexsetPath	Array of vertexsetPath objects	Path for storing the vertex data set

Table 3-129 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"> ● success: Imported successfully. ● Failed: Failed to import the file. ● partFailed: Partially failed.
cause	String	Import failure cause

Table 3-130 edgesetPath

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

Table 3-131 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"> ● success: Imported successfully. ● Failed: Failed to import the file. ● partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 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	OK
400	Bad Request

3.5.2 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 3-132 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Table 3-133 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 10 .
offset	No	String	Start position of the request. The default value is 0 .
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"> • running • waiting • success • failed

Request Parameters

Table 3-134 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 3-135 Response body parameter

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.
jobCount	Integer	Total number of jobs
jobList	Array of Job objects	Task list

Table 3-136 Job

Parameter	Type	Description
jobId	String	Job ID
status	String	Job status. <ul style="list-style-type: none"> • pending • running • success • failed
jobType	String	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	JobDetail object	This parameter is returned only when jobName is set to ImportGraph 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 3-137 JobDetail

Parameter	Type	Description
schemaPath	Array of schemaPath objects	Path for storing metadata
edgesetPath	Array of edgesetPath objects	Path for storing the edge data set
vertexsetPath	Array of vertexsetPath objects	Path for storing the vertex data set

Table 3-138 schemaPath

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

Table 3-139 edgesetPath

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

Parameter	Type	Description
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 indicates that this field is not returned in the current version.

Table 3-140 vertexsetPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: <ul style="list-style-type: none"> ● success: Imported successfully. ● Failed: Failed to import the file. ● partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 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	OK
400	Bad Request

4 Service Plane APIs

4.1 Vertex Operation APIs

4.1.1 Querying Vertices That Meet Filter Criteria

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query
- Parameter description

Table 4-1 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-2 Request body parameter description

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type
vertexFilters	Either labels or vertexFilters is mandatory.	Json	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 10 .
sorts	No	Json	Result sorting property, in JSONArray format

Table 4-3 sorts parameter description

Parameter	Mandatory	Type	Description
key	Either Key or propertyName is mandatory.	String	Possible values are id , label , and property . These values indicate that IDs, labels, or properties are sorted.
propertyName	Either Key or propertyName is mandatory.	String	Property name

Parameter	Mandatory	Type	Description
orderValue	No	String	Possible values are incr and decr , which indicate ascending and descending order respectively. The default value is incr .

Table 4-4 vertexFilters parameter structure

Parameter	Mandatory	Type	Description
propertyName	Yes	String	Property name
predicate	Yes	String	Logical relationship. Possible values are =, <, >, <=, >=, range , has , hasNot .
values	No	String	Property value.
type	No	String	Logical relationship of filter criteria. Possible values are and and or . The default value is and .

- Example 1 for vertexFilters

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

Response

Table 4-5 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.

Parameter	Mandatory	Type	Description
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.
job_id	No	String	ID of the vertex query job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-6 Return code for failed requests

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	Service internal error.
503 Service Unavailable	Service unavailable.

4.1.2 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

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds={vertex_ids}
- Parameter description

Table 4-7 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertexIds	Yes	String	IDs of the vertices to be queried. When multiple IDs are specified by vertexIds , separate the IDs with commas (,) in the URL. NOTE For graphs of the 100-billion-edge type, only one vertex ID is supported.

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

Table 4-8 Parameter description

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.

Parameter	Type	Description
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.
data	Json	Query results

Table 4-9 data parameter description

Parameter	Type	Description
vertices	List	Vertex result set. If no corresponding vertices are found, the value of vertices is empty.

- Response example (successful request)

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"
          ]
        }
      }
    ]
  }
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8204"
}
```

Return Value

- Normal
200

- Abnormal

Table 4-10 Return code for failed requests

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.

4.1.3 Adding a Vertex

Function

This API is used to add a vertex.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices
- Parameter description

Table 4-11 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-12 Request body parameter description

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

Response

- Parameter description

Table 4-13 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.

Parameter	Mandatory	Type	Description
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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-14 Return code for failed requests

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.

4.1.4 Deleting a Vertex

Function

This API is used to delete a vertex.

URI

- URI format
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}
- Parameter description

Table 4-15 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertexId	Yes	String	Vertex ID

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-16 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-17 Return code for failed requests

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.

4.1.5 Updating Vertex Properties

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/properties/action?action_id={actionId}
- Parameter description

Table 4-18 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertexId	Yes	String	Vertex ID
actionId	Yes	String	Operator. Possible values: <ul style="list-style-type: none"> • UPDATE: Update the value of a property. • ADD: Add the value to a property. When the property's cardinality is single, the operation is the same as that of UPDATE. When cardinality is list or set, the operator adds a value to a set. • DEL: Delete a property value.

Request

- Request example

```

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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-19 Request body parameter description

Parameter	Mandatory	Type	Description
properties	Yes	Json	Value of each property

Parameter	Mandatory	Type	Description
label	No	String	Name of a label

Response

- Parameter description

Table 4-20 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-21 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	No operation permission.

Return Value	Description
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	Service unavailable.

4.1.6 Querying Vertex Data 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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query
- Parameter description

Table 4-22 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).
- Request body parameter description

Table 4-23 Request body parameter description

Parameter	Mandatory	Type	Description
vertices	Yes	String	Vertex ID array to be queried

Response

- Parameter description

Table 4-24 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.
data	No	Json	The data field is contained when the query is successful, and the data field contains the vertices query result.
result	Yes	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .

- Response example (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"
}

```

- Response example (failed request)

Http Status Code: 400

```

{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}

```

4.1.7 Adding Vertices in Batches

Function

This API is used to add vertices in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
- Parameter description

Table 4-25 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

```

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 [Constraints of Using Service Plane APIs](#).
- In the example, if vertex **6** already exists in the graph, properties of vertex **6** are overwritten.
- Parameter description

Table 4-26 Request body parameter description

Parameter	Mandatory	Type	Description
vertices	Yes	Json	Vertex array to be added. You are advised to add a maximum of 10,000 vertices at a time.
overrideExists	No	Boolean	Whether to overwrite the existing vertices in the vertices parameter. The default value is false , indicating that existing vertices are ignored. For 100-billion-edge graphs, an error is reported for existing vertices. The value true indicates that existing vertices in the vertices parameter are overwritten.

Table 4-27 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

- Parameter description

Table 4-28 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-29 Return code for failed requests

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.

4.1.8 Deleting Vertices in Batches

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete
- Parameter description

Table 4-30 URI parameter description

Parameter	Mandator y	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-31 Request body parameter description

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

Response

- Parameter description

Table 4-32 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

4.1.9 Updating Vertex Properties in Batches

Function

This API is used to update vertex properties in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?action_id={actionId}
- Parameter description

Table 4-33 URI parameter description

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

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-34 Request body parameter description

Parameter	Mandatory	Type	Description
vertices	Yes	Json	Vertex array to be updated
ignoreError (2.2.13)	No	Boolean	Whether to ignore the update error of specific vertices. The default value is false , indicating that an error that causes the update failure must be detected. For example, if the vertex to be updated does not exist, an error is reported and no vertex is updated. If the value is true , similar errors are ignored and other vertex properties without errors are updated.

Table 4-35 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

- Parameter description

Table 4-36 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-37 Return code for failed requests

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.

4.1.10 Adding a Vertex Label

Function

This API is used to add a vertex label.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels
- Parameter description

Table 4-38 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-39 Request body parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Vertex label

Response

- Parameter description

Table 4-40 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": "Same label [user] already exists",
  "errorCode": "GES.8213"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-41 Return code for failed requests

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.

4.1.11 Deleting a Vertex Label

Function

This API is used to delete a vertex label.

URI

- URI format
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels/{label_name}
- Parameter description

Table 4-42 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID
label_name	Yes	String	Vertex label

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-43 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.

Parameter	Mandatory	Type	Description
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)
Http Status Code: 200

```
{
}
```
- Response example (failed request)
Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-44 Return code for failed requests

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.

4.1.12 Exporting Filtered Vertices

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export
- Parameter description

Table 4-45 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example (Only the asynchronous mode is supported.)

```
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"
  }
}
```

- Parameter description

Table 4-46 Request body parameter description

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type
vertexFilters	Either labels or vertexFilters is mandatory.	Json	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 4-4 .
exportPath	Yes	String	Export path
fileName	No	String	Name of the exported file

Parameter	Mandatory	Type	Description
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 4-256 .

Response

Table 4-47 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.
job_id	No	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-48 Return code for failed requests

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	Service internal error.
503 Service Unavailable	Service unavailable.

4.1.13 Deleting Filtered Vertices

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete

Request

- Request example (Only the asynchronous mode is supported.)
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"
      ]
    }
  ]
}
```
- Parameter description

Table 4-49 Request body parameter description

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type

Parameter	Mandatory	Type	Description
vertexFilters	Either labels or vertexFilters is mandatory.	Json	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 4-4 .

Response

Table 4-50 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.
job_id	No	String	ID of the vertex query job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane .

- Response example (successful request)

Http Status Code: 200

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

- Response example (failed request)

Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-51 Return code for failed requests

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	Service internal error.
503 Service Unavailable	Service unavailable.

4.2 Edge Operation APIs

4.2.1 Querying Edges That Meet Filter Criteria

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query
- Parameter description

Table 4-52 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Table 4-53 Request body parameter description

Parameter	Mandatory	Type	Description
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type
edgeFilters	Either labels or edgeFilters 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	Json	Result sorting property. It is in JSONArray format.

Table 4-54 sorts parameter description

Parameter	Mandatory	Type	Description
key	Either Key or propertyName is mandatory.	String	Possible values are label and property , which indicate that labels or properties are sorted.

Parameter	Mandatory	Type	Description
propertyName	Either Key or propertyName is mandatory.	String	Property name
orderValue	No	String	Possible values are incr and decr , which indicate ascending and descending order respectively. The default value is incr .

Table 4-55 edgeFilters parameter structure

Parameter	Mandatory	Type	Description
propertyName	Yes	String	Property name
predicate	Yes	String	Logical relationship. Possible values are =, <, >, <=, >=, range , has , hasNot
values	No	String	Property value.
type	No	String	Logical relationship of filter criteria. Possible values are and and or . The default value is and .

Response

Table 4-56 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.
job_id	No	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-57 Return code for failed requests

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.

4.2.2 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

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source={sourceVertex}&target={targetVertex}&index={index}
- Parameter description

Table 4-58 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
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, all edges between the source and target vertices are queried NOTE This parameter is not supported for graphs of the 100-billion-edge type.
label (for 100-billion-edge)	No	String	Label of an edge

Request

- Request example

```
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 [Constraints of Using Service Plane APIs](#).

Response

Table 4-59 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.
data	No	Json	Query results. This parameter is left blank when the query fails.

Table 4-60 data parameter description

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge result set. If no corresponding edges are found, the value of edges is empty.

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "edges": [
      {
        "index": "6",
        "source": "Ray",
        "label": "rate",
        "properties": {
```

```

    "Score": [
      3
    ],
    "Datetime": [
      "2000-11-22 19:23:05"
    ]
  },
  "target": "Rocky"
}
]
}
}

```

- Response example (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8107"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-61 Return code for failed requests

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.

4.2.3 Adding an Edge

Function

This API is used to add an edge.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges
- Parameter description

Table 4-62 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

POST `http://{SERVER_URL}/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
  }
}
```

 **NOTE**

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

- Request body parameter description

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 __DEFAULT__ .
properties	No	Json	Value of each property
parallelEdge	No	Object	Repetitive edge processing

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

Response

- 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

```

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

```

- Response example (failed request)
Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-63 Return code for failed requests

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.

4.2.4 Deleting an Edge

Function

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

URI

- URI format

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

Table 4-64 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex name
target	Yes	String	Target vertex name
index	No	Integer	Edge index <ul style="list-style-type: none"> • If property has been set, ignore this parameter. • If property is not set, the edge is deleted based on index. • If neither property nor index is set, all edges between source and target are deleted.
label	No	String	Indicates the label of an edge, which can accelerate the search of property values. This parameter must be used together with property .
property	No	String	Property name of the edge to be deleted. This parameter must be used together with value .
value	No	String	Indicates the property value of the edge to be deleted. This parameter must be used together with property .

