

ServiceStage

API Reference

Issue 01
Date 2024-01-09



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	2
1.4 Constraints.....	2
1.5 Concepts.....	2
2 API Overview.....	4
3 Calling APIs.....	17
3.1 Making an API Request.....	17
3.2 Authentication.....	20
3.3 Response.....	21
4 Application Management V2 APIs.....	23
4.1 Meta.....	23
4.1.1 Obtaining All Supported Runtime Systems of Application Components.....	23
4.1.2 Obtaining All Supported Flavors of Application Resources.....	25
4.2 Environment.....	27
4.2.1 Creating an Environment.....	27
4.2.2 Obtaining All Environments.....	32
4.2.3 Modifying Environment Information.....	35
4.2.4 Deleting an Environment Based on the Environment ID.....	38
4.2.5 Obtaining Environment Details Based on the Environment ID.....	40
4.2.6 Modifying Environment Resources.....	42
4.3 Application.....	47
4.3.1 Creating an Application.....	47
4.3.2 Obtaining All Applications.....	49
4.3.3 Modifying Application Information.....	52
4.3.4 Deleting an Application Based on the Application ID.....	55
4.3.5 Obtaining Application Details Based on the Application ID.....	56
4.3.6 Modifying Application Configurations.....	59
4.3.7 Deleting Application Configurations.....	62
4.3.8 Obtaining Application Configurations.....	64
4.4 Component.....	66

4.4.1 Creating an Application Component.....	67
4.4.2 Obtaining All Components of an Application.....	74
4.4.3 Modifying Component Information Based on the Component ID.....	79
4.4.4 Deleting a Component Based on the Component ID.....	87
4.4.5 Obtaining Component Information Based on the Component ID.....	89
4.5 Instance.....	93
4.5.1 Creating an Application Component Instance.....	94
4.5.2 Obtaining All Component Instances.....	106
4.5.3 Querying the Operations Performed on a Component Instance.....	111
4.5.4 Modifying a Component Instance.....	114
4.5.5 Deleting a Component Instance.....	118
4.5.6 Querying Instance Details Based on the Instance ID.....	120
4.5.7 Obtaining Component Instance Snapshots.....	125
4.6 Deployment Jobs.....	128
4.6.1 Obtaining Job Details.....	128
5 Application Management V3 APIs.....	134
5.1 Environment.....	134
5.1.1 Creating an Environment.....	134
5.1.2 Obtaining All Environments.....	137
5.1.3 Deleting an Environment Based on the Environment ID.....	141
5.1.4 Modifying an Environment Based on the Environment ID.....	142
5.1.5 Obtaining Environment Details Based on the Environment ID.....	146
5.1.6 Modifying an Environment Resource Based on the Environment ID.....	149
5.1.7 Querying an Environment Resource Based on the Environment ID.....	152
5.2 Application.....	154
5.2.1 Creating an Application.....	154
5.2.2 Obtaining All Applications.....	157
5.2.3 Modifying Application Information Based on the Application ID.....	160
5.2.4 Deleting an Application Based on the Application ID.....	163
5.2.5 Obtaining Application Details Based on the Application ID.....	165
5.2.6 Obtaining Application Configurations Based on the Application ID.....	167
5.2.7 Modifying Application Configurations Based on the Application ID.....	170
5.2.8 Deleting Application Configurations Based on the Application ID.....	173
5.3 Component.....	175
5.3.1 Creating an Application Component.....	175
5.3.2 Obtaining All Components of an Application.....	192
5.3.3 Obtaining All Components.....	198
5.3.4 Modifying Component Information by Component ID.....	204
5.3.5 Deleting a Component Based on the Component ID.....	220
5.3.6 Obtaining Component Information Based on the Component ID.....	222
5.3.7 Delivering Component Tasks by Component ID.....	237
5.3.8 Obtaining Records by Component ID.....	240

5.4 Runtime System.....	243
5.4.1 Querying a Runtime System Stack.....	243
6 Git Repository Access APIs.....	246
6.1 Obtaining a Git Repository Authorization List.....	246
6.2 Obtaining an Authorization Redirection URL.....	248
6.3 Creating OAuth Authorization.....	250
6.4 Creating Private Token Authorization.....	254
6.5 Creating Password Authorization.....	256
6.6 Deleting Repository Authorization.....	259
6.7 Obtaining a Repository Namespace.....	260
6.8 Obtaining Repository Information Based on the Clone URL.....	262
6.9 Obtaining All Projects in a Namespace.....	264
6.10 Creating a Software Repository Project.....	266
6.11 Obtaining a Project Branch.....	268
6.12 Obtaining a Project Tag.....	270
6.13 Creating a Project Tag.....	271
6.14 Deleting a Project Tag.....	274
6.15 Obtaining Project Commits.....	275
6.16 Obtaining a Project Hook.....	277
6.17 Creating a Project Hook.....	279
6.18 Deleting a Project Hook.....	281
6.19 Obtaining a Repository File Directory.....	283
6.20 Obtaining Repository File Contents.....	285
6.21 Creating a Repository File.....	287
6.22 Modifying Repository File Contents.....	289
6.23 Deleting a Repository File.....	292
7 CSE API.....	294
7.1 API Calling.....	294
7.2 Dynamic Configuration.....	294
7.2.1 Importing Configurations.....	294
7.2.2 Exporting Configurations.....	299
7.3 Engine Management.....	303
7.3.1 Querying Flavors Supported by an Exclusive Microservice Engine.....	303
7.3.2 Querying the Exclusive Microservice Engine List.....	305
7.3.3 Creating an Exclusive Microservice Engine.....	311
7.3.4 Querying Details About an Exclusive Microservice Engine.....	315
7.3.5 Deleting an Exclusive Microservice Engine.....	323
7.3.6 Querying Details About an Exclusive Microservice Engine Job.....	326
8 ServiceComb API.....	331
8.1 API Calling.....	331
8.2 Authentication.....	331

8.2.1 Obtaining the User Token of an Exclusive Microservice Engine.....	331
8.3 Microservice.....	333
8.3.1 Querying Information About a Microservice.....	334
8.3.2 Deleting Definition Information About a Microservice.....	338
8.3.3 Querying Information About All Microservices.....	341
8.3.4 Creating Static Information for a Microservice.....	346
8.3.5 Deleting Static Information About Microservices in Batches.....	353
8.3.6 Modifying Extended Attributes of a Microservice.....	357
8.3.7 Querying the Unique Service or Schema ID of a Microservice.....	360
8.4 Schema.....	364
8.4.1 Querying a Microservice Schema.....	364
8.4.2 Modifying a Microservice Schema.....	367
8.4.3 Querying All Schema Information About a Microservice.....	369
8.5 Microservice Instance.....	373
8.5.1 Registering a Microservice Instance.....	373
8.5.2 Querying a Microservice Instance Based on service_id.....	378
8.5.3 Deregistering a Microservice Instance.....	382
8.5.4 Querying Details About a Microservice Instance.....	385
8.5.5 Modifying the Extended Information About a Microservice Instance.....	389
8.5.6 Modifying Status of a Microservice Instance.....	392
8.5.7 Sending Heartbeat Information.....	395
8.5.8 Querying a Microservice Instance by Filter Criteria.....	398
8.5.9 Querying Microservice Instances in Batches.....	403
8.6 Dependency.....	410
8.6.1 Querying All Providers of a Microservice.....	411
8.7 Configuration Management.....	415
8.7.1 Creating a Configuration.....	415
8.7.2 Modifying a Configuration.....	419
8.7.3 Querying a Configuration.....	421
8.7.4 Querying the Configuration List.....	424
8.7.5 Deleting a Configuration.....	428
8.7.6 Deleting All Configurations.....	431
9 Out-of-Date APIs.....	434
9.1 Querying Configurations.....	434
9.2 Creating a Dependency Between Services.....	437
9.3 Reporting Service Metrics.....	439
10 Data Structure.....	444
10.1 Parameter Description.....	444
10.1.1 HealthCheck.....	444
10.1.2 MicroServiceInstance.....	445
10.1.3 HeartbeatSetElement.....	446
10.1.4 InstanceHbRst.....	446

10.1.5 DelServicesResponse.....	446
10.1.6 MetricData.....	446
10.1.7 FunctionMetricInfo.....	447
10.1.8 InstanceMetricInfo.....	448
10.2 Common Request Parameters.....	448
10.2.1 MicroService.....	448
10.2.2 Properties.....	450
10.2.3 DependencyMicroService.....	450
10.2.4 Rule.....	451
10.2.5 AddOrUpdateRule.....	452
10.2.6 DataCenterInfo.....	453
10.2.7 Schema.....	453
10.3 Common Response Parameters.....	453
10.3.1 WatchMicroServiceKey.....	454
10.3.2 SuccessdResponse.....	454
10.3.3 ServiceInfo.....	454
10.3.4 AggregateMetricInfo.....	455
10.3.5 ServiceDetail.....	456
10.3.6 TenantProject.....	457
10.3.7 Error.....	457
11 Permissions and Supported Actions.....	458
11.1 Introduction.....	458
12 Appendix.....	463
12.1 Status Codes.....	463
12.2 ServiceStage Error Codes.....	464
12.3 CSE Error Codes.....	468
12.4 ServiceComb Error Codes.....	474
12.5 Obtaining a Project ID.....	479
12.6 Obtaining an Account ID.....	480
13 Change History.....	481

1 Before You Start

1.1 Overview

ServiceStage is an application management and O&M platform that lets you deploy, roll out, monitor, and maintain applications all in one place. It supports technology stacks such as Java, PHP, Python, Node.js, Docker, and Tomcat, and supports microservice applications such as Apache ServiceComb Java Chassis (Java chassis) and Spring Cloud, making it easier to migrate enterprise applications to the cloud.

ServiceStage provides the following application programming interfaces (APIs):

- Application management APIs
- Git repository access APIs
- CSE API
- ServiceComb API

1.2 API Calling

ServiceStage supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Table 1-1](#).

Table 1-1 API calling

API	Call Method
Application management APIs	For details, see Calling APIs .
Git repository access APIs	For details, see Calling APIs .
CSE API	For details, see API Calling .

API	Call Method
ServiceComb API	<ul style="list-style-type: none">• For a professional microservice engine, see Calling APIs.• For an exclusive microservice engine, see API Calling.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions.

For details, see [Regions and Endpoints](#).

1.4 Constraints

- The number of resources that you can create is determined by your quota. For details, see [Quotas](#).
- For more constraints, see API description.

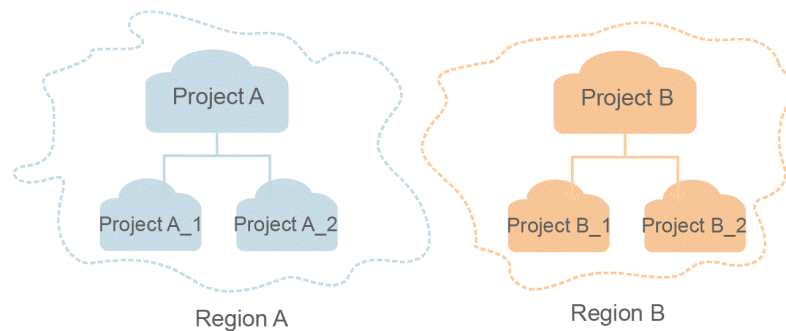
1.5 Concepts

- Account
An account is created upon successful registration with the cloud platform. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- User
A user is created in Identity and Access Management (IAM) to use cloud services. Each user has its own identity credentials (password and access keys).
An IAM user can view the account ID and user ID on the [My Credentials](#) page of the console. 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.
For details, see [Regions and Endpoints](#).
- AZ
AZs are physically isolated locations in a region, but are interconnected through an internal network for enhanced application availability.
- Project

Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

For details about how to check the project ID, see [Obtaining a Project ID](#).

Figure 1-1 Project isolating model



- Enterprise project

Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

For details about how to obtain the enterprise project ID and features, see [Enterprise Management User Guide](#).

2 API Overview

ServiceStage provides open APIs for developers and partners to develop, deploy, host, operate, and manage applications, helping them effectively innovate services at low costs and shorten the application rollout period.

For details about the APIs, see the following table.

Table 2-1 Overview

Type	Subtype	Description
Application management V3 APIs Use the following endpoint: servicestage.<regionname>.myhuaweicloud.com	Environment V3 APIs	APIs related to environments.
	Application V3 APIs	APIs related to applications.
	Component V3 APIs	APIs related to components.
	Runtime System V3 APIs	APIs related to runtime systems.
Application management V2 APIs Use the following endpoint: servicestage.<regionname>.myhuaweicloud.com NOTE Application management V2 APIs are about to go offline. You are advised to use the V3 APIs.	Meta V2 APIs	APIs related to application templates, runtime systems, and application resource specifications.
	Environment V2 APIs	APIs related to application environments.
	Application V2 APIs	APIs related to applications.
	Component V2 APIs	APIs related to application components.

Type	Subtype	Description
	Instance V2 APIs	APIs related to application component instances.
	Deployment Job V2 APIs	APIs related to deployment jobs.
Git repository access APIs Use the following endpoint: servicestage.<regionname>.myhuaweicloud.com	Git Repository Access APIs	APIs related to Git repository authorization, namespaces, projects, branches, tags, commits, hooks, repository file directories, and contents.
CSE APIs Use the following endpoint: cse.<regionname>.myhuaweicloud.com	<ul style="list-style-type: none">• Dynamic Configuration• Engine Management	<ul style="list-style-type: none">• Dynamic Configuration: provides APIs related to configuration import to and export from the configuration management center.• Engine Management: provides APIs related to exclusive microservice engine creation, query, and deletion.

Type	Subtype	Description
ServiceComb-native APIs	<ul style="list-style-type: none"> • Authentication • Microservice • Schema • Microservice Instance • Dependency • Configuration Management 	<ul style="list-style-type: none"> • Authentication: provides APIs related to microservice identity authentication. • Microservice: provides APIs related to microservice management. • Schema: provides APIs related to microservice schema query and upload. • Microservice Instance: provides APIs related to microservice instance registry and heartbeat reporting. • Dependency: provides APIs related to microservice dependency query. • Configuration Management: provides APIs related to configuration management of the configuration center.

Environment V3 APIs

Table 2-2 Environment APIs

API	Description
Creating an Environment	Create an environment.
Obtaining All Environments	Obtain all environments.
Deleting an Environment Based on the Environment ID	Delete an environment based on the environment ID.
Modifying an Environment Based on the Environment ID	Modify an environment based on the environment ID.
Obtaining Environment Details Based on the Environment ID	Obtain environment details based on the environment ID.