Request

- Request example

```
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 [Constraints of Using Service Plane APIs](#).

Response

- 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

Http Status Code: 200

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

- Response example (failed request)

Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-65 Return code for failed requests

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.

4.2.5 Updating Edge Properties

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id={actionId}&source={sourceVertex}&target={targetVertex}&index={index}
- Parameter description

Table 4-66 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
actionId	Yes	String	Operator. Possible values: <ul style="list-style-type: none"> • update: Update a property value. • add: Add a property value. When the property's cardinality is single, the operation is the same as that of UPDATE. When cardinality is list or set, the operator adds a value to a set. • del: Delete a property value.
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 vertices are modified.

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Parameter description

Table 4-67 Request body parameter description

Parameter	Mandatory	Type	Description
properties	Yes	Json	Value of each property
label	No	String	Name of a label
targetProperties	No	Array	<p>Properties used to determine duplicate edges.</p> <ul style="list-style-type: none"> • 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. • If this parameter is left blank or no property is specified for the input edge, the first edge that meets the criteria is updated. <p>For details about the property elements, see Table 4-68.</p>

Table 4-68 targetProperty parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Label name. The label of duplicate edges is determined by the property.

Parameter	Mandatory	Type	Description
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.

Response

- Parameter description

Table 4-69 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-70 Return code for failed requests

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.

4.2.6 Querying Edge Data 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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query
- Parameter description

Table 4-71 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Example request (unavailable for 100-billion-edge)
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"
    }
  ]
}

```

- Example request (for 100-billion-edge)

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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-72 Request body parameter description

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be queried

Table 4-73 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 NOTE This parameter is not supported for graphs of the 100-billion-edge type.
label (for 100-billion-edge)	No	String	Edge label

Response

- Parameter description

Table 4-74 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.
data	No	String	The data field is contained when the query is successful, and the data field contains the edges query result.
result (for 100-billion-edge)	Yes	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .

- Response example of a successful request (unavailable for 100-billion-edge)

```
{
  "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"
      }
    ]
  }
}
```

- Response example of a successful request (for 100-billion-edge)

```
{
  "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"
}

```

- Response example (failed request)

```

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

```

4.2.7 Adding Edges in Batches

Function

This API is used to add edges in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
- Parameter description

Table 4-75 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

```
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": true
  },
  "createNotExists": true
}
```

 **NOTE**

- SERVER_URL**: Address for accessing a graph. For details about its value, see [Constraints of 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.
- Request body parameter description

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be added
parallelEdge	No	Object	Repetitive edge processing

Parameter	Mandatory	Type	Description
action	No	String	<p>Processing mode of repetitive edges. The value can be allow, ignore, or override. The default value is allow.</p> <ul style="list-style-type: none"> • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten. <p>NOTE For 100-billion-edge graphs, the value can only be override.</p>
ignoreLabel	No	Boolean	<p>Whether to ignore labels on repetitive edges. The value is true or false, and the default value is true.</p> <ul style="list-style-type: none"> • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge. <p>NOTE For 100-billion-edge graphs, the value can only be false.</p>
createNotExists	No	Boolean	<p>Whether to add source or target vertices that do not exist in the edges parameter before adding edges. The default value is false, which does not affect the original functions and semantics.</p> <p>If this parameter is set to true, source or target vertices that do not exist in the edges parameter are added prior to the edges.</p> <p>NOTE For 100-billion-edge graphs, the value can only be false.</p>

Table 4-76 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	Json	Value of each property

Response

- 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

```

Http Status Code: 200
{
  "result": "success",
  "data": {
    "edges": [
      {
        "index": "7",
        "source": "46",
        "target": "39"
      },
      {
        "index": "0",
        "source": "46",
        "target": "38"
      }
    ]
  }
}

```

- Response example (failed request)

```

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

Return Value

- Normal
200
- Abnormal

Table 4-77 Return code for failed requests

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.

4.2.8 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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete
- Parameter description

Table 4-78 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

```
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_Jiuhua Hot Spring"
    }
  ],
  "ignoreError": true
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-79 Request body parameter description

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be deleted
executionMode	No	String	sync indicates the synchronous mode, and async indicates the asynchronous mode. The default value is sync .
ignoreError	No	Boolean	Whether to ignore errors, for example, the edge to delete does not exist. The default value is false , indicating that errors will not be ignored. Errors in JSON format cannot be ignored.

Table 4-80 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

Parameter	Mandatory	Type	Description
label	No	String	Label of an edge. If the index parameter is set, this parameter is ignored. If the index parameter is not set, an edge that meets the source , target , and label conditions is deleted. If the specified label value does not exist in the schema or the edge with the same label does not exist, no edge will be deleted.

Response

- Sync mode

Table 4-81 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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

- Async mode

Table 4-82 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.
job_id	No	String	ID of the deletion job. This parameter is left blank when the request fails. This ID can be used as a parameter to obtain the deletion result through the API for querying the job status.
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

- Response example (successful request)

Http Status Code: 200

```
{
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",
  "jobType": 3
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [test_117d] is not found",
  "errorCode": "GES.8402"
}
```

4.2.9 Updating Edge Properties in Batches

Function

This API is used to update edge properties in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}
- Parameter description

Table 4-83 URI parameter description

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

Request

- Request example

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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be updated
ignoreError	No	Boolean	Whether to ignore the update error of specific edges. The default value is false , 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. If the value is true , similar errors are ignored and other edge properties without errors are updated.

Table 4-84 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	Json	Value of each property

Response

- 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.

Parameter	Mandatory	Type	Description
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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "edge source vertex [46] does not exist",
  "errorCode": "GES.8221"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-85 Return code for failed requests

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.

4.2.10 Exporting Filtered Edges

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export

Request

- Request example (Only the asynchronous mode is supported.)
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"
}
}
```
- Parameter description

Table 4-86 Request body parameter description

Parameter	Mandatory	Type	Description
export Path	Yes	String	Export path
fileName	No	String	Name of the exported file
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 4-256 .
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type

Parameter	Mandatory	Type	Description
edgeFilters	Either labels or edgeFilters is mandatory.	String	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 4-55 .

Response

Table 4-87 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.
job_id	No	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

- Response example (successful request)

Http Status Code: 200

```
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 0
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8103"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-88 Return code for failed requests

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.

4.2.11 Deleting Filtered Edges

Function

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

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete

Request

- Request example (Only the asynchronous mode is supported.)
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"
  ]
}
```
- Parameter description

Table 4-89 Request body parameter description

Parameter	Mandatory	Type	Description
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type
edgeFilters	Either labels or edgeFilters is mandatory.	String	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 4-55 .

Response

Table 4-90 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.
job_id	No	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200

- Abnormal

Table 4-91 Return code for failed requests

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.

4.3 Metadata Operation APIs

4.3.1 Adding a Label

Function

This API is used to add labels.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
- Parameters

Table 4-92 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Example request (unavailable for 100-billion-edge)

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
{
  "name": "book",
  "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"
      }
    }
  ]
}
```

- Example request (for 100-billion-edge graphs)

```
POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/schema/labels
{
  "name": "book",
  "properties": [
    "Title",
    "Version",
    "Category"
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

- Request body parameters

Table 4-93 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label A label name can contain a maximum of 256 characters. Only letters, digits, spaces, and special characters %, @, #, \$, :, ? , * , . , + , - are allowed.
properties	Yes	Json	Property array to be added. The array element is property. For details about the parameters, see Table 4-94 .

Parameter	Mandatory	Type	Description
properties (for 100-billion-edge)	Yes	Json	Property array to be added. The array element is of the string type and corresponds to the property name.

Table 4-94 property parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. Possible values: <ul style="list-style-type: none"> • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata types in Table 3-106 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2, the parameter contains typeName1:science and typeName2:literature .

Response

- Parameters

Table 4-95 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.
result	No	String	If the execution is successful, the value of result is success .
cause (for 100-billion-edge)	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.

- Response example (successful request)

Http Status Code: 200

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

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-96 Return code for failed requests

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.

Return Value	Description
503 Service Unavailable	Service unavailable.

4.3.2 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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties
- URI format (for 100-billion-edge)
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
- Parameter description

Table 4-97 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_name	Yes	String	Label name

Request

- Example request (unavailable for 100-billion-edge)
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties

```
{
  "name": "book",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    }
  ]
}
```

```

    }
  },
  {
    "property": {
      "name": "Category",
      "typeName1": "science",
      "typeName2": "literature",
      "typeNameCount": "2",
      "dataType": "enum"
    }
  }
]
}

```

- Example request (for 100-billion-edge)

```

POST http://{SERVER_URL}/ges/v1.0/{projectId}/graphs/{graphName}/schema?label={labelName}
{
  "properties": [
    "Title",
    "Version",
    "Category"
  ]
}

```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-98 Request body parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label. A label name can contain a maximum of 256 characters. Only letters, digits, spaces, and special characters %, @, #, \$, :, ;, ?, *, ., +, - are allowed.
properties	Yes	Json	Property array to be appended. The array element is property. For details about the parameters, see Table 4-99 .
properties (for 100-billion-edge)	Yes	Json	Property array to be updated. The array element is property. For details about the parameters, see Table 4-99 .

Table 4-99 property parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 and 30. 3. The property under a label must be unique.
cardinality	Yes	String	Composite type of a property. Possible values: <ul style="list-style-type: none"> • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata types in Table 3-106 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2, the parameter contains typeName1:science and typeName2:literature .

Response

- Parameter description

Table 4-100 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.

Parameter	Mandatory	Type	Description
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.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-101 Return code for failed requests

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.

4.3.3 Query Labels (for 100-billion-edge graphs only)

Function

This API is used to query a label.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
- Parameters

Table 4-102 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_name	Yes	String	Label name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameters

Table 4-103 Parameter description

Parameter	Mandatory	Type	Description
properties	Yes	Property	Property array.
result	Yes	String	If the execution is successful, the value of result is success .

- Response example (successful request)

```

Http Status Code: 200
{
  "data": {
    "properties": [
      {
        "name": "Rating",
        "type": "int",
        "cardinality": "single"
      },
      {

```

```

        "name": "Datetime",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  "result": "success"
}

```

- Response example (failed request)

```

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

```

Return Value

- Normal
200
- Abnormal

Table 4-104 Return code for failed requests

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.

4.3.4 Querying Graph Metadata Details

Function

This API is used to query graph metadata details.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema
- Parameter description

Table 4-105 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

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 [Constraints of Using Service Plane APIs](#).

Response

Table 4-106 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.
data	No	Json	Query results. This parameter is left blank when the request fails.

Table 4-107 data parameter description

Parameter	Type	Description
schema	List	Definitions of each label and associated property field

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "schema": [
      {
        "label": "__DEFAULT__"
      },
      {
        "label": "friends"
      },
      {
        "label": "movie",
        "properties": [
          {
            "name": "Title",
            "type": "string",
            "cardinality": "single"
          },
          {
            "name": "Year",
            "type": "int",
            "cardinality": "single"
          },
          {
            "name": "Genres",
            "type": "string",
            "cardinality": "set"
          }
        ]
      }
    ],
    {
      "label": "user",
      "properties": [
        {
          "name": "Name",
          "type": "string",
          "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",
          "type": "string",

```

```

        "cardinality": "single"
      },
      {
        "name": "Zip-code",
        "type": "char array",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "rate",
    "properties": [
      {
        "name": "Score",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "Datetime",
        "type": "date",
        "cardinality": "single"
      }
    ]
  }
]
}
}
}

```

- Response example (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8003"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-108 Return code for failed requests

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.

4.3.5 Changing Property Names in Batches

Function

This API is used to change property names in batches.

URI

- URI format
PUT /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/properties
- Parameter description

Table 4-109 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-110 Request body parameter description

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

- Parameter description

Table 4-111 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.
updated_count	No	Integer	Number of properties that are successfully updated

- Response example (successful request)

```
Http Status Code: 200
{
  "updated_count": 4
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label name does not exist",
  "errorCode": "GES.8807"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-112 Return code for failed requests

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.

4.3.6 Deleting a Label

Function

This API is used to delete a label as well as the vertices and edges associated with the label.

URI

- URI format
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{labelName}
- Parameter description

Table 4-113 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_name	Yes	String	Name of a label

Request

- Request example
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{labelName}

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-114 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.
data	No	Json	Query results. This parameter is left blank when the request fails.

Table 4-115 data parameter description

Parameter	Type	Description
outputs	int	Number of deleted vertices or edges when a label is deleted.

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "outputs": 3
  },
  "status": "success"
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8003"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-116 Return code for failed requests

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.

4.3.7 Adding Labels in Batches

Function

This API is used to add labels in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add
- Parameter description

Table 4-117 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add

```
{
  "labels": [
    {
      "name": "book",
      "properties": [
        {
          "property": {
            "name": "title",
            "cardinality": "single",
            "dataType": "string"
          }
        }
      ]
    }
  ]
}
```



```

    }
  ]
},
{
  "name": "movie",
  "properties": [
    {
      "property": {
        "name": "movieid",
        "cardinality": "single",
        "dataType": "int"
      }
    }
  ]
}
]
}

```

 **NOTE**

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

Table 4-118 label parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label A label name can contain a maximum of 256 characters. Only letters, digits, spaces, and special characters %, @, #, \$, :, ?, *, ., +, - are allowed.
properties	Yes	Json	Property array to be added. The array element is property. For details about the parameters, see Table 4-119 .

Table 4-119 property parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Property name <ol style="list-style-type: none"> 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14, 15 or 30. 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"> • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata types in Table 3-106 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2 , the parameter contains typeName1:science and typeName2:literature .

Response

- Parameter description

Table 4-120 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.
result	No	String	If the execution is successful, the value of result is success .

Parameter	Mandatory	Type	Description
data	No	Json	If some labels fail to be added, this field contains the names of the failed labels and the failure causes.

- Response example (successful request)

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

- Request example (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<"
      }
    ]
  }
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "label already exists",
  "errorCode": "GES.8801"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-121 Return code for failed requests

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.

4.3.8 Adding Properties (for 100-billion-edge graphs only)

Function

This API is used to add properties.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/properties
- Parameters

Table 4-122 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/properties