API	Description
Modifying an Environment Resource Based on the Environment ID	Modify an environment resource based on environment ID.
Querying an Environment Resource Based on the Environment ID	Query an environment resource based on environment ID.

Application V3 APIs

Table 2-3 Application APIs

API	Description
Creating an Application	Create an application.
Obtaining All Applications	Obtain all applications.
Modifying Application Information Based on the Application ID	Modify application information based on the application ID.
Deleting an Application Based on the Application ID	Delete an application based on the application ID.
Obtaining Application Details Based on the Application ID	Obtain application details based on the application ID.
Obtaining Application Configurations Based on the Application ID	Obtain application configurations based on the application ID.
Modifying Application Configurations Based on the Application ID	Modify application configurations based on the application ID.
Deleting Application Configurations Based on the Application ID	Delete application configurations based on the application ID.

Component V3 APIs

Table 2-4 Component APIs

API	Description
Creating an Application Component	Create an application component.
Obtaining All Components of an Application	Obtain all components of an application.
Obtaining All Components	Obtain all components.

API	Description
Modifying Component Information by Component ID	Modify component information based on the component ID.
Deleting a Component Based on the Component ID	Delete a component based on the component ID.
Obtaining Component Information Based on the Component ID	Obtain component information based on the component ID.
Delivering Component Tasks by Component ID	Deliver a component action based on the component ID.
Obtaining Records by Component ID	Obtain records based on the component ID.

Runtime System V3 APIs

Table 2-5 Runtime system APIs

API	Description
Querying a Runtime System Stack	Obtain runtime system information.

Meta V2 APIs

Table 2-6 Meta APIs

API	Description
Obtaining All Supported Runtime Systems of Application Components	Obtain all supported runtime systems of application components.
Obtaining All Supported Flavors of Application Resources	Obtain all supported flavors of application resources.

Environment V2 APIs

Table 2-7 Environment APIs

API	Description
Creating an Environment	Create an environment.
Obtaining All Environments	Obtain all environments.

API	Description
Modifying Environment Information	Modify an environment based on the environment ID.
Deleting an Environment Based on the Environment ID	Delete an environment based on the environment ID.
Obtaining Environment Details Based on the Environment ID	Obtain environment details based on the environment ID.
Modifying Environment Resources	Modify environment resources.

Application V2 APIs

Table 2-8 Application APIs

API	Description
Creating an Application	Create an application.
Obtaining All Applications	Obtain all created applications.
Modifying Application Information	Modify application information based on the application ID.
Deleting an Application Based on the Application ID	Delete an application based on the application ID.
Obtaining Application Details Based on the Application ID	Obtain application details based on the application ID.
Modifying Application Configurations	Add or modify application configurations.
Deleting Application Configurations	Delete application configurations.
Obtaining Application Configurations	Obtain application configurations.

Component V2 APIs

Table 2-9 Component APIs

API	Description
Creating an Application Component	Create an application component.
Obtaining All Components of an Application	Obtain all components of an application.

API	Description
Modifying Component Information Based on the Component ID	Modify component information based on the component ID.
Deleting a Component Based on the Component ID	Delete a component based on the component ID.
Obtaining Component Information Based on the Component ID	Obtain component information based on the component ID.

Instance V2 APIs

Table 2-10 Instance APIs

API	Description
Creating an Application Component Instance	Create a component instance.
Obtaining All Component Instances	Obtain all instances of a component.
Querying the Operations Performed on a Component Instance	Perform operations on a component instance.
Modifying a Component Instance	Modify a component instance.
Deleting a Component Instance	Delete a component instance.
Querying Instance Details Based on the Instance ID	Obtain instance details based on the instance ID.
Obtaining Component Instance Snapshots	Obtain component instance snapshots.

Deployment Job V2 APIs

Table 2-11 Deployment job API

API	Description
Obtaining Job Details	Obtain job details.

Git Repository Access APIs

Table 2-12 Git repository access APIs

API	Description
Obtaining a Git Repository Authorization List	Obtain the Git repository authorization list.
Obtaining an Authorization Redirection URL	Obtain an authorization redirection URL.
Creating OAuth Authorization	Create OAuth authorization.
Creating Private Token Authorization	Create private token authorization.
Creating Password Authorization	Create password authorization for a Git repository.
Deleting Repository Authorization	Delete repository authorization based on the name.
Obtaining a Repository Namespace	Obtain namespaces of a repository.
Obtaining Repository Information Based on the Clone URL	Obtain repository information based on the clone URL.
Obtaining All Projects in a Namespace	Obtain all projects in a namespace.
Creating a Software Repository Project	Create a software repository project.
Obtaining a Project Branch	Obtain a project branch.
Obtaining a Project Tag	Obtain a project tag.
Creating a Project Tag	Create a project tag.
Deleting a Project Tag	Delete a project tag.
Obtaining Project Commits	Obtain the latest ten project commits.
Obtaining a Project Hook	Obtain a project hook.

API	Description
Creating a Project Hook	Create a project hook.
Deleting a Project Hook	Delete a project hook.
Obtaining a Repository File Directory	Obtain a repository file directory.
Obtaining Repository File Contents	Obtain repository file contents.
Creating a Repository File	Create a repository file.
Modifying Repository File Contents	Modify repository file contents.
Deleting a Repository File	Delete repository file contents.

Dynamic Configuration

Table 2-13 Dynamic configuration APIs

API	Description
Importing Configurations	Import configurations to the configuration management center.
Exporting Configurations	Export configurations from the configuration management center.

Engine Management

Table 2-14 Engine management APIs

API	Description
Querying Flavors Supported by an Exclusive Microservice Engine	Query the flavors supported by an exclusive microservice engine.
Querying the Exclusive Microservice Engine List	Query the exclusive microservice engine list.
Creating an Exclusive Microservice Engine	Create an exclusive microservice engine.

API	Description
Querying Details About an Exclusive Microservice Engine	Query details about an exclusive microservice engine.
Deleting an Exclusive Microservice Engine	Delete an exclusive microservice engine.
Querying Details About an Exclusive Microservice Engine Job	Query details about an exclusive microservice engine job.

Authentication

Table 2-15 Authentication APIs

API	Description
Obtaining the User Token of an Exclusive Microservice Engine	<p>This API is intended only for exclusive microservice engines with security authentication enabled.</p> <p>Before accessing APIs of an exclusive microservice engine with security authentication enabled, you need to call this API to obtain a user token. This token is the unique credential for accessing the APIs.</p>

Microservice

Table 2-16 Microservice APIs

API	Description
Querying Information About a Microservice	Query the definition information about a microservice based on serviceld .
Deleting Definition Information About a Microservice	Delete the definition and related information about a microservice, and deregisters all instances of the microservice.
Querying Information About All Microservices	Query information about microservices that meet filter criteria.

API	Description
Creating Static Information for a Microservice	Create static information for a microservice before registering a microservice instance. The registered instance is associated with the static information based on the serviceld field. One service corresponds to multiple instances. The serviceld field can be customized. If serviceld is not customized, the system generates a random serviceld .
Deleting Static Information About Microservices in Batches	Delete the definitions and related information about microservices in batches, and deregisters all instances of the microservices.
Modifying Extended Attributes of a Microservice	Modify static information about a microservice To update some fields in the static information, input all static information (including the fields that do not need to be updated) in JSON format.
Querying the Unique Service or Schema ID of a Microservice	Query the unique serviceld or schema ID of a microservice based on filter criteria.