```
{
  "properties": [
    {
      "name": "title",
      "cardinality": "single",
      "dataType": "string"
    }
  ]
}
```

 **NOTE**

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-123 property parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. Possible values: <ul style="list-style-type: none"> • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata type in Metadata property constraints .

Response

- Parameters

Table 4-124 Parameter description

Parameter	Mandatory	Type	Description
result	Yes	String	If the execution is successful, the value of result is success .

- Response example (successful request)

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

- Response example (failed request)

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

Return Value

- Normal
200
- Abnormal

Table 4-125 Return code for failed requests

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.

4.3.9 Query Properties (for 100-billion-edge graphs only)

Function

This API is used to query a property.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema?property={propertyName}
- Parameters

Table 4-126 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
propertyName	Yes	String	Property name

Request

- Request example
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema?propertyName={propertyName}

 **NOTE**

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

Response

- Parameters

Table 4-127 property parameter description

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. Possible values: <ul style="list-style-type: none"> single list set
type	Yes	String	Data type of a property. For details, see the metadata type in Metadata property constraints .

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "property": {
      "name": "creation",
      "type": "string",
      "cardinality": "single"
    }
  },
  "result": "success"
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8204"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-128 Return code for failed requests

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.

4.3.10 Querying the Property List (for 100-billion-edge graphs only)

Function

This API is used to query the property list.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/properties
- Parameters

Table 4-129 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/properties

 **NOTE**

SERVER_URL: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).

Response

- Parameters

Table 4-130 property parameter description

Parameter	Mandatory	Type	Description
properties	Yes	List	Property list. For details, see Table 3 property parameters .

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "properties": [
      {
        "name": "test_weight",
        "type": "float",
        "cardinality": "single"
      },
      {
        "name": "test_gender",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  "result": "success"
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8204"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-131 Return code for failed requests

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.

4.3.11 Querying Schema Structure

Function

This API is used to query the structure of the generated schema (obtained from OBS).

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/structure?detail={details}
- Parameters

Table 4-132 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
details	No	String	Detailed information of the schema structure. The value can be SIMPLE or FULL . If you set this parameter to SIMPLE , only the labels of vertices or edges are returned. If you set it to FULL , the number of vertices or edges is returned in addition to the labels. If this parameter is left empty, the default value SIMPLE is used.

Request

- Example
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure?detail=SIMPLE

Response

- Parameters

Table 4-133 property parameter description

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.
schema	JSON	Schema structure. For details, see schema parameter description .

Table 4-134 schema parameter description

Parameter	Type	Description
vertices	JSONArray	Vertex result set. If the graph is empty, the return value is empty. For details, see vertices parameter description .
edges	JSONArray	Edge result set. If the graph is empty, the return value is empty. For details, see edges parameter description .

Table 4-135 vertices parameter description

Parameter	Type	Description
vertex	String	Label name
weight	String	Number of vertices that have the label

Table 4-136 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

- Response example (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"
      }
    ]
  }
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "Bad Request, parameter [detail] cannot be null.",
  "errorCode": "GES.8813"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-137 Return code for failed requests

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.

4.3.12 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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build
- Parameters

Table 4-138 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Example request
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build

Response

- Parameters

Table 4-139 property parameter description

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.
job_id	String	ID of the vertex query job. This parameter is left blank when the request fails.

- Response example (successful request)

```
{
  "jobId": "2e0c08e1-3fbb-4b33-8776-4809176068d7154236181",
  "jobType": 1
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Bad Request ",
  "errorCode": "GES.8813"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-140 Return code for failed requests

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.

4.4 Index Operation APIs

4.4.1 Creating an Index

Function

This API is used to create indexes based on the specified information such as indexName and IndexType. Currently, GES supports composite indexes.

- 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 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, full-text indexes are available for 100-billion-edge graphs only.

Index Feature Comparison

Feature	Fuzzy Search	Speed	Flexibility
Composite indexes	No	Fast	Fixed composite property keys only
Full-text indexes	Yes	Slower than composite indexes	Randomly combined property keys

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices
- Parameter description

Table 4-141 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Sample request (composite 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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-142 Request body parameter description

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. GlobalCompositeVertex-Index is a global composite vertex index.
hasLabel (for 100-billion-edge)	No	Boolean	Whether labels exist. The default value is false . <ul style="list-style-type: none"> • true • false
indexProperty	No (If hasLabel is false or null , this parameter is mandatory.)	String	Index property list. 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, it is recommended that the length be less than or equal to 40 bytes. The excess part will be deleted.

Response

- Parameter description

Table 4-143 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.

Parameter	Mandatory	Type	Description
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.
job_id	No	String	ID of an asynchronous job NOTE <ul style="list-style-type: none"> You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.
jobType	No	String	Type of an asynchronous job
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

Http Status Code: 200

```
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 8
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8603"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-144 Return code for failed requests

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.

4.4.2 Deleting an Index

Function

This API is used to delete an index based on the specified indexName.

URI

- URI format
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName}
- Parameter description

Table 4-145 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
indexName	Yes	String	Index name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-146 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.

Parameter	Mandatory	Type	Description
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.
job_id	No	String	ID of an asynchronous job NOTE <ul style="list-style-type: none"> You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs. This parameter is not supported for graphs of the 100-billion-edge type.
jobType	No	String	Type of an asynchronous job

- Response example (successful request)

Http Status Code: 200

```
{
  "jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232",
  "jobType": 9
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8604 "
}
```

Return Value

- Normal
200
- Abnormal

Table 4-147 Return code for failed requests

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.

Return Value	Description
503 Service Unavailable	Service unavailable.

4.4.3 Querying Indexes

Function

This API is used to query all indexes created on a graph.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices
- Parameter description

Table 4-148 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-149 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.
result	No	String	Query results. If the query is successful, success is displayed.
indices	No	List	Indexes of the query results
indexType	No	String	Index types of the query results
indexName	No	String	Index names of the query results
indexProperty	No	List	Index properties of the query results
hasLabel (for 100-billion-edge)	No	Boolean	Whether the indexes of the query results contain labels NOTE For full-text indexes, the default value is false .

- Response example (successful request)

Http Status Code: 200

```
{
  "data": {
    "result": "success",
    "indices": [
      {
        "indexType": "GlobalCompositeVertexIndex",
        "indexName": "ageIdx",
        "indexProperty": [
          "age"
        ],
        "hasLabel": "true"
      }
    ]
  }
}
```

- Response example of a successful request (for hundred-billion-edge)

```

Http Status Code: 200
{
  "data": {
    "indices": [
      {
        "indexType": "GlobalCompositeVertexIndex",
        "indexName": "ageIndx",
        "indexProperty": [
          "age"
        ],
        "hasLabel": true
      }
    ]
  },
  "result": "success"
}

```

- Response example (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8605"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-150 Return code for failed requests

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.

4.5 Gremlin Operation APIs

4.5.1 Executing Gremlin Queries

Function

This API is used to return the query result of a Gremlin statement.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query
- Parameter description

Table 4-151 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).
- The size of the request body cannot exceed 64 MB.
- Request body parameter description

Table 4-152 Request body parameter description

Parameter	Mandatory	Type	Description
command	Yes	String	Query command (Gremlin language)

Response

- Parameter description

Table 4-153 Response 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.
data	No	Json	Query results. This parameter is left blank when the request fails.

- Response example (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"
          ]
        }
      }
    ]
  }
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "org.apache.tinkerpop.gremlin.driver.exception.ResponseException: No such property: g1 for class: Script4",
  "errorCode": "GES.8503"
}
```


Return Value

- Normal
200
- Abnormal

Table 4-154 Return code for failed requests

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.

4.6 Algorithm APIs

4.6.1 Running Algorithms

Function

This API is used to run specified algorithms based on entered parameters.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
- Parameter description

Table 4-155 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example**
 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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description**
 For details about the parameters, see [Table 4-158](#).

Response

Table 4-156 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.
job_id	No	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Parameter	Mandatory	Type	Description
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

- Response example (successful request)
 Http Status Code: 200

```
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```
- Response example (failed request)
 Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-157 Return code for failed requests

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.

4.6.2 Algorithm API Parameter References

4.6.2.1 Common Algorithm Parameters

Algorithm request

- Request body example

```
{
  "algorithmName": "XXX",
  "parameters": {
    ...
  }
}
```

```
}  
}
```

- Request body parameter description

Table 4-158 Request body parameter description

Parameter	Mandatory	Type	Description
algorithm Name	Yes	String	<p>Algorithm name.</p> <p>Available values are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> • pagerank • personalrank • kcore • k_hop • shortest_path • all_shortest_paths • filtered_shortest_path • sssp • shortest_path_of_vertex_sets • n_paths • closeness • label_propagation • louvain • link_prediction • node2vec • realtime_recommendation • common_neighbors • connected_component • degree_correlation • triangle_count • cluster_coefficient • common_neighbors_of_vertex_sets • all_shortest_paths_of_vertex_sets • filtered_circle_detection • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths • filtered_n_paths
parameters	Yes	Json	<p>Algorithm parameters. For details, see the parameter description of each algorithm.</p>

Table 4-159 New Body parameters of version 2.1.7

Parameter	Mandatory	Type	Description
execution Mode	No	String	<ul style="list-style-type: none"> • sync: synchronous • async: asynchronous <p>The default value is async.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> • k_hop • shortest_path • all_shortest_paths • filtered_shortest_path • shortest_path_of_vertex_sets • n_paths • realtime_recommendation
offset	No	Integer	<p>Synchronization result offset. The default value is 0.</p> <p>NOTE This parameter is valid when executionMode is sync.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> • k_hop • shortest_path • all_shortest_paths • shortest_path_of_vertex_sets • n_paths • realtime_recommendation • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths

Parameter	Mandatory	Type	Description
limit	No	Integer	<p>Maximum number of returned synchronization results. The maximum value is 100000. The default value is 100000.</p> <p>NOTE This parameter is valid when executionMode is sync.</p> <p>Supported algorithms are as follows (algorithm names you can call):</p> <ul style="list-style-type: none"> • k_hop • shortest_path • all_shortest_paths • shortest_path_of_vertex_sets • n_paths • realtime_recommendation • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths

Table 4-160 New Body parameters of version 2.2.4

Parameter	Mandatory	Type	Description
vertex_filter	No	Json	<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"> • filtered_shortest_path • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths <p>For details about the format, see Table 4-268 in "Filtered-query API".</p>
edge_filter	No	Json	<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"> • filtered_shortest_path • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths <p>For details about the format, see Table 4-268 in "Filtered-query API".</p>

Parameter	Mandatory	Type	Description
filters	No	Json	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 filters element formats . Supported algorithms: <ul style="list-style-type: none"> filtered_n_paths

Result

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.

- Response when an algorithm is successfully executed:

```
{
  "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.

- Response when an algorithm fails to be executed:

```
Http Status Code: 400
{
  "errorMessage": "Running algorithm [XXXX] error: YYYYYYYYYY!",
  "errorCode": "GES.8301"
}
```

- Parameter description

Table 4-161 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.

Parameter	Mandatory	Type	Description
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 waiting , running , and complete . This parameter is left blank when the query fails.
data	No	Json	Algorithm execution result. This parameter is left blank when the query fails.

- Parameter description

Table 4-162 outputs parameter description

Parameter	Type	Description
response_data	Json	Result returned after an algorithm is executed NOTE The results vary with algorithms. For details, see parameter descriptions of each algorithm.
runtime	Double	Algorithm execution time. The unit is second.
data_return_size	Integer	Number of records returned from a query
data_offset	Integer	Result offset of a query
data_total_size	Integer	Total amount of result data generated by asynchronous jobs.

 **NOTE**

Parameters **data_return_size**, **data_offset**, and **data_total_size** are used for pagination queries. After each of some algorithms (Shortest Path, Closeness Centrality, Link Prediction, Degree Correlation, Triangle Count, and Cluster Coefficient) is executed, only one value is returned and the result does not contain the parameter.

4.6.2.2 PageRank

Table 4-163 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
alpha	No	Weight coefficient (also called damping coefficient)	Double	A real number between 0 and 1 (excluding 0 and 1)	0.85
convergence	No	Convergence	Double	A real number between 0 and 1 (excluding 0 and 1)	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	1000
directed	No	Whether to consider the edge direction	Boolean	true or false	true

 **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 4-164 response_data parameter description

Parameter	Type	Description
pagerank	List	PageRank value of each vertex. The format is as follows: <code>[{vertexId:rankValue},...]</code> , where vertexId is of the string type. rankValue is of the double type.