Schema

Table 2-17 Schema APIs

API	Description
Querying a Microservice Schema	Query a microservice schema based on service_id and schema_id .
Modifying a Microservice Schema	Modify a microservice schema based on schema_id .
Querying All Schema Information About a Microservice	Query all schema information (including schemald and summary) about a microservice.

Microservice Instance

Table 2-18 Microservice instance APIs

API	Description
Registering a Microservice Instance	Register a microservice instance after a microservice is created. Information about the instance must be provided during registration. instanceId can be customized. If it is customized, all contents are overwritten upon re-registration. If it is not customized, the system automatically generates an ID. If the endpoint content is duplicate, the original ID is used.
Querying a Microservice Instance Based on service_id	Query all instances of a microservice based on service_id after the instances are registered.
Deregistering a Microservice Instance	Deregister an instance based on instance_id after the instance is registered.
Querying Details About a Microservice Instance	Query details about an instance based on service_id and instance_id after the instance is registered.
Modifying the Extended Information About a Microservice Instance	Add or update the extended information about a microservice instance based on instance_id after the instance is registered.
Modifying Status of a Microservice Instance	Modify the status of a microservice instance based on instance_id after the instance is registered.
Sending Heartbeat Information	Send heartbeat information from service provider to service center, so that the service center can check whether service instances are normal.
Querying a Microservice Instance by Filter Criteria	Query a microservice instance based on microservice version rules or field filter criteria after the instance is registered.
Querying Microservice Instances in Batches	Query microservice instances in batches based on microservice version rules or field filter criteria after the instances are registered.

Dependency

Table 2-19 Dependency APIs

API	Description
Querying All Providers of a Microservice	Query all providers of a microservice based on consumer_id . For example, if service A depends on services B and C, you can query the information about services B and C based on service A ID.

Configuration Management

Table 2-20 Configuration management APIs

API	Description
Creating a Configuration	Create a configuration.
Modifying a Configuration	Modify a configuration.
Querying a Configuration	Query a configuration.
Querying the Configuration List	Query the configuration list.
Deleting a Configuration	Delete a configuration.
Deleting All Configurations	Delete all configurations.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a Representational State Transfer (REST) API request, and uses the Identity and Access Management (IAM) API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

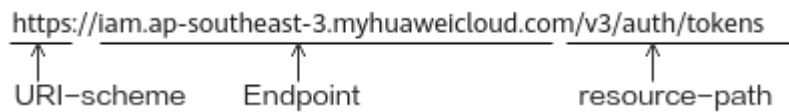
{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

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:** Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in the AP-Singapore region is iam.ap-southeast-3.myhuaweicloud.com.
- **resource-path:** Access path of an API for performing a specified operation. Obtain the value from the URI of an API. For example, the **resource-path** of the API for obtaining 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.

For example, to obtain an IAM token in the region, obtain the endpoint of IAM (iam.ap-southeast-3.myhuaweicloud.com) for this region and the **resource-path (/v3/auth/tokens)** in the URI of the API for **obtaining a user token**. Then, construct the URI as follows:

```
https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
```


Figure 3-1 Example URI**NOTE**

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

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**: requests a server resource 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. If the resource is unavailable, the PATCH method is used to create a resource.

For example, in the case of the API for [obtaining a user token](#), the request method is POST. The request is as follows:

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
```

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:

- **Content-Type**: 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.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. The user token is a response to the API for [obtaining a user token](#). This API is the only one that does not require authentication.
- **X-Project-ID**: specifies a subproject ID. This parameter is optional and can be used in multi-project scenarios.
- **X-Domain-ID**: specifies an account ID.

The API for [obtaining a user token](#) does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

Request Body

A request body is generally sent in structured format. It corresponds to **Content-Type** in the request header and transfers content except the request header. If the request body contains Chinese characters, set **Content-type** to **utf-8**, for example, Content-Type: application/json;utf-8.

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.

In the case of the API for **obtaining a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace the italic fields in bold with the actual values.

- ***username***: login user name
- ***********: login password of the user
- ***domainname***: account to which the user belongs
- ***xxxxxxxxxxxxxxxxxxxx***: project name, that is, the region name obtained from **Regions and Endpoints**.

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

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

NOTE

scope specifies the scope of the token. The value can be *project* or *domain*. In the preceding example, the value is **project**, indicating that the obtained token can access only resources in the specified project. If the value is **domain**, the obtained token takes effect for all resources of the specified account. For details about the scope parameter, see **Obtaining a User Token**.

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

3.2 Authentication

You can use either of the following authentication methods to call APIs:

- Token-based authentication: Requests are authenticated using a token.
- Authentication using AK/SK: Requests are encrypted using access key ID (AK)/secret access key (SK).

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the Identity and Access Management (IAM) API for obtaining a user token.

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

In [Making an API Request](#), the process of calling the API for [obtaining a user token](#) is described.

A project-level token is required for calling ServiceStage APIs. When calling the API for [obtaining a user token](#), set **project** in **auth.scope** in the request body, as shown in the following figure.

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

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

```
GET https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signature SDK, see [API Request Signing Guide](#).

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

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

Status Code

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](#).

For example, if status code 201 is returned for calling the API for [obtaining a user token](#), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

In the response to the API for [obtaining a user token](#), **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

Response Body

A response body is generally returned in structured format. It corresponds to **Content-Type** in the response header and transfers content except the response header.

The following is part of the response body of the API for [obtaining a user token](#).

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "xx-xxxxxx-1",
            .....

```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
  "error_code": "FGS.0111",
  "error_msg": "xxxxxxxx"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Application Management V2 APIs

4.1 Meta

4.1.1 Obtaining All Supported Runtime Systems of Application Components

Function

This API is used to obtain all supported runtime systems of application components.

URI

GET /v2/{project_id}/cas/metadata/runtimes

Table 4-1 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 4-2 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-3 Response parameters

Parameter	Type	Description
runtimes	Array of objects	Runtime system parameters. See Table 4-4 .

Table 4-4 runtimes parameters

Parameter	Type	Description
type_name	String	Type.
display_name	String	Display name.
container_default_port	Integer	Default container port.
type_desc	String	Type description.

Example Request

None

Example Response

```
{
  "runtimes": [
    {
      "type_name": "Java8",
      "display_name": "Java 8",
      "container_default_port": 8080,

```

```
    "type_desc": "Java 8 runtime server"  
  }  
]  
}
```

Status Code

Table 4-5 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.1.2 Obtaining All Supported Flavors of Application Resources

Function

This API is used to obtain all supported flavors of application resources.

URI

GET /v2/{project_id}/cas/metadata/flavors

Table 4-6 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 4-7 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-8 Response parameters

Parameter	Type	Description
flavors	Array of objects	Flavor parameters. See Table 4-9 .

Table 4-9 flavors parameters

Parameter	Type	Description
flavor_id	String	Flavor ID.
storage_size	String	Storage size.
num_cpu	String	CPU limit.
num_cpu_init	String	Initial CPU value.
memory_size	String	Memory limit.
memory_size_init	String	Initial memory value.
label	String	Label.
custom	boolean	Whether resource specifications are customized.

Example

None

Example Response

```
{
  "flavors": [
    {
      "flavor_id": "MICRO-5G:0.5C:1G",
      "storage_size": "5G",
      "num_cpu": "500m",
      "num_cpu_init": "200m",
      "memory_size": "1Gi",
      "memory_size_init": "200Mi",
      "label": "Micro: 5G Storage, 0.5 CPU, 1G Memory",
      "custom": false
    }
  ]
}
```

Status Code

Table 4-10 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2 Environment

4.2.1 Creating an Environment

Function

An environment is a collection of compute, storage, and network resources used for deploying and running an application. ServiceStage combines the compute resources (such as CCE clusters and ECSs), network resources (such as ELB instances and EIPs), and middleware (such as DCS instances, RDS instances, and CSE engines) in the same VPC into an environment, such as a development environment, testing environment, pre-production environment, or production environment. The resources within an environment can be networked together. Managing resources and deploying services by environment simplifies O&M.

This API is used to create an environment.

URI

POST /v2/{project_id}/cas/environments

Table 4-11 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 4-12 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-13 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Environment name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
alias	No	String	Environment alias. The value can contain up to 64 characters.
deploy_mode	Yes	String	Environment type. <ul style="list-style-type: none">virtualmachinecontainer: Kubernetes

Parameter	Mandatory	Type	Description
description	No	String	Environment description. The value can contain up to 128 characters.
enterprise_project_id	No	String	Enterprise project ID.
charge_mode	No	String	Billing mode. Value: provided, on_demanded, and monthly. provided (default) indicates that provided resources are used and no fees are charged, on_demanded indicates the pay-per-use billing mode, and monthly indicates the monthly billing mode.
vpc_id	Yes	String	VPC ID.
base_resources	Yes	Array of objects	Basic resources. See Table 4-14 .
optional_resources	No	Array of objects	Optional resources. See Table 4-14 .

Table 4-14 resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID.
type	Yes	String	Resource name.
name	No	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Response

Table 4-15 Response parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
alias	String	Environment alias.

Parameter	Type	Description
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
charge_mode	String	Billing mode.
deploy_mode	String	Environment type. <ul style="list-style-type: none">• virtualmachine• container: Kubernetes
vpc_id	String	VPC ID.
base_resources	Array of objects	Basic resources. See Table 4-16 . The returned value is empty.
optional_resources	Array of objects	Optional resources. See Table 4-16 . The returned value is empty.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-16 resources parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Example Request

Create a VM environment **development-env**. The ID of the VPC where the environment is located is **29d55020-ae0e-4a18-871c-93e6976ee7bd**.

```
{
  "name": "development-env",
  "description": "",
  "charge_mode": "provided",
  "deploy_mode": "container",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "b6862a62-d916-11e9-bdf1-0255ac101fd9",
```

```

        "type": "cce"
      }
    ],
    "optional_resources": [
      {
        "id": "default",
        "type": "cse"
      }
    ]
  }

```

Example Response

```

{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env",
  "alias": null,
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "deploy_mode": "container",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "b6862a62-d916-11e9-bdf1-0255ac101fd9",
      "type": "cce"
    }
  ],
  "optional_resources": [
    {
      "id": "default",
      "type": "cse"
    }
  ],
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610418873730
}

```

Status Code

Table 4-17 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2.2 Obtaining All Environments

Function

This API is used to obtain all created environments.

URI

GET /v2/{project_id}/cas/environments

Table 4-18 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Table 4-19 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of records to be displayed. The value is 1000 or ranges from 0 to 100. If the specified value is not within the range, value 10 is assigned. In the non-pagination scenario, the value is 1000 . In the pagination scenario, the value ranges from 0 to 100.
offset	No	Integer	Offset, which indicates the result after the Nth data record is queried.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time. If the transferred value is not within the supported enumerated value range, the default sorting field is used.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 4-20 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-21 Response parameters

Parameter	Type	Description
count	Integer	Total number of environments.
environments	Array of objects	Environment parameters. See Table 4-22 .

Table 4-22 environments parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
alias	String	Environment alias.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
charge_mode	String	Billing mode.
deploy_mode	String	Environment type. <ul style="list-style-type: none">virtualmachinecontainer: Kubernetes

Parameter	Type	Description
vpc_id	String	VPC ID.
base_resources	Array of objects	Basic resources. See Table 4-23 .
optional_resources	Array of objects	Optional resources. See Table 4-23 .
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-23 resources parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Example Request

None

Example Response

```
{
  "environments": [
    {
      "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
      "name": "development-env",
      "alias": null,
      "description": "",
      "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
      "enterprise_project_id": "0",
      "charge_mode": "provided",
      "deploy_mode": "container",
      "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
      "base_resources": [
        {
          "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
          "type": "cce",
          "name": "cce-test"
        }
      ],
      "optional_resources": [
        {
          "id": "9963306a-791e-468c-9306-cf80a9d00298",
          "type": "elb",

```

```
        "name": "elb-2dd2"
      }
    ],
    "creator": "ss-test",
    "create_time": 1610418873730,
    "update_time": 1610418873730
  }
],
"count": 1
}
```

Status Code

Table 4-24 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2.3 Modifying Environment Information

Function

This API is used to modify environment information based on the environment ID.

URI

PUT /v2/{project_id}/cas/environments/{environment_id}

Table 4-25 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 4-26 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-27 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Environment name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
alias	No	String	Environment alias. The value can contain up to 64 characters.
description	No	String	Environment description. The value can contain up to 128 characters.

Response

Table 4-28 Response parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
alias	String	Environment alias.
description	String	Environment description.

Parameter	Type	Description
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
charge_mode	String	Billing mode.
deploy_mode	String	Environment type. <ul style="list-style-type: none">virtualmachinecontainer: Kubernetes
vpc_id	String	VPC ID.
base_resources	Array of objects	Basic resources. See Table 4-29 .
optional_resources	Array of objects	Optional resources. See Table 4-29 .
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-29 resources parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Example Request

Add environment description for environment **development-env2**.

```
{
  "name": "development-env2",
  "description": "here is description"
}
```

Example Response

```
{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env2",
  "alias": null,
  "description": "here is description",
}
```

```
"project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
"enterprise_project_id": "0",
"charge_mode": "provided",
  "deploy_mode": "container",
"vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
"base_resources": [
  {
    "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
    "type": "cce",
    "name": "cce-test"
  }
],
"optional_resources": [
  {
    "id": "9963306a-791e-468c-9306-cf80a9d00298",
    "type": "elb",
    "name": "elb-2dd2"
  }
],
"creator": "ss-test",
"create_time": 1610418873730,
"update_time": 1610420992462
}
```

Status Code

Table 4-30 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2.4 Deleting an Environment Based on the Environment ID

Function

This API is used to delete an environment based on the environment ID.

URI

DELETE /v2/{project_id}/cas/environments/{environment_id}

Table 4-31 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 4-32 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 4-33 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request

HTTP Status Code	Description
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2.5 Obtaining Environment Details Based on the Environment ID

Function

This API is used to obtain environment details based on the environment ID.

URI

GET /v2/{project_id}/cas/environments/{environment_id}

Table 4-34 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 4-35 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-36 Response parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
alias	String	Environment alias.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
charge_mode	String	Billing mode.
deploy_mode	String	Environment type. <ul style="list-style-type: none">virtualmachinecontainer: Kubernetes
vpc_id	String	VPC ID.
base_resources	Array of objects	Basic resources. See Table 4-37 .
optional_resources	Array of objects	Optional resources. See Table 4-37 .
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-37 resources parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Example Request

None

Example Response