4.6.2.3 PersonalRank

Table 4-165 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Vertex ID	String	-	-
alpha	No	Weight coefficient	Double	A real number between 0 and 1 (excluding 0 and 1)	0.85
convergence	No	Convergence	Double	A real number between 0 and 1 (excluding 0 and 1)	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	1000
directed	No	Whether to consider the edge direction	Boolean	true or false	true

 **NOTE**

For details about algorithm iterations and convergence, see [Iterations and Convergence of PageRank](#).

Table 4-166 response_data parameter description

Parameter	Type	Description
source	String	-
personalrank	List	PersonalRank value of each vertex. The format is as follows: [{vertexId:rankValue},...], where vertexId is of the string type. rankValue is of the double type.

4.6.2.4 K-core

Table 4-167 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
k	Yes	Number of cores The algorithm returns vertices whose number of cores is greater than or equal to k.	Integer	Greater than or equal to 0	-

Table 4-168 response_data parameter description

Parameter	Type	Description
coreness	List<Map<String,Integer>>	Coreness value ($\text{coreness} \geq k$) of each vertex. The format is as follows: [{vertexId:corenessValue},...], where vertexId is of the string type. corenessValue is of the integer type.

4.6.2.5 K-hop

Table 4-169 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
k	Yes	Number of hops	Integer	1 to 100	-
source	Yes	Vertex ID	String	-	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
mode	No	Direction <ul style="list-style-type: none"> • OUT: Hop from the outgoing edges • IN: Hop from the incoming edges • All: Hop from edges in both directions 	String	OUT, IN, or ALL	OUT

Table 4-170 response_data parameter description

Parameter	Type	Description
vertices	List	ID of the vertex within k hops. The format is as follows: [vertexId,...], where vertexId 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)

4.6.2.6 Shortest Path

Table 4-171 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	<p>Empty or character string</p> <ul style="list-style-type: none"> Empty: The default weight and distance of edges are 1. Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. <p>NOTE The weight of an edge must be greater than 0.</p> <p>This parameter is not supported for graphs of the 100-billion-edge type.</p>	-
directed	No	Whether to consider the edge direction	Boolean	<p>The value can be true or false.</p> <p>For graphs of the 100-billion-edge type, only the value true is supported.</p>	false
timeWindow	No	Time window used for time filtering	Json	<p>For details, see Table 4-172.</p> <p>NOTE timeWindow does not support the shortest path with weight. That is, parameters timeWindow and weight cannot be both specified.</p>	-

Table 4-172 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.	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
filterType	No	Filtering by vertex or edge	String	V : filtering by vertex E : filtering by edge BOTH : filtering by vertex and edge	BOTH
startTime	No	Start time	String	Date character string or timestamp	-
endTime	No	End time	String	Date character string or timestamp	-

Table 4-173 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

4.6.2.7 All Shortest Paths

Table 4-174 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	true or false	false

Table 4-175 response_data parameter description

Parameter	Type	Description
paths	List	All shortest 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

4.6.2.8 Filtered Shortest Path

Request

- Parameter description

Table 4-176 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 false .

- 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 4-177 response_data parameter description

Parameter	Mandatory	Type	Description
path	Yes	List	Vertex result set. If the last layer of filters is vertex filtering, the data contains vertices.
source	Yes	String	Source vertex ID
target	Yes	String	Target vertex ID

Parameter	Mandatory	Type	Description
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 4-178 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.
job_id	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"
}
```

4.6.2.9 SSSP

Table 4-179 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	true or false	true

Table 4-180 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: [{vertexId:distanceValue},...], where vertexId is of the string type. distanceValue is of the double type.
source	String	Source vertex ID

4.6.2.10 Shortest Path of Vertex Sets

Table 4-181 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, Alice, Nana . 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, Alice, Nana . The maximum ID number is 100000.	-
directed	No	Whether to consider the edge direction	Boolean	true or false	false
timeWindow	No	Time window used for time filtering	Json	For details, see Table 4-182 .	-

Table 4-182 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	V : filtering by vertex E : filtering by edge BOTH : filtering by vertex and edge	BOTH
startTime	No	Start time	String	Date character string or timestamp	-
endTime	No	End time	String	Date character string or timestamp	-

Table 4-183 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

4.6.2.11 n-Paths

Table 4-184 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	true or false	false
n	No	Number of paths	Integer	1 to 100	10
k	No	Number of hops	Integer	1 to 10	5

Table 4-185 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

4.6.2.12 Closeness Centrality

Table 4-186 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	ID of the vertex to be calculated	String	-	-

Table 4-187 response_data parameter description

Parameter	Type	Description
closeness	Double	Closeness centrality degree
source	String	Vertex ID to be calculated

4.6.2.13 Label Propagation

Table 4-188 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
coveragence	No	Convergence	Double	A real number between 0 and 1 (excluding 0 and 1)	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	1000

Parameter	Mandatory	Description	Type	Value Range	Default Value
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"> Empty: Each vertex is allocated with a unique initialization label. This method is applicable to scenarios where no vertex label information exists. 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. <p>NOTE If the value of initial 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 4-189 response_data parameter description

Parameter	Type	Description
community	List	Community corresponding to each vertex. The format is: [{vertexId:communityId},...] where vertexId is of the string type. communityId is of the string type.

4.6.2.14 Louvain

Table 4-190 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
convergence	No	Convergence	Double	A real number between 0 and 1 (excluding 0 and 1)	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	100
weight	No	Weight of an edge	String	Empty or null character string <ul style="list-style-type: none"> • Empty: The default weight and distance of edges are 1. • Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	weight

 **NOTE**

For details about algorithm iterations and convergence, see [Iterations and Convergence of PageRank](#).

Table 4-191 response_data parameter description

Parameter	Type	Description
modularity	Double	Modularity
community_number	Integer	Number of communities

Parameter	Type	Description
community	List	Community corresponding to each vertex. The format is: <code>[{vertexId:communityId},...]</code> where vertexId is of the string type. communityId is of the string type.

4.6.2.15 Link Prediction

Table 4-192 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	-	-
target	Yes	Target vertex ID	String	-	-

Table 4-193 response_data parameter description

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID
link_prediction	Double	Link prediction result

4.6.2.16 Node2vec

Table 4-194 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

Parameter	Mandatory	Description	Type	Value Range	Default Value
dim	No	Mapping dimension	Integer	An integer between 1 and 200 (including 1 and 200)	50
walkLength	No	Random walk length	Integer	An integer between 1 and 100 (including 1 and 100)	40
walkNumber	No	Number of random walk steps of each vertex.	Integer	An integer between 1 and 100 (including 1 and 100)	10
iterations	No	Number of iterations	Integer	An integer between 1 and 100 (including 1 and 100)	10

Table 4-195 response_data parameter description

Parameter	Type	Description
embedding	List	<p>Vector representation of each vertex mapped to the Euclidean space. The format is as follows:</p> <p>[{vertexId:vectorValue}]</p> <p>where</p> <p>vertexId is of the string type.</p> <p>vectorValue: is a euclidean vector, for example, [-0.485, -0.679, 0.356].</p>

4.6.2.17 Real-time Recommendation

Table 4-196 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 source_limit). 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 (excluding 0 and 1)	0.85
N	No	Total number of walk steps	Integer	1 to 200000	10000
nv	No	Parameter indicating that the walk process ends ahead of schedule: minimum number of access times of a potential recommended vertex NOTE If a vertex is accessed during random walk and the number of access times reaches nv , the vertex will be recorded as the potential recommended vertex.	Integer	1 to 10	5

Parameter	Mandatory	Description	Type	Value Range	Default Value
np	No	Parameter indicating that the walk process ends ahead of schedule: number of potential recommended vertices NOTE If the number of potential recommended vertices of a source vertex reaches np , 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. NOTE <ul style="list-style-type: none"> 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. If the value is not null, vertices with the label are filtered from the calculation result. 	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
restricted	No	Whether to accept invalid source vertices restricted=true: If a vertex that does not exist in the graph is passed to sources , an error is reported. restricted=false: A vertex that does not exist in the graph can be passed to sources . However, if all source vertices do not exist, an error is reported.	Boolean	true or false	true

Table 4-197 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: [{vertexId: scoreValue} ,...] where vertexId is of the string type. scoreValue is of the double type.
sources	List	ID of the source vertex

4.6.2.18 Common Neighbors

Table 4-198 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	-	-
target	Yes	Target vertex ID	String	-	-

Table 4-199 response_data parameter description

Parameter	Type	Description
vertices	List	Common neighbor vertices. The format is as follows: [vertexId ,...], where vertexId is of the string type
common_neighbors	Integer	Number of common neighbor vertices
source	String	Source vertex ID
target	String	Target vertex ID

4.6.2.19 Connected Component

 NOTE

This algorithm can run without specifying its **parameters**.

Table 4-200 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. The format is [{vertexId:communityId},...]. where vertexId is of the string type. communityId is of the string type.

4.6.2.20 Degree Correlation

Table 4-201 response_data parameter description

Parameter	Type	Description
degree_correlation	Double	Degree correlation

4.6.2.21 Triangle Count

Table 4-202 Parameter description

Parameter	Mandatory	Description	Type	Value Range
statistics	No	Whether to export only the total statistical result. <ul style="list-style-type: none"> true: Export only the statistical result. false: Export the number of triangles corresponding to each vertex. 	Boolean	true or false . The default value is true .

Table 4-203 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: [{vertexId : vertexTriangleCount},...], where vertexId is of the string type. vertexTriangleCount is of the integer type.

4.6.2.22 Cluster Coefficient

Table 4-204 response_data parameter description

Parameter	Type	Description
cluster_coefficient	Double	Cluster coefficient

4.6.2.23 Common Neighbors of Vertex Sets

Table 4-205 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, Alice, Nana . 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, Mike,Amy . 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>true or false</p> <ul style="list-style-type: none"> false: 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. true: 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. <p>NOTE For graphs of the hundred-billion-edge type, only the value true is supported.</p>	true

Table 4-206 response_data parameter description

Parameter	Type	Description
vertices	List	Common neighbor vertices. The format is as follows: [vertexId,...], where vertexId is of the string type.

Parameter	Type	Description
common_neighbors	Integer	Number of common neighbors

4.6.2.24 All Shortest Paths of Vertex Sets

Table 4-207 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, Alice, Nana . 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, Alice, Nana . The maximum ID number is 100000.	-
directed	No	Whether to consider the edge direction	Boolean	true or false . It is a Boolean value.	false

Table 4-208 response_data parameter description

Parameter	Type	Description
paths	List	All shortest paths between the source vertex and target vertex. The format is as follows: [[path1],[path2]]
source	String	Source ID of a path
target	String	Target ID of a path

4.6.2.25 Filtered Circle Detection (2.2.15)

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 4-209 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, Alice, Nana.
n	No	Upper limit of the number of enumerated circles that meet the filter criteria	Integer	[1,100000]	100

Parameter	Mandatory	Description	Type	Value Range	Default Value
statistics	No	Whether to export the number of circles that meet the filter criteria	Boolean	true or false	false
batch_number	No	Number of source vertices for batch processing	Integer	[1,1000]	10
output_format	No	Output format	String	vertexId , edgeId , or edgeObject	edgeObject
filters	Yes	Filter criteria. Each element in the array corresponds to a filter.	Json	-	-

Table 4-210 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	out , in , or both	out
edge_filter	No	Filter criteria for the current layer. For details, see Table 4-268 in the Filtered-query API .	Json	-	-
vertex_filter	No	Filter criteria of vertices at the current layer. For details, see Table 4-268 in the Filtered-query API .	Json	-	-
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 4-211 response_data parameter description

Parameter	Mandatory	Type	Description
circles	Yes	List	Set of circles found. The format is [[circle1], [circle2], ...] . The circle format is as follows: <ul style="list-style-type: none"> • If output_format is edgeObject, the format is [{"source": sourceId, "target": targetId, "index": edgeIndex}, ...], where sourceId, targetId, and edgeIndex are of the string type. • If output_format is edged, the format is [sourceId-targetId-edgeIndex,...], where sourceId-targetId-edgeIndex is of the string type. • If output_format is vertexId, the format is [vertexId, ...], where vertexId is of the string type.
runtime	Yes	Double	Algorithm running time
n	Yes	Integer	Maximum number of enumerated circles
circle_number	No	Integer	When statistics is set to true , the number of circles that meet filter criteria is displayed.

4.6.2.26 Subgraph Matching (2.2.16)

Table 4-212 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 size_t 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, 1,2\n2,3 .
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, 1,BP\n2,FBP\n3,CP .
directed	No	Whether to consider the direction of the graph	Boolean	The value can be true or false . The default value is true .
n	No	Maximum number of subgraphs to be searched for	Integer	The value range is [1,100000]. The default value is 100 .
batch_number	No	Number of queries processed in batches each time	Integer	The value range is [1,1000000]. The default value is 10000 .
statistics	No	Whether to display the number of all subgraphs that meet the conditions	Boolean	The value can be true or false . The default value is false .

Table 4-213 response_data parameter description

Parameter	Mandatory	Type	Description
subgraphs	Yes	List	Subgraphs with the same pattern of the pattern_graph . The value is in the [[subgraph1],[subgraph2], ...] format. Each subgraph is in the [vertex1,vertex2, ...] format, where vertex is of the string type. The vertices of each subgraph correspond to those of pattern_graph .
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 statistics is set to true , the total number of graphs that meet query conditions is displayed.

4.6.2.27 Filtered All Pairs Shortest Paths (2.2.17)

Table 4-214 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.	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
directed	No	Whether the edges are directed	Boolean	The value can be true or false .	false
cutoff	No	Maximum length	Integer	1-100	6
path_limit	No	Maximum number of paths	Integer	<ul style="list-style-type: none"> For synchronous tasks: The value ranges from 1 to 100000. The default value is 100000. For asynchronous tasks: The value ranges from 1 to 1000000. The default value is 1000000. 	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 4-215 response_data parameter description

Parameter	Type	Description
batch_paths	List	Batch paths. Format: [paths_element,...] where Paths_element indicates the path from a source to a target. The format is as follows: { "paths": [["Alice", "Janet", "Sue", "Serena", "Bonnie"]], "source": "Alice", "target": "Bonnie" },
paths_number	Integer	Number of paths

4.6.2.28 Filtered All Shortest Paths (2.2.17)

Parameters

Table 4-216 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	- -	-
target	Yes	Target vertex ID	String	-	-
directed	No	Whether the edges are directed	Boolean	The value can be true or false .	false

Table 4-217 response_data parameter description

Parameter	Type	Description
paths	List	Paths between the source and target vertices. The format is as follows: [[path1],[path2]] where For the format of each path, see Shortest Path .
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 4-218 response_data parameter description

Parameter	Type	Description
paths	List	Paths between the source and target vertices. The format is as follows: [[path1],[path2]] where For the format of each path, see Shortest Path .
paths_number	Integer	Number of paths
source	String	Source vertex ID
target	String	Target vertex ID

4.6.2.29 Topicrank (2.2.20)

Table 4-219 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Node 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 true or false .	false

Parameter	Mandatory	Description	Type	Value Range	Default Value
only_neighbors	No	Whether to display only the neighboring vertices of the sources	Boolean	The value can be true or false .	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	The value ranges from 1 to 2000.	1000
directed	No	Whether the edges are directed	Boolean	The value can be true or false .	true
num_thread	No	Number of threads	Integer	The value ranges from 1 to 40.	4

Table 4-220 response_data parameter description

Parameter	Type	Description
topicrank	List	TopicRank value of each vertex. The format is as follows: [{vertexId:rankValue},...], where vertexId is of the string type. rankValue is of the double type.

4.6.2.30 Filtered n-Paths (2.2.22)

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 4-221 Body format

Field	Mandatory	Type	Description
algorithmName	Yes	String	The value is filtered_n_paths .
parameters	Yes	JSON format	For details about the format, see Table 4-222 .
filters	Yes	JSON Array	Filter criteria. Each element in the array corresponds to a filter. For details about the format, see Table 4-223 .

Table 4-222 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 4-223 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 4-224 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: ["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",  
          "57",  
          "56"  
        ]  
      ],  
      "runtime": 0.000308,  
      "source": "111",  
      "path_length": 6,  
      "data_offset": 0,  
      "paths_number": 3,  
      "data_total_size": 3,  
      "target": "56"  
    }  
  },  
  "status": "success"  
}
```

4.7 Path APIs

4.7.1 Querying Path Details

Function

This API is used to query the path details. All possible paths will be listed.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail
- Parameter description

Table 4-225 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Table 4-226 Request body parameter description

Parameter	Mandatory	Type	Description
paths	Yes	List	Set of paths to be queried

Parameter	Mandatory	Type	Description
directed	No	Boolean	Whether the querying path is directional or non-directional: true : directional false : non-directional default=false

Response

Table 4-227 Parameter description

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
data	No	Json	Query results. This parameter is left blank when the query fails.

Table 4-228 data parameter description

Parameter	Type	Description
outputs	Json	Query results containing the paths

Parameter	Type	Description
paths	List	<p>Collection of paths that contain detailed vertex and edge information, in JSONArray format</p> <p>NOTE In the returned paths:</p> <ul style="list-style-type: none"> • If the vertex does not exist, the corresponding position is {}. • If there is no edge between vertices, the corresponding position is {"edges": []}.

- Response example (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"]
        }
      }
    ]
  }
}
```


- Response example (failed request)
Http Status Code: 400

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

Return Value

- Normal
200
- Abnormal

Table 4-229 Return code for failed requests

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.

4.8 Graph Statistics APIs

4.8.1 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

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details={labelDetails}
- Parameter description

Table 4-230 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
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 false . If this parameter is set to true , the numbers of vertices and edges under each label are returned.

Request

- Request example

```
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 [Constraints of Using Service Plane APIs](#).

Response

Table 4-231 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.
data	No	Json	Query results. This parameter is left blank when the request fails.

Table 4-232 data parameter description

Parameter	Mandatory	Type	Description
vertexNum	Yes	Integer	Number of vertices in a graph
edgeNum	Yes	Integer	Number of edges in a graph
labelDetails	Yes	Json	Numbers of vertices and edges under each label. To properly display this parameter, create vertex and edge indexes based on Table 4-233 .

Table 4-233 Description of each element in **labelDetails** when the execution is successful

Parameter	Mandatory	Type	Description
labelInVertex	No	Json	Number of vertices under each label. If the number of vertices under a label is 0, the label is not displayed. To include this parameter in the response, create an index by referring to Creating an Index . During index creation, set indexType to GlobalCompositeVertexIndex , set hasLabel to true , and leave indexProperty blank.
labelInEdge	No	Json	Number of edges under different labels. If the number of edges under a label is 0, the label is not displayed. To include this parameter in the response, create an index by referring to Creating an Index . During index creation, set indexType to GlobalCompositeEdgeIndex , set hasLabel to true , and leave indexProperty blank.
errorMessage	No	String	System prompt. If execution succeeds, this parameter is left blank. If execution fails, this parameter is used to display the error message.
errorCode	No	String	System prompt. If execution succeeds, this parameter is left blank. If execution fails, this parameter is used to display the error code.

- 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
  }
}

```

- Example response of a successful request

Http Status Code: 200

```

{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}

```

- Example response of a failed request

Http Status Code: 400

```

{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8001"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-234 Return code for failed requests

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.

4.8.2 Querying the Graph Version

Function

This API is used to query the graph version.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/version
- Parameter description

Table 4-235 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

Table 4-236 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.

Parameter	Mandatory	Type	Description
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.
version	No	String	Query results. This parameter is left blank when the request fails.

- Response example (successful request)

```
Http Status Code: 200
{
  "version":"2.0.0"
}
```

- Response example (failed request)

```
Http Status Code: 404
{
  "errorMessage":"Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-237 Return code for failed requests

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.

4.9 Subgraph Operation APIs

4.9.1 Querying a Subgraph

Function

This API is used to query the subgraphs formed by the entered vertices and edges between the vertices.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?action_id=query
- Parameter description

Table 4-238 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example


```
{
  "vertices": [
    "Ray",
    "Ella",
    "Lethal Weapon"
  ]
}
```
- Parameter description

Table 4-239

Parameter	Mandatory	Type	Description
vertices	Yes	String	Vertex ID array of the subgraph NOTE 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

Table 4-240 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.
data	No	Json	The data field is contained when the query is successful, and the data field contains the subgraph query result. NOTE 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.

- Response example (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"
  }
}
]
}

```

- Response example (failed request)

```

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

```

Return Value

- Normal
200
- Abnormal

Table 4-241 Return code for failed requests

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.

4.9.2 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, which is used for resource isolation. Obtain the project ID by following the instructions in "Obtaining the Project ID".
graph_name	Yes	String	Graph name

Request

- Example request**
 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
  }
}
```

- Parameters

Table 4-242 Request body parameter description

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name Available values are as follows: <ul style="list-style-type: none"> connected_component kcore

Parameter	Mandatory	Type	Description
parameters	Yes	JSON	Algorithm parameters <ul style="list-style-type: none"> • connected_component • Kcore
subgraphCreator	Yes	Json	Subgraph parameters For details, see subgraphCreator parameters .

Table 4-243 subgraphCreator parameters

Parameter	Mandatory	Type	Description
name	No	String	Type of the subgraph creator. Currently, only filtered is available.
parameters	Yes	JSON	The parameter format varies according to the name of the subgraph creator.

Table 4-244 Parameters when name=filtered

Parameter	Mandatory	Type	Description
vertex_filter	No	String	Vertex filtering criteria
edge_filter	No	String	Edge filtering criteria

Response

Table 4-245 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.

Parameter	Mandatory	Type	Description
job_id	No	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	No	Integer	Task type. This parameter is left blank if the request fails.

- Response example (successful request)

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-246 Return code for failed requests

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.

4.10 Job Management APIs

4.10.1 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

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=*offset*&limit=*limit*
- Parameter description

Table 4-247 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
job_id	Yes	String	Job ID
offset	No	Integer	Offset of a query. The default value is 0 .
limit	No	Integer	Maximum number of records that can be queried. The default value is 100000 .

Request

- Request example
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-248 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.
status	No	String	Returned job status after the query is successful. Possible values: <ul style="list-style-type: none"> • pending • running • success • failed This parameter is left blank when the query fails.
data	No	Json	Algorithm execution result. This parameter is left blank when the query fails.

- Parameter description

Table 4-249 data parameter description

Parameter	Mandatory	Type	Description
vertices	No	List	Vertex-associated algorithm result
edges	No	List	Edge-associated algorithm result
outputs	No	Json	Other results
data_return_size	No	Integer	Number of records returned after a query
data_offset	No	Integer	Result offset of a query
data_total_size	No	Integer	Total amount of result data generated by asynchronous jobs

- Response example (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"
}

```

- **Response example (failed request)**

```

Http Status Code: 400
{
  "errorMessage": "can not find job, jobId is 9440a7ebXXXXXXXXXXXXXXXXXXXX2d079a67001679122",
  "errorCode": "GES.8301"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-250 Return code for failed requests

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.

4.10.2 Canceling a Job

Function

This API is used to cancel a job that has been submitted.

You need to use the job IDs returned by asynchronous APIs to cancel jobs. These APIs include querying vertices or edges that meet filter criteria, running algorithms, adding indexes, and running read-only Cypher queries. If the job execution is complete or fails, you cannot cancel this job.

 **NOTE**

Only jobs returned by APIs used to query vertices or edges that meet filter criteria, run algorithms, and add indexes can be canceled. If you cancel jobs returned by other APIs, an error message stating **Unsupported Operation** is reported.

URI

- URI format
DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}
- Parameter description

Table 4-251 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
job_id	Yes	String	Job ID
graph_name	Yes	String	Graph name

Request

- Request example

```
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 [Constraints of Using Service Plane APIs](#).

Response

- Parameter description

Table 4-252 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": "can not find job to cancel, id is 9440a7ebXXXXXXXXXXXXXXXXXXXX2d079a67001679122",
  "errorCode": "GES.8303"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-253 Return code for failed requests

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.

4.10.3 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 Queries \(2.2.16\)](#)

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result
- Parameter description

Table 4-254 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name
job_id	Yes	String	ID of the job corresponding to the response

Request

- Parameter description

Table 4-255 Request 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 Table 4-256 .

Table 4-256 obsParameters parameter description

Parameter	Mandatory	Type	Description
accessKey	Yes	string	AK value
secretKey	Yes	string	SK value

- Request example
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result

- Exporting the result to OBS

```
{
  "exportPath": "demo_movie/",
  "fileName": "louvain",
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

Response

- Parameter description

Table 4-257 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.
job_id	No	String	ID of an asynchronous job. You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane (1.0.0) .

- Response example (successful request)

```

HttpStatusCode: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
    
```

- Response example (failed request)

```

HttpStatusCode: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8011"
}
    
```

Return Value

- Normal
200
- Abnormal

Table 4-258 Return code for failed requests

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.

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
```

4.10.4 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.

NOTE

A maximum of 100 thousand records can be returned for graphs of the 100-billion-edge type.

URI

- URI format
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}
- Parameter description

Table 4-259 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
offset	No	Integer	Offset of a query. The default value is 0 . This parameter is not supported for graphs of the 100-billion-edge type.
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 100000 . This parameter is not supported for graphs of the 100-billion-edge type.

Request

- Request example
GET /ges/v1.0/1/graphs/movie/jobs/status

Response

Table 4-260 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.
result	Yes	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .
jobs	No	Json	Job status list stored in the system. If execution succeeds, this parameter is contained in the response. The following table describes the structure of a single jobs field.