```
{
  "id": "ea011e01-2eb5-453f-87bf-874e4a855abe",
  "name": "dev-env",
  "alias": null,
  "description": "develop environment",
  "project_id": "bf8523d898b64e4eb956e3be3555ca16",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "deploy_mode": "container",
  "vpc_id": "234241234124xxvasf2342xxxxxxxxxx",
  "base_resources": [
    {
      "id": "211112333sd332w23322332",
      "type": "cce"
    }
  ],
  "optional_resources": null,
  "creator": "string",
  "create_time": 1578984198394,
  "update_time": 1578984198394
}
```

Status Code

Table 4-38 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.2.6 Modifying Environment Resources

Function

This API is used to modify environment resources.

URI

PATCH /v2/{project_id}/cas/environments/{environment_id}/resources

Table 4-39 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 4-40 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-41 Request body parameters

Parameter	Mandatory	Type	Description
add_base_resources	No	Array of objects	Basic resources to be added. See Table 4-42 .
add_optional_resources	No	Array of objects	Optional resources to be added. See Table 4-43 .
remove_resources	No	Array of objects	Resources to be removed. See Table 4-44 .

Table 4-42 add_base_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID. (To ensure that the API can be called properly, enter the correct resource ID.)

Parameter	Mandatory	Type	Description
name	No	String	Resource name.
type	Yes	String	Basic resources: cce, as, and ecs.

Table 4-43 add_optional_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID. (To ensure that the API can be called properly, enter the correct resource ID.)
name	No	String	Resource name.
type	Yes	String	Optional resources: rds, dcs, elb, and other services.

Table 4-44 remove_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID. (To ensure that the API can be called properly, enter the correct resource ID.)
name	No	String	Resource name.
type	Yes	String	Basic resources: cce, as, and ecs. When deploy_mode is set to virtualmachine , you can add ecs and as. When deploy_mode is set to container , you can add cce. Optional resources: rds, dcs, elb, and other services.

Response

Table 4-45 Response parameters

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

Parameter	Type	Description
alias	String	Environment alias.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
charge_mode	String	Billing mode.
deploy_mode	String	Environment type. <ul style="list-style-type: none">• virtualmachine• container: Kubernetes
vpc_id	String	VPC ID.
base_resources	Array of objects	Basic resources. See Table 4-46 .
optional_resources	Array of objects	Optional resources. See Table 4-46 .
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-46 schemas parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Example Request

Add ECS basic resource **ecs-9876** and CSE optional resource **Cloud Service Engine**, and remove ECS resource **test**.

```
{
  "add_base_resources": [
    {
      "id": "ed2f3420-7031-4d93-b92b-e360cd4acf9e",
      "type": "ecs",
      "name": "ecs-9876"
    }
  ]
}
```

```

    ],
    "add_optional_resources": [
      {
        "id": "default",
        "type": "cse",
        "name": "Cloud Service Engine"
      }
    ],
    "remove_resources": [
      {
        "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
        "name": "test",
        "type": "ecs"
      }
    ]
  ]
}

```

Example Response

```

{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env2",
  "alias": null,
  "description": "here is description",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "deploy_mode": "virtualmachine",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "ed2f3420-7031-4d93-b92b-e360cd4acf9e",
      "type": "ecs",
      "name": "ecs-9876"
    }
  ],
  "optional_resources": [
    {
      "id": "9963306a-791e-468c-9306-cf80a9d00298",
      "type": "elb",
      "name": "elb-2dd2"
    },
    {
      "id": "default",
      "type": "cse",
      "name": "Cloud Service Engine"
    }
  ],
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610420992462
}

```

Status Code

Table 4-47 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found

HTTP Status Code	Description
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3 Application

4.3.1 Creating an Application

Function

An application is a service system with complete functions and consists of one or more components related to features.

This API is used to create an application.

URI

POST /v2/{project_id}/cas/applications

Table 4-48 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 4-49 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-50 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Application name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.
description	No	String	Application description. The value can contain up to 128 characters.
enterprise_project_id	No	String	Enterprise project ID.

Response

Table 4-51 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Parameter	Type	Description
unified_model	String	Whether to enable the unified model.

Example Request

Create application **app-xpmtii** and add application description **test**.

```
{
  "name": "app-xpmtii",
  "description": "test"
}
```

Example Response

```
{
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",
  "name": "app-xpmtii",
  "description": "test",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "creator": "ss-test",
  "create_time": 1610432385245,
  "update_time": 1610432385245,
  "unified_model": null
}
```

Status Code

Table 4-52 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.2 Obtaining All Applications

Function

This API is used to obtain all created applications.

URI

GET /v2/{project_id}/cas/applications

Table 4-53 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. See Obtaining a Project ID .

Table 4-54 Query parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of records to be displayed. The value is 1000 or ranges from 0 to 100. If the specified value is not within the range, value 10 is assigned. In the non-pagination scenario, the value is 1000 . In the pagination scenario, the value ranges from 0 to 100.
offset	No	String	Offset, which indicates the result after the Nth data record is queried.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time. If the transferred value is not within the supported enumerated value range, the default sorting field is used.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 4-55 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-56 Response parameters

Parameter	Type	Description
count	Integer	Total number of applications.
applications	Array of objects	Application information. See Table 4-57 .

Table 4-57 applications parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
unified_model	String	Whether to enable the unified model.
component_count	Integer	Number of components.

Example Request

None

Example Response

```
{
  "count": 1,
  "applications": [
    {
      "id": "e5213b62-0e3c-476f-9960-3e4108787350",
      "name": "app-xpmtii",
      "description": "test",
      "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
      "enterprise_project_id": "0",
      "creator": "ss-test",
      "create_time": 1610432385245,
      "update_time": 1610432385245,
      "unified_model": null,
      "component_count": 0
    }
  ]
}
```

Status Code

Table 4-58 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.3 Modifying Application Information

Function

This API is used to modify application information based on the application ID.

URI

PUT /v2/{project_id}/cas/applications/{application_id}

Table 4-59 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 4-60 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-61 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Application name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.
description	No	String	Application description. The value can contain up to 128 characters.

Response

Table 4-62 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
unified_model	String	Whether to enable the unified model.

Example Request

Modify application name to **app-test**.