Table 4-261 Job status structure

Parameter	Mandatory	Type	Description
jobId	Yes	String	JobId
request	Yes	Json	Request content, including the command, URL, and body.
status	Yes	String	Job status. The value can be pending , running , or complete .

- Response example of a successful request (for 100-billion-edge)

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"
}

```

- Example response of a failed request (for 100-billion-edge)

```

Http Status Code: 400
{
  "errorMessage": "graph : movidde not exist",
  "errorCode": "GES.8000",
  "result": "failed"
}

```

Return Value

- Normal
200
- Abnormal

Table 4-262 Return code for failed requests

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.

4.11 Querying K Hop Vertices or Edges Using a Filter

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

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query
- Parameter description

Table 4-263 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

- Synchronization

```
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"
  ]
}
```

- Asynchronization

```
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"
  ]
}
```

```

    }
  }
},
"full_path": false,
"vertices": [
  "tr_10"
]
}

```

- Nested property_filter

```

{
  "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"
  ]
}

```

- Request body parameter description

 NOTE

- If **executionMode** is set to **sync**, the number of returned vertices cannot exceed 100,000.

Table 4-264 Request body parameter description

Parameter	Mandatory	Type	Description
execution Mode	No	String	<ul style="list-style-type: none"> • sync: synchronous • async: asynchronous The default value is sync , indicating synchronous response.
vertices	Yes	Array of Json	List of IDs of source vertices to be queried
query_type	No	String	Possible values are Default , AllVertices , SimpleEdges , Path . <ul style="list-style-type: none"> • Default indicates the default mode, that is, the <i>k</i> hop is returned. • AllVertices returns details about all vertices within <i>k</i> hops. • SimpleEdges returns all edges within <i>k</i> hops, contain only the ID and label information of the edges. • Path returns the path information, that is, the set of paths.
by	No	Array of Json	Specified output field. This parameter is valid only when query_type is set to Default or AllVertices . 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 vertices is selected. For details, see Table 4-265 .
filters	Yes	Array of Json	Filter criteria. Each element in the array corresponds to a filter. For details about the formats, see Table 4-266 .
full_path	No	Boolean	Whether to return a complete path. The default value is false . <ul style="list-style-type: none"> • If the value is true, the paths from the source vertex to all leaf vertices are returned. • If the value is false, the paths from the source vertex to the leaf vertices at layer k are returned.

Parameter	Mandatory	Type	Description
visualized	No	Boolean	Whether to enable visualization. The default value is false . In asynchronous mode: <ul style="list-style-type: none"> When visualized is false, the job query result is returned on multiple pages. When visualized is true, the job query result is returned on one page.
restricted(2.2.28)	No	Boolean	Whether to verify the input. The default value is true . <ul style="list-style-type: none"> true: If vertices contains vertices that do not exist, the query exits and an error is reported. false: The system filters out vertices that do not exist and then performs the query task.

Table 4-265 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 4-266 Filters element formats

Parameter	Mandatory	Type	Description
operator	Yes	String	<p>Query type. Possible values:</p> <ul style="list-style-type: none"> • inV: incoming vertex • outV: outgoing vertex • bothV: incoming and outgoing vertices • vertex: 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. • in: incoming edge • out: outgoing edge • both: incoming and outgoing edges • edge: all edges. Filtering is available only at the first layer. The usage is similar to that of vertices <p>The query result of the previous layer is the input of the next layer.</p> <ul style="list-style-type: none"> • If the result of the previous layer is a vertex, the corresponding operations can be inV, outV, bothV, in, out, and both. • If the result of the previous layer is an edge, the corresponding operation can be inV, outV, and bothV.
vertex_filter	No	Json String	This parameter is optional when operator is set to inV , outV , or bothV . For details about the formats, see Table 4-268 .
edge_filter	No	Json String	This parameter is optional when operator is set to in , out , or both . For details about the formats, see Table 4-268 .

Table 4-267 by element formats

Parameter	Mandatory	Type	Description
id	No	Boolean	Whether to output the ID. The default value is false .
label	No	Boolean	Whether to output the label. The default value is false .

Parameter	Mandatory	Type	Description
properties	No	Boolean	Whether to output properties. The default value is false .
selectedProperties	No	Array of String	When properties is set to true , 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 4-268 property_filter element formats

Parameter	Mandatory	Type	Description
leftvalue	Yes	Json String	Left value. For details about the formats, see Table 4-269 .

Parameter	Mandatory	Type	Description
predicate	Yes	String	<p>Filtering type. The supported operations include:</p> <ul style="list-style-type: none"> • =: equal to • !=: not equal to • <: less than • ≤: Less than or equal to • >: greater than • ≥: greater than or equal to • &: and • : or • HAS/HASNOT: whether the property exists • CONTAIN/NOTCONTAIN: whether the property value contains the right value. • SUBSET: The right value is a subset of the property value. • IN/NOTIN: whether the left value and right value have an intersection • PREFIX: The right value is the prefix of the left value. • FUZZY: fuzzy match • REGEX: expression match • SUBSTRING: The right value is a sub-string of the left value. • CISUBSTRING: sub-string that ignores cases
rightvalue	Yes	Json String	Right value. For details about the formats, see Table 4-270 .

Table 4-269 leftvalue element formats

Parameter	Mandatory	Type	Description
label_name	No	String	If label is used as the filter criterion, label_name can be selected and the value is labelName . Set the value field of rightvalue to the label name.

Parameter	Mandatory	Type	Description
property_name	No	String	If property is used as the filter criterion, property_name can be selected and the value is the property name. Set the value field of rightvalue to the property value.
id	No	String	If the vertex ID is filtered, this parameter is optional.
property_filter	No	Json String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Table 4-270 rightvalue element formats

Parameter	Mandatory	Type	Description
value	Yes	String	<ul style="list-style-type: none"> If label is used as the filter criterion, the value is the label name. If property is used as the filter criterion, the value is the property name.
property_filter	No	Json String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Table 4-271 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
=! =/<</<=>/>=	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

- Synchronous response
 - Parameter description

Table 4-272 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.
data	No	Json	Query results. This parameter is left blank when the query fails.

Table 4-273 data parameter description

Parameter	Mandatory	Type	Description
vertices	No	List	Vertex result set. If the last layer of filters is vertex filtering, the data contains vertices.
edges	No	List	Edge result set. If the last layer of filters is edge filtering, the data contains edges.
paths	No	List	Path set. This parameter is available only when with_path is set to true . For details about the formats, see Table 4-274 .

Table 4-274 path parameter description

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID
target	Yes	String	Target vertex ID
index	Yes	String	Edge index
label	Yes	String	Edge label

– Response example (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"
}
]
}

```

- Response example (failed request)

```

Http Status Code: 400
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8806"
}

```

- Asynchronous response
 - Parameter description

Table 4-275 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.
job_id	No	String	ID of the algorithm execution job. This parameter is left blank when the request fails.

Parameter	Mandatory	Type	Description
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

– Response example (successful request)

Http Status Code: 200

```
{
  "jobId": "6622f13c-4b88-45f5-89a9-eea096647c4a",
  "jobType": 1
}
```

– Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "executionMode is not correct, it should be sync or async",
  "errorCode": "GES.8806"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-276 Return code for failed requests

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.

4.12 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.

URL

- URI format
POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties
- Parameter description

Table 4-277 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
POST http://Endpoint/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
}

```

- Parameter description
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 4-278 Request body parameter description

Parameter	Mandatory	Type	Description
vertexsetPath	Either vertexsetPath or edgesetPath 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 vertexsetPath exists	Json	Label of a vertex and list of properties to be updated in a vertex file, in JSONArray format
edgesetPath	Either vertexsetPath or edgesetPath 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 edgesetPath exists	Json	Label of an edge and list of properties to be updated in an edge file, in JSONArray format
targetProperties	Mandatory if edgesetPath exists	Json	Indicates property information used to distinguish duplicate edges in the edge file, in JSONArray format.

Parameter	Mandatory	Type	Description
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 ("). It is used to enclose a field if the field contains separators or line breaks.
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 4-256 .
vertexFileContainLabel	No	Boolean	Whether the vertex file contains label information. This parameter is optional. The default value is true .

Table 4-279 vertexProperties parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Name of a label
properties	Yes	Json	Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the vertex file.

Table 4-280 edgeProperties parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Name of a label
properties	Yes	Json	Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the edge file.

Table 4-281 targetProperties parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Name of a label
properties	Yes	Json	Edge ID properties, in JSONArray format. Currently, only one property is supported.

Response

- Parameter description

Table 4-282 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.
job_id	No	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

- Response example (successful request)

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "parameter format error",
  "errorCode": "GES.8013"
}
```

Return Value

- Normal
200
- Abnormal

Table 4-283 Return code for failed requests

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.

4.13 Deleting Vertices and Edges by Files

Function

This API is used to delete vertices and edges by reading the files.

URL

- URI format
POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file
- Parameters

Table 4-284 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

```
POST http://Endpoint/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": "XXXXXXX",
"secretKey": "XXXXXXX"
}
}
}

```

- Parameters

Table 4-285 Request body parameter description

Parameter	Mandatory	Type	Description
vertexsetPath	Either vertexsetPath or edgesetPath 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 vertexsetPath or edgesetPath 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	Json	Indicates property information used to distinguish duplicate edges in the edge file, in JSONArray format. For details, see Table 4-286 .
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 ("). They are used to enclose a field if the field contains separators or line breaks.
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 4-256 .

Table 4-286 targetProperties parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Name of a label
properties	Yes	Json	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

- 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.

Parameter	Mandatory	Type	Description
job_id	No	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane .

- Response example (successful request)

Http Status Code: 200

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

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "parameter format error",
  "errorCode": "GES.8013"
}
```

Return Value

- Normal

200

- Abnormal

Table 4-287 Return code for failed requests

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.

4.14 Performing 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](#).

URL

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
- Parameter description

Table 4-288 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation.
graph_name	Yes	String	Graph name

Request

- Request example
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
  }]
}
```
- Parameter description

Table 4-289 Request body parameter description

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements. statements parameter description describes the format of each element.

Table 4-290 statements parameter description

Parameter	Mandatory	Type	Description
statement	Yes	String	Cypher statement

Parameter	Mandatory	Type	Description
parameters	Yes	Json	Cypher statement parameters, which are used for parameterized query. By default, this parameter is left blank. For details, see Parameterized queries .
resultDataContents	No	String or List	Format of the returned result. You can set one or more formats. Available values are row , graph , and raw (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.
executionMode (2.2.23)	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async 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 Querying Job Status on the Service Plane .
limit (2.2.23)	No	Int	Maximum number of results of the asynchronous query. This parameter is valid only when executionMode is sync . The default value is 100000 .

 NOTE

In asynchronous mode (**executionMode** is set to **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

- Parameter description

Table 4-291 Parameter description

Parameter	Mandatory	Type	Description
results	Yes	List	A List. Each element is the return result of a Cypher statement.
errors	Yes	List	A list. Each element in the list contains the code and message information in the form of character strings.

Table 4-292 Elements of the results parameter

Parameter	Mandatory	Type	Description
columns	Yes	List	Name of a returned field
data	Yes	List	Returned data value. Each element indicates a record.
stats	No	Json	Addition, deletion, and modification statistics
plan	No	Json	If the cypher statement contains the explain prefix, this field contains the query plan. Otherwise, this field is not displayed.

Table 4-293 Elements of the data parameter

Parameter	Mandatory	Type	Description
row	No	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	No	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	No	Json	Returns information in graph format. This parameter is displayed only when resultDataContents contains graph .
raw(2.2.2.7)	No	List	Returns information in raw format. This parameter is displayed only when resultDataContents contains raw .

- Example of a successful synchronous request

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": []
}
```

Table 4-294 Response parameters of **stats** elements

Parameter	Mandatory	Type	Description
contains_updates	Yes	Boolean	Whether data is modified during the query
edges_created	Yes	Integer	Number of created edges
edges_deleted	Yes	Int	Number of deleted edges
labels_set	Yes	Integer	Number of labels that have been set
properties_set	Yes	Integer	Number of properties that have been set

Parameter	Mandatory	Type	Description
vertices_created	Yes	Integer	Number of created vertices
vertices_deleted	Yes	Integer	Number of deleted vertices

- Example of a successful asynchronous request

Http Status Code: 200

```
{
  "results": [
    {
      "columns": [
        "jobId",
        "jobType"
      ],
      "data": [
        {
          "row": [
            "b64a5846-e306-4f87-b0f1-d595ee2a9910",
            1
          ],
          "meta": [
            null,
            null
          ]
        }
      ]
    }
  ],
  "errors": []
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "results": [],
  "errors": [
    {
      "code": "GES.8904",
      "message": "Label index in vertices is not found."
    }
  ]
}
```

Response Code

- Normal

200

- Abnormal

Table 4-295 Return code for failed requests

Response Code	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.

Response Code	Description
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.

Cypher Prerequisites

NOTE

The prerequisites are not required for graphs of the 100-billion-edge type.

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. The following is an example:

- Example for creating a vertex label index:

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "cypher_vertex_index",
  "indexType": "GlobalCompositeVertexIndex",
  "hasLabel": "true",
  "indexProperty": []
}
```

- Example for creating an edge label 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 must create two indexes (vertex label index and edge label index) at the same time to use Cypher for query.
2. If a vertex index or an edge index whose **hasLabel** is **true** and **indexProperty** is empty exists in the graph, you do not need to create the vertex index or edge index again.
3. The API for creating an index is an asynchronous API. To check whether the index is successfully created, use the [API for querying the job status](#).