```
{
  "name": "app-test",
  "description": "test"
}
```

Example Response

```
{
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",
  "name": "app-test",
  "description": "test",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "creator": "ss-test",
  "create_time": 1610432385245,
  "update_time": 1610433070875,
  "unified_model": null
}
```

Status Code

Table 4-63 Status codes

HTTP Status Code	Description
200	OK

HTTP Status Code	Description
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.4 Deleting an Application Based on the Application ID

Function

This API is used to delete an application based on the application ID.

URI

DELETE /v2/{project_id}/cas/applications/{application_id}

Table 4-64 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 4-65 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 4-66 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found
409	Conflict
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.5 Obtaining Application Details Based on the Application ID

Function

This API is used to obtain application details based on the application ID.

URI

GET /v2/{project_id}/cas/applications/{application_id}

Table 4-67 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 4-68 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-69 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.

Parameter	Type	Description
create_time	Integer	Creation time.
update_time	Integer	Update time.
unified_model	String	Whether to enable the unified model.
component_count	Integer	Number of components.

Example Request

None

Example Response

```
{
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",
  "name": "app-test",
  "description": "test",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "creator": "ss-test",
  "create_time": 1610432385245,
  "update_time": 1610433070875,
  "unified_model": null,
  "component_count": 0
}
```

Status Code

Table 4-70 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.6 Modifying Application Configurations

Function

This API is used to modify application configurations.

Application configurations refer to common environment variables of an application. Components deployed in the application inherit these environment variables.

URI

PUT /v2/{project_id}/cas/applications/{application_id}/configuration

Table 4-71 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 4-72 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-73 Request body parameters

Parameter	Mandatory	Type	Description
environment_id	Yes	String	Environment ID, which can be obtained in the corresponding environment by referring to Obtaining All Environments .
configuration	Yes	Object	Application configurations, such as public environment variables. See Table 4-74 .

Table 4-74 configuration parameter

Parameter	Mandatory	Type	Description
env	Yes	Array of objects	Environment variables of the application. See Table 4-75 . If the names of multiple environment variables are the same, only the last environment variable takes effect.

Table 4-75 env parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the environment variable. The value contains 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and dots (.), and cannot start with a digit.
value	Yes	String	Value of the environment variable.

NOTICE

The **env** variable in the request parameter **configuration** will overwrite the existing environment variable configuration of the application. Set this parameter based on the scenario where environment variables are added or replaced.

Response

Table 4-76 Response parameters

Parameter	Type	Description
application_id	String	Application ID.
environment_id	String	Environment ID.
configuration	Object	Application configurations. See Table 4-77 .

Table 4-77 configuration parameter

Parameter	Type	Description
env	Array of objects	Environment variables of the application. See Table 4-78 .

Table 4-78 env parameters

Parameter	Type	Description
name	String	Name of the environment variable.
value	String	Value of the environment variable.

Example Request

Modify the environment whose ID is **0fa75dfe-0d32-4bc1-848b-8008cf3f2567** by changing the name to **app-env** and the value to **env**.

```
{
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {
        "name": "app-env",
        "value": "env"
      }
    ]
  }
}
```

Example Response

```
{
  "application_id": "06cf2fda-af45-44b4-9e22-6294c4804515",
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {
        "name": "app-env",
```

```
    "value": "env"  
  }  
  ]  
}  
}
```

Status Code

Table 4-79 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.7 Deleting Application Configurations

Function

This API is used to delete application configurations.

URI

DELETE /v2/{project_id}/cas/applications/{application_id}/configuration

Table 4-80 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 4-81 Query parameters

Parameter	Mandatory	Type	Description
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 4-82 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 4-83 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found

HTTP Status Code	Description
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.3.8 Obtaining Application Configurations

Function

This API is used to obtain application configurations.

URI

GET /v2/{project_id}/cas/applications/{application_id}/configuration

Table 4-84 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 4-85 Query parameters

Parameter	Mandatory	Type	Description
environment_id	No	String	Environment ID. If this parameter is not specified, all environments are queried. See Obtaining All Environments .

Request

Table 4-86 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-87 Response parameters

Parameter	Type	Description
configuration	Array of objects	Configuration information. See Table 4-88 .

Table 4-88 Application configuration parameters

Parameter	Type	Description
application_id	String	Application ID.
environment_id	String	Environment ID.
configuration	Object	Application configurations. See Table 4-89 .

Table 4-89 configuration parameters

Parameter	Type	Description
env	Array of objects	Environment variables of the application component. See Table 4-90 .

Table 4-90 env parameters

Parameter	Type	Description
name	String	Name of the environment variable.
value	String	Value of the environment variable.

Example Request

None

Example Response

```
{
  "configuration": [
    {
      "application_id": "e5213b62-0e3c-476f-9960-3e4108787350",
      "environment_id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
      "configuration": {
        "env": [
          {
            "name": "env",
            "value": "value1"
          }
        ]
      }
    }
  ]
}
```

Status Code

Table 4-91 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.4 Component

4.4.1 Creating an Application Component

Function

An application component implements a service feature of an application. It is in the form of code or software packages and can be deployed independently.

This API is used to create an application component.

URI

POST /v2/{project_id}/cas/applications/{application_id}/components

Table 4-92 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 4-93 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-94 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Application component name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
runtime	Yes	String	Runtime. The value can be obtained from type_name returned by the API in Obtaining All Supported Runtime Systems of Application Components .
category	Yes	String	Application component category. Example: Webapp, MicroService, or Common.
sub_category	No	String	Application component sub-category. Webapp sub-categories include Web. MicroService sub-categories include Java Chassis, and Spring Cloud. Common sub-category can be empty.
description	No	String	Description. The value can contain up to 128 characters.
source	No	Object	Source of the code or software package. See Table 4-95 .
build	No	Object	Component build. See Table 4-99 .

Table 4-95 source parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	Type. Option: source code or artifact software package.
spec	Yes	Object	For details about the source code, see Table 4-96 . For details about the artifact software package, see Table 4-97 .

Table 4-96 code spec parameters

Parameter	Mandatory	Type	Description
repo_type	No	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
repo_url	No	String	Code repository URL. Example: https://github.com/example/demo.git.
repo_ref	No	String	Code branch or tag. Default value: master.
repo_auth	No	String	Authorization name, which can be obtained from the authorization list.

Table 4-97 artifact spec parameters

Parameter	Mandatory	Type	Description
storage	No	String	Storage mode. Value: swr or obs.
type	No	String	Type. Value: package.
url	No	String	Address of the software package or source code.
auth	No	String	Authentication mode. Value: iam or none. Default value: iam.
properties	No	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-98 .
webUrl	No	String	Address of a software repository.

Table 4-98 artifact spec properties parameters

Parameter	Mandatory	Type	Description
endpoint	No	String	OBS endpoint address. Example: https://obs.region_id.external_domain_name.com.
bucket	No	String	Name of the OBS bucket where the software package is stored.
key	No	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-99 build parameter

Parameter	Mandatory	Type	Description
parameters	No	Map<String, Object>	See Table 4-100 . This parameter is provided only when no ID is available during build creation.

Table 4-100 parameters

Parameter	Mandatory	Type	Description
build_cmd	No	String	Compilation command. By default: 1. When build.sh exists in the root directory, the command is ./build.sh . 2. When build.sh does not exist in the root directory, the command varies depending on the OS. Example: <ul style="list-style-type: none">• Java and Tomcat: mvn clean package• Node.js: npm build
dockerfile_path	No	String	Address of the Docker file. By default, the Docker file is in the root directory (.).
artifact_namespace	No	String	Build archive organization. Default value: <code>cas_{project_id}</code> .
cluster_id	No	String	ID of the cluster to be built.
node_label_selector	No	Map<String, String>	key : key of the tag. value : value of the tag.

Response

Table 4-101 Response parameters

Parameter	Type	Description
id	String	Application component ID.
name	String	Application component name.
status	Integer	Value: 0 or 1. 0: Normal. 1: Being deleted.

Parameter	Type	Description
runtime	String	Runtime system.
category	String	Application component category. Example: Webapp, MicroService, or Common.
sub_category	String	Application component sub-category. Webapp sub-categories include Web. MicroService sub-categories include Java Chassis, and Spring Cloud. Common sub-category can be empty.
description	String	Description.
project_id	String	Project ID.
application_id	String	Application ID.
source	Object	Source of the code or software package. See Table 4-102 .
build	Object	Build. See Table 4-106 .
create_time	Integer	Creation time.
update_time	Integer	Update time.
creator	String	Creator.

Table 4-102 source parameters

Parameter	Type	Description
kind	String	Type. Option: source code or artifact software package.
spec	Object	For details about the artifact software package, see Table 4-104 ; For details about the source code, see Table 4-103 .

Table 4-103 code spec parameters

Parameter	Type	Description
repo_type	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git .
repo_ref	String	Code branch or tag. Default value: master.

Parameter	Type	Description
repo_auth	String	Authorization name, which can be obtained from the authorization list.

Table 4-104 artifact spec parameters

Parameter	Type	Description
storage	String	Storage mode.
type	String	Type.
url	String	Address of the software package or source code.
auth	String	Authentication mode.
properties	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-105 .

Table 4-105 artifact spec properties parameters

Parameter	Type	Description
endpoint	String	OBS endpoint address. Example: https://obs.region_id.external_domain_name.com.
bucket	String	Name of the OBS bucket where the software package is stored.
key	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-106 build parameter

Parameter	Type	Description
id	String	Type.
parameters	Map<String, Object>	See Table 4-107 .

Table 4-107 parameters

Parameter	Type	Description
build_cmd	String	Compilation command.
dockerfile_path	String	Address of the Docker file.
artifact_namespace	String	Build archive organization.
cluster_id	String	ID of the cluster to be built.
node_selector	Map<String, String>	key: key of the tag. value: value of the tag.

Example Request

Create an application microservice component for software package deployment. Set runtime to Java8 and the component software package source to the **demo.jar** package in the myapp bucket of OBS.