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

Operation	Cypher Statement
Querying information by specifying filtering criteria	match(n:user) where n.userid>=5 return n
Grouping and aggregation	match(n:movie) return n.genres, count(*)
Deduplication	match(n:movie) return distinct n.genres
Sorting	match(n:movie) return n order by n.movieid
Creating vertices	create (n:user{userid:1}) return n
Creating edges	match (n:user{userid:15}),(m:movie{movieid:10}) create (n)-[r:rate]->(m)
Deleting vertices	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

- Clauses supported by Cypher

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 following Cypher clauses:

Table 4-296 Clauses supported by Cypher

Clause	Support	Example
match	Partially supported	match (n:movie) return n
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
create	Supported	create (n:user{_ID_: 'Jack' }) return n

Clause	Support	Example
delete	Supported	match (n:movie)<-[r]-(m:user) delete r
set	Supported	match (n:user{userid:0}) set n.gender='M' 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) NOTE Union is available for graphs smaller than 10 billion edges only.

 **NOTE**

1. Currently, merge, foreach, and optional operations are not supported. Cypher statements cannot be used to 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 the sorting of the list type. When Cardinality of the property value is not single, the sorting result is unknown.
- Parameter-based queries

Cypher supports parameter-based query. 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**

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**
- 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 mapping between other types and Cypher is as follows:

Table 4-297 Mapping between types of GES and Cypher

GES Type	Cypher type	Description
char	String	-
char_array	String	-
string	String	-
enum	String	The Cypher syntax does not provide the enumeration-related syntax. During Cypher query, enum is output as a string. When Cypher is used to set properties, values that are not in the enumeration list fail to be set. NOTE This parameter is not supported by graphs of the 100-billion-edge type.
date	Temporal	Currently, dates can be input and output in the GES date format, but cannot be input by calling the Cypher date function.

Table 4-298 Special types supported by Cypher

Type	Support	Example
Node	Supported	match (n) return n limit 10
Relationship	Supported	match (n)-[r]->(m) return r limit 10
List	Supported	return [1,2,3] as li
Map	Supported	match (n)-->(m) return {start:id(n), end:id(m)}

Type	Support	Example
Path	Supported	<pre>match p=(n1)-[:friends*1..2]-(n2) return p limit 10</pre> <p>NOTE This parameter is not supported by graphs of the 100-billion-edge type.</p>
Point, Spatial	Not supported	-

 **NOTE**

For the special types listed in [Special types supported by Cypher](#), only the list type is used to match multi-value properties in GES. Other types cannot be set to the value of a property of a vertex or edge by using the set statement.

- Expressions

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
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

Operation Type	Expression	Example
	[]	match(n:user) return n['userid'] with [1, 2, 3, 4] as list return list[0] with [1, 2, 3, 4] as list return list[0..1] match p=(n)-->(m) return [x in nodes(p) where x.gender='F' id(x)] This operator is not supported for graphs of the 100-billion-edge type.

 **NOTE**

The where clause in Cypher queries does not support arithmetic operators and regular expression matching.

- Functions and procedures
 - Function

During grouping, aggregation, and vertex and edge operations, Cypher supports a series of functions. Currently, the following functions are supported:

a. Aggregate function

Currently, the **count** and **collect** aggregate functions are supported.

Function	Description	Example
count	Returns the total number of results.	match (n) return count(*) match (n) return count(n.userid)
collect(2.2.17)	Collects results in a list.	match (n:movie) return n.genres, collect(n) as movieList

b. Common function

Based on the types of input parameters, common functions are classified into vertex and edge functions, path functions, list functions, and value functions.

Table 4-299 Vertex and edge functions

Function	Description	Example
id	Obtains the ID of a vertex.	match (n) return id(n)
labels	Obtains labels of a vertex.	match (n) return labels(n)

Function	Description	Example
type	Obtains the label of an edge.	match(n)-[r]->(m) return type(r)
degree(2.2.2.6)	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)

Table 4-300 Path functions (2.2.19)

Function	Description	Example
nodes	Obtains the list of vertices on a path.	match p=(n)-[:friends*1..2]->(m) return nodes(p) This function is not supported by graphs of the 100-billion-edge type.
relationships	Obtains the list of edges on a path.	match p=(n)-[:friends*1..2]->(m) return relationships(p) This function is not supported by graphs of the 100-billion-edge type.
length	Obtains the path length.	match p=(n)-[:friends*1..2]->(m) return length(p)

Table 4-301 List functions

Function	Description	Example
head	Obtains the first element of a list.	with [1,2,3,4] as list return head(list)
last	Obtains the last element of a list.	with [1,2,3,4] as list return last(list)
size	Obtains the list length.	with [1,2,3,4] as list return size(list)

Table 4-302 Value functions

Function	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)

Table 4-303 Predicate functions (2.2.19)

Function	Description	Example
all	If all elements meet the expression, true is returned.	all (x in p where x>1)
any	If any element meets the expression, true is returned.	any (x in p where x>1)
none	If all elements cannot meet the expression, true is returned.	none (x in p where x>1)
single	If only one element meets the expression, true is returned.	single (x in p where x>1)

Table 4-304 Algorithm expression (2.3.2)

Function	Description	Example
shortestPath	Returns the shortest path between two vertices.	The following expression returns the shortest path between the given vertices n and m . The direction is m to n, and the edge label is rate : with n,m, shortestPath((n)-[:rate*]-(m)) as p return p
allShortestPaths	Returns all shortest paths between two vertices.	The following expression returns all shortest paths between the given vertices n and m : with n,m, allShortestPaths((n)-[*]-(m)) as p return p

 **NOTE**

- Aggregate functions, such as sum(), avg(), max(), and min(), are not available for 100-billion-edge graphs. Mathematical functions, such as sin() and cos(), will be available in the future.

▪ Procedure

Currently, GES supports the following procedures.

Procedure	Statement
Obtaining graph mode information	call db.schema()
Obtaining vertex labels	call db.labels()
Obtaining edge labels	call db.relationshipTypes() This procedure is not supported by graphs of the 100-billion-edge type.
Querying the Cypher statements that are being executed	call dbms.listQueries()
Terminating a Cypher statement based on queryId	call dbms.killQuery('queryId')

 **NOTE**

Function and procedure names are case sensitive and must be in lower camel case.

- Vertex ID compatibility
 - Cypher does not provide the syntax for setting the ID when a vertex is added. In GES, an ID of the character string type is required to uniquely identify a vertex. To be compatible with the Cypher syntax, the current create statement uses a special identifier `_ID_` to specify the ID of a vertex. 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.

- Parallel edge processing policy
When using Cypher to add edges, you can add duplicate edges. The duplicate edges are defined as 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
This API is used to query the structure of the generated schema (obtained from OBS).
- Query statement
 - Name: Schema structure query
 - Command: **call db.schema ()**
 - Note:
If you did not call the API for generating the schema structure, this API returns all labels in the schema file.
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.

4.15 Granular Permission Control APIs

4.15.1 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. 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 4-305 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
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 [Constraints of Using Service Plane APIs](#).

- Request body parameter description

Table 4-306 Request body parameter description

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name
userId	Yes	String	Grantee
acl	Yes	JsonArray	Authorization details
type	Yes	String	Permission type. The value can be read , write , traverse , or schema .
detail	Yes	JsonArray	Permission details
label	Yes	String	Label name
properties	No	List	Properties

Response

- Parameter description

Table 4-307 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"
}
```

Response Code

- Normal
200
- Abnormal

Table 4-308 Return code for failed requests

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.

4.15.2 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 4-309 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

 NOTE

- SERVER_URL**: Address for accessing a graph. For details about its value, see [Constraints of Using Service Plane APIs](#).
- Request body parameter description

Table 4-310 Request body parameter description

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name
userId	Yes	String	Grantee
acl	Yes	JsonArray	Authorization details
type	Yes	String	Permission type. The value can be read , write , traverse , or schema .
detail	Yes	JsonArray	Permission details
label	Yes	String	Label name
properties	No	List	Properties

Response

- Parameter description

Table 4-311 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"
}

```

Response Code

- Normal
200
- Abnormal

Table 4-312 Return code for failed requests

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.

4.15.3 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 4-313 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For details, see Obtaining a Project ID .
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 [Constraints of 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"
      }
    ]
  }
}
```

Response Code

- Normal
200
- Abnormal

Table 4-314 Return code for failed requests

Response Code	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	No operation permission.

Response Code	Description
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	Service unavailable.

5 GES Metrics

Function

This chapter describes metrics reported by GES as well as their namespaces, lists, and dimensions. You can use APIs to query the metric information generated for GES.

Namespace

SYS.GES

Metrics

Table 5-1 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. Unit: %	0 to 100 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. Unit: %	0 to 100 Value type: Float	GES instance
ges003_average_import_rate	Average Import Rate	Average rate of importing vertices or edges to a graph instance Unit: count/s	0 to 400000 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 Unit: count	≥ 0 Value type: Integer	GES instance
ges005_average_response_time	Average Response Time	Average response time of requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges006_min_response_time	Minimum Response Time	Minimum response time of requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges007_max_response_time	Maximum Response Time	Maximum response time of requests received by a graph instance Unit: ms	≥ 0 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. Unit: count	≥ 0 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 Unit: ms	≥ 0 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 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. Unit: count	≥ 0 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 Unit: ms	≥ 0 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 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. Unit: count	≥ 0 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 Unit: ms	≥ 0 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 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. Unit: count	≥ 0 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 Unit: ms	≥ 0 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 Value type: Integer	GES instance
ges020_computer_resource_usage	Computing Resource Usage	Computing resource usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges021_memory_usage	Memory Usage	Memory usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges022_iops	IOPS	Number of I/O requests processed by each graph instance per second Unit: count/s	≥ 0 Value type: Integer	GES instance
ges023_bytes_in	Network Input Throughput	Data input to each graph instance per second over the network Unit: byte/s	≥ 0 Value type: Float	GES instance
ges024_bytes_out	Network Output Throughput	Data sent to the network per second from each graph instance Unit: byte/s	≥ 0 Value type: Float	GES instance
ges025_disk_usage	Disk Usage	Disk usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges026_disk_total_size	Total Disk Size	Total data disk space of each graph instance Unit: GB	≥ 0 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 Unit: GB	≥ 0 Value type: Float	GES instance
ges028_disk_read_throughput	Disk Read Throughput	Data volume read from the disk in a graph instance per second Unit: byte/s	≥ 0 Value type: Float	GES instance
ges029_disk_write_throughput	Disk Write Throughput	Data volume written to the disk in a graph instance per second Unit: byte/s	≥ 0 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 Unit: second	≥ 0 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 Unit: second	≥ 0 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 Unit: count	≥ 0 Value type: Integer	GES instance

Dimensions

Table 5-2 Dimensions

Key	Value
instance_id	GES instance

Mapping Between Task Types and Names

Table 5-3 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

6 Appendix

6.1 Status Codes

[Table 6-1](#) describes status codes.

Table 6-1 Status codes

Stat us Cod e	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 accepted, but the processing has not been completed.
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 us Cod 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. 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. 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 us Cod e	Message	Description
403	Forbidden	The server has received the request and understood it, but the server is refusing to respond to it. 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. 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. 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. 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. 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. 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 Content-Length .
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.

Stat us Cod 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 Retry-After 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 Expect 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 Retry-After 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. 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 Timeout parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

6.2 Error Codes

6.2.1 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 are unable to identify the cause of an error, contact technical personnel and provide the error code so that we can help you solve the problem as soon as possible.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in .

Table 6-2 Error codes

Status Code	Error Code	Error Message	Description	Solution
400	GES.0001	Incorrect parameter.	Incorrect parameter.	<ol style="list-style-type: none"> 1. Check whether the project ID or graph ID in the URL is correct. 2. Check whether the request header is correct, for example, whether X-Auth-Token is correct.
400	GES.0016	Resource not found	Resource not found.	<ol style="list-style-type: none"> 1. Check whether the project ID in the URL is the same as the project ID of the token. 2. Ensure that the project ID in the URL is the same as that of the token.
400	GES.7000	The graph does not exist or has been deleted.	The graph does not exist or has been deleted.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 2. Check whether the project ID or graph ID in the URL is correct.
400	GES.7001	The graph is not running.	The graph is not running.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 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 200.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7002	The graph is being backed up.	The graph is being backed up.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 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 903.
400	GES. 7003	The graph is being stopped or is stopped.	The graph is being stopped or is stopped.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 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 900 or 901.
400	GES. 7004	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. 7005	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. 7006	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 personnel.
400	GES. 7007	The job does not exist.	The job does not exist.	Check whether the job ID in the URL is correct.
400	GES. 7008	The job is stopped.	The job is stopped.	Jobs cannot be stopped repeatedly.
400	GES. 7009	The job operation is not supported.	The job operation is not supported.	The job operation is not supported.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7010	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. 7011	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. 7012	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. 7013	The graph name already exists.	The graph name already exists.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 2. Query the graph list returned in the preceding step and check whether the name field in the request body already exists.
400	GES. 7014	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 action_id is check-schema .
400	GES. 7015	The graph is not running or is stopped.	The graph is not running or is stopped.	<ol style="list-style-type: none"> 1. Call the graph query API to query all graphs. 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 900 status.
400	GES. 7016	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 s Code	Error Code	Error Message	Description	Solution
400	GES. 7017	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. 7018	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. 7019	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. 7020	The VPC does not exist.	The VPC does not exist.	Check whether the VPC ID in the request body exists.
400	GES. 7021	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. 7022	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. 7023	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. 7024	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"> 1. Call the backup query API to query all backups of a specified graph. 2. Check whether the backup ID or graph ID in the URL is correct.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7027	Failed to create an agency.	Failed to create an agency.	<ol style="list-style-type: none"> 1. Assign the security administrator role to the user group to which the user belongs. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 7028	Failed to authorize an agency.	Failed to authorize an agency.	<ol style="list-style-type: none"> 1. Assign the security administrator role to the user group to which the user belongs. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 7029	The agency resource exceeds the quota limit.	The agency resource exceeds the quota limit.	Check whether the agency resource reaches the quota limit on the .
400	GES. 7030	Agency query error.	Agency query error.	Check the error message for detailed information.
400	GES. 7031	Invalid binding type of an EIP.	Invalid binding type of an EIP.	<p>Confirm the EIP binding type. The value can be either of the following:</p> <ul style="list-style-type: none"> • bind_existing
400	GES. 7032	The EIP resource exceeds the quota limit.	The EIP resource exceeds the quota limit.	Check whether the EIP resource reaches the quota limit on the VPC page.
400	GES. 7033	Invalid EIP ID.	Invalid EIP ID.	If the EIP binding type is set to bind_existing , ensure that the EIP ID exists.
400	GES. 7034	Resources in the current AZ have been sold out.	Resources in the current AZ have been sold out.	Switch to another AZ and try again.
400	GES. 7035	Invalid region code.	Invalid region code.	Enter the correct region code.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7036	The target version is earlier than the current version.	The target version is earlier than the current version.	A graph can only be upgraded to a later version.
400	GES. 7037	The graph is not in the Stopped state.	The graph is not in the Stopped state.	Check whether the graph is in the Stopped state.
400	GES. 7040	Failed to back up a graph.	Failed to back up a graph.	Failed to restore a graph from the backup you select.
400	GES. 7041	Insufficient permission.	Insufficient permission.	Insufficient permission.
400	GES. 7042	The graph is being created.	The graph is being created.	The graph is being created.
400	GES. 7048	Invalid graph operation.	Invalid graph operation.	Check whether the value of action_id is start , stop , import-graph , export-graph , clear-graph , or upgrade .
400	GES. 7049	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. 7050	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. 7051	Components at the IaaS layer are faulty.	Components at the IaaS layer are faulty.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, obtain logs and analyze them for fault locating.
400	GES. 7052	Invalid CPU architecture of the graph instance.	Invalid CPU architecture of the graph instance.	Check whether the value of arch is set to x86_64 or aarch64 when the graph is created.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7054	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.
400	GES. 7056	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. 7057	Invalid graph flavor for scale-out.	Invalid graph flavor for resize.	graphSizeTypeIndex in the resize request body can be set to 2 , 3 , 4 , or 5 , indicating the ten-million-edge, hundred-million-edge, billion-edge, or ten-billion-edge graph.
400	GES. 7059	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. 7061	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. 7062	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 errorMessage .
400	GES. 7063	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 errorMessage .

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 7064	Failed to create the ECS.	Failed to create the ECS.	See the returned errorMessage 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.
400	GES. 7065	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. 7066	Failed to query the flavor.	Failed to query the flavor.	Check whether the flavor ID configured on the GES management plane exists.
400	GES. 7067	Insufficient ECS quota.	Insufficient ECS quota.	Check whether the flavor ID configured on the GES management plane exists.
400	GES. 7068	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. 7069	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. 7070	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.

6.2.2 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 are unable to identify the cause of an error, contact technical personnel and provide the error code so that we can help you solve the problem as soon as possible.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in .

Table 6-3 Error codes

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8000	Incorrect parameter format.	Incorrect parameter format.	Check whether the request body is the same as that described in the document.
400	GES. 8001	Failed to query graph statistics.	Failed to query graph statistics.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8002	Graph statistics query error.	Graph statistics query error.	<ol style="list-style-type: none"> 1. Check whether the token has expired. If it is expired, obtain a new one. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8005	Incorrect parameter.	Incorrect parameter.	<ol style="list-style-type: none"> 1. Check whether the project ID in the URL is correct. 2. Check whether the request header is correct, for example, whether X-Auth-Token is correct.
400	GES. 8006	Invalid resource access.	Invalid resource access.	Check whether the project ID in the URL is correct.
400	GES. 8007	Invalid token.	Invalid token.	Check whether the token is correct.
400	GES. 8008	An error occurs in the underlying authentication system.	An error occurs in the underlying authentication system.	Try again later or contact technical personnel.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8011	Failed to export a graph.	Failed to export a graph.	<ol style="list-style-type: none"> 1. Check whether the graph name is correct. 2. Check whether the export path is correct. 3. Check whether the account has the OBS write permission.
400	GES. 8012	Failed to clear a graph.	Failed to clear a graph.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8013	Failed to incrementally import data to the graph.	Failed to incrementally import data to the graph.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8020	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. 8101	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. 8102	Invalid label for edge filtering queries.	Invalid label for edge filtering queries.	Check whether the labels are in the correct JSON format.
400	GES. 8103	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.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8104	Invalid edge filtering query sequence.	Invalid edge filtering query sequence.	Check whether the edge filtering query sequence is valid.
400	GES. 8105	Failed to query edges that meet filter criteria.	Failed to query edges that meet filter criteria.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8106	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. 8107	Failed to query edge details.	Failed to query edge details.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8108	Edge details query error.	Edge details query error.	Try again later or contact technical personnel.
400	GES. 8109	Invalid edge filtering query operator.	Invalid edge filtering query operator.	Ensure that the values of edge filtering query operators are in , out , both , and edge .
400	GES. 8110	Parameter edges cannot be left blank.	Parameter edges cannot be left blank.	Check whether the value of edges in the batch edge query request body is empty.
400	GES. 8201	Invalid label for vertex filtering queries.	Invalid label for vertex filtering queries.	Check whether the labels are in the correct JSON format.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8202	Invalid filter criteria for vertex queries.	Invalid filter criteria for vertex queries.	<ol style="list-style-type: none"> 1. Check whether propertyName of the vertex query API is left blank. 2. Check whether values of the vertex query API is left blank. 3. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8203	Both the condition and label of vertex filtering queries are empty.	Both the condition and label of vertex filtering queries are empty.	Ensure that the condition and label of vertex filtering queries are not both empty.
400	GES. 8204	Failed to query vertices that meet filter criteria.	Failed to query vertices that meet filter criteria.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8205	Invalid vertex filtering query sequence.	Invalid vertex filtering query sequence.	In the vertex filtering query API, orderValue must be set to incr or decr .
400	GES. 8206	Both vertexid and vertextids exist.	Both vertexid and vertextids exist.	vertexid and vertextids cannot coexist.
400	GES. 8207	Both vertexid and vertextids are empty.	Both vertexid and vertextids are empty.	The vertexid or vertextids parameter is empty.
400	GES. 8208	Incorrect vertextids format.	Incorrect vertextids format.	Check whether vertextids is a JSON array.
400	GES. 8209	Failed to query vertex details.	Failed to query vertex details.	Check whether the graph name exists.

Statu s Code	Error Code	Error Message	Description	Solution
500	GES. 8210	Vertex details query error.	Vertex details query error.	Try again later or contact technical personnel.
400	GES. 8211	Invalid vertex filtering query operator.	Invalid vertex filtering query operator.	Ensure that values of vertex filtering query operators are inV , outV , bothV , and vertex .
400	GES. 8212	Failed to delete the vertex label.	Failed to delete the vertex label.	Check whether the label exists.
400	GES. 8213	Failed to add the vertex label.	Failed to add the vertex label.	Check whether the label exists.
400	GES. 8214	Parameter vertices cannot be left blank.	Parameter vertices cannot be left blank.	Check whether the value of vertices in the batch vertex query request body is empty.
400	GES. 8220	Failed to update the vertex properties.	Failed to update the vertex properties.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8221	Failed to update the edge properties.	Failed to update the edge properties.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8301	Failed to query a job.	Failed to query a job.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8302	Job query error.	Job query error.	Try again later or contact technical personnel.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8303	Failed to terminate a job.	Failed to terminate a job.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8304	Job termination error.	Job termination error.	Try again later or contact technical personnel.
400	GES. 8401	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. 8402	Failed to run the algorithm.	Failed to run the algorithm.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. Check whether the graph name in the algorithm running API is correct. 3. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8403	Algorithm running error.	Algorithm running error.	Try again later or contact technical personnel.
400	GES. 8404	Invalid algorithm running format.	Invalid algorithm running format.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8501	The Gremlin command is not supported.	The Gremlin command is not supported.	Replace the unsupported Gremlin statements: tryNext, explain, and tree.
400	GES. 8502	Failed to find the Gremlin configuration file.	Failed to find the Gremlin configuration file.	Try again later or contact technical personnel.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8503	Gremlin query failed.	Gremlin query failed.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES. 8504	Gremlin query error.	Gremlin query error.	Try again later or contact technical personnel.
400	GES. 8505	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. 8506	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. 8601	Gremlin service unavailable.	Gremlin service unavailable.	Try again later or contact technical personnel.
500	GES. 8602	Engine service unavailable.	Engine service unavailable.	Try again later or contact technical personnel.
400	GES. 8603	Failed to create an index	Failed to create an index	<ol style="list-style-type: none"> 1. Check whether the index name contains only letters, digits, hyphens (-), and underscores (_). 2. Check whether the index parameter type complies with that specified by the API.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8604	Failed to delete an index	Failed to delete an index	<ol style="list-style-type: none"> 1. Check whether the graph name is correct. 2. Check whether the index name is correct. 3. Check whether Method type of the request is delete.
400	GES. 8605	Failed to query an index	Failed to query an index	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8609	The request body for querying path details is invalid.	The request body for querying path details is invalid.	<ol style="list-style-type: none"> 1. Check whether the graph name is correct. 2. Check whether the parameter format of the API for querying path details is correct. 3. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8610	The path parameter of the request body for querying path details is invalid.	The path parameter of the request body for querying path details is invalid.	<ol style="list-style-type: none"> 1. Check whether the parameter format of the API for querying path details is correct. 2. Check whether the mandatory parameters of the API for querying path details are set. 3. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8611	Failed to query path details.	Failed to query path details.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES. 8612	The operation of querying path details is not supported.	The operation of querying path details is not supported.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES. 8801	Failed to add a label to metadata.	Failed to add a label to metadata.	<ol style="list-style-type: none"> 1. Check whether the label to be added already exists. 2. Check whether the format of the parameter for adding the label is correct. 3. Check whether the mandatory parameters for adding the label are set.
400	GES. 8803	Failed to query the metadata.	Failed to query the metadata.	<ol style="list-style-type: none"> 1. Check whether the graph to be queried exists. 2. Check whether the value of graph_name in the API for querying graph metadata is correct.
500	GES. 8804	Metadata query error.	Metadata query error.	Try again later or contact technical personnel.
400	GES. 8806	K-Hop query with filter criteria failed.	K-Hop query with filter criteria failed.	<ol style="list-style-type: none"> 1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

6.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain a project ID by calling an API

For details about API authentication, see [Making a Management Plane API Request](#).

The following is an example response. The value of **id** under **projects** is the project ID. The following is an example response. If GES is deployed in the *xxx* region, the value of **name** in the response body is *xxx*, and the value of **id** in **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 management console.
2. Click the username and select **My Credentials** from the drop-down list.

In the project list on the **My Credentials** page, select a project ID based on the region and project. For example, select **5a3314075bfa49b9ae360f4ecd333695** for **Project ID**.

6.4 Obtaining an Account ID

An account ID (**domain-id**) is required for some URLs when an API is called. To obtain the account ID, perform the following operations:

1. Register an account and log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

On the **My Credentials** page, view the **Account ID**.