```
{
  "name": "mycomponent",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "build": {
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": false,
      "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "cluster_name": "cce-test",
      "cluster_type": "VirtualMachine"
    }
  },
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://myapp/demo.jar",
      "properties": {
        "bucket": "myapp",
        "key": "demo.jar",
        "endpoint": "https://obs.region_id.external_domain_name.com"
      }
    }
  }
}
```

Example Response

```
{
  "id": "384eb8d4-c193-4d84-9558-6fda2366b536",
  "name": "mycomponent",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "project_id": "384eb8d4-c193-4d84-9558-6fda23698536",
  "application_id": "a8f7eed5-0aa0-4251-9723-c9119a6bf56d",
}
```



```
"source": {
  "kind": "artifact",
  "spec": {
    "storage": "obs",
    "type": "package",
    "url": "obs://myapp/demo.jar",
    "properties": {
      "bucket": "myapp",
      "key": "demo.jar",
      "endpoint": "https://obs.region_id.external_domain_name.com"
    }
  }
},
"build": {
  "id": "w3dpv7p0t1vpxvey5hjb22iuwxway1vupwx0nae1",
  "parameters": {
    "artifact_namespace": "ns",
    "dockerfile_path": "./"
  }
},
"status": 0,
"creator": "xxx",
"create_time": 1610333934288,
"update_time": 1610333934288
}
```

Status Code

Table 4-108 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.4.2 Obtaining All Components of an Application

Function

This API is used to obtain all components of an application.

URI

GET /v2/{project_id}/cas/applications/{application_id}/components

Table 4-109 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 4-110 Query parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of records to be displayed. The value is 1000 or ranges from 0 to 100. If the specified value is not within the range, value 10 is assigned. In the non-pagination scenario, the value is 1000 . In the pagination scenario, the value ranges from 0 to 100.
offset	No	String	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 4-111 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-112 Response parameters

Parameter	Type	Description
count	Integer	Total number of components.
components	Array of objects	Component parameters. See Table 4-113 .

Table 4-113 components parameters

Parameter	Type	Description
id	String	Application component ID.
name	String	Application component name.
status	Integer	Value: 0 or 1. 0: Normal. 1: Being deleted.
runtime	String	Runtime system.
category	String	Application component category. Example: Webapp, MicroService, or Common.
sub_category	String	Application component sub-category. Webapp sub-categories include Web. MicroService sub-categories include Java Chassis, and Spring Cloud. Common sub-category can be empty.
application_id	String	Application ID.
source	Object	Source of the code/software package. See Table 4-114 .
description	String	Description.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
project_id	String	Tenant's project ID.
build	Object	Build. See Table 4-118 .

Table 4-114 source parameters

Parameter	Type	Description
kind	String	Type. Option: source code or artifact software package.
spec	Object	For details about the source code, see Table 4-115 . For details about the artifact software package, see Table 4-116 .

Table 4-115 code spec parameters

Parameter	Type	Description
repo_type	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.
repo_ref	String	Code branch or tag. Default value: master.
repo_auth	String	Authorization name, which can be obtained from the authorization list.

Table 4-116 artifact spec parameters

Parameter	Type	Description
storage	String	Storage mode.
type	String	Type.
url	String	Address of the software package/source code.
auth	String	Authentication mode.
properties	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-117 .

Table 4-117 artifact spec properties parameters

Parameter	Type	Description
endpoint	String	OBS endpoint address. Example: https://obs.region_name.external_domain_name.com.
bucket	String	Name of the OBS bucket where the software package is stored.

Parameter	Type	Description
key	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-118 build parameters

Parameter	Type	Description
id	String	Job ID.
parameters	Map<String,String>	See Table 4-119 .

Table 4-119 parameters description

Parameter	Type	Description
build_cmd	String	Compilation command.
dockerfile_path	String	Address of the Docker file.
artifact_namespace	String	Build archive organization.
cluster_id	String	ID of the cluster to be built.
node_label_selector	Map<String, String>	key indicates the key of the tag, and value indicates the value of the tag.

Example Request

None

Example Response

```
{
  "components": [
    {
      "id": "384eb8d4-c193-4d84-9558-6fda2366b536",
      "name": "mycomponent",
      "runtime": "Java8",
      "category": "MicroService",
      "sub_category": "Java Chassis",
      "description": "",
      "project_id": "384eb8d4-c193-4d84-9558-6fda23698536",
      "application_id": "a8f7eed5-0aa0-4251-9723-c9119a6bf56d",
      "source": {
```

```
{
  "kind": "artifact",
  "spec": {
    "storage": "obs",
    "type": "package",
    "url": "obs://myapp/demo.jar",
    "properties": {
      "bucket": "myapp",
      "key": "demo.jar",
      "endpoint": "https://obs.region_id.external_domain_name.com"
    }
  },
  "build": {
    "id": "w3dpv7p0t1vpxvey5hjb22iuwxway1vupwx0nae1",
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": true
    }
  },
  "status": 0,
  "creator": "xxx",
  "create_time": 1610333934288,
  "update_time": 1610333934288
},
"count": 1
}
```

Status Code

Table 4-120 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.4.3 Modifying Component Information Based on the Component ID

Function

This API is used to modify component information based on the component ID.

URI

PUT /v2/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 4-121 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 4-122 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-123 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Application component name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
description	No	String	Description. The value can contain up to 128 characters.

Parameter	Mandatory	Type	Description
source	No	Object	Source of the code/software package. See Table 4-124 .
build	No	Object	Build. See Table 4-128 .

Table 4-124 source parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	Type. Option: source code or artifact software package.
spec	Yes	Object	For details about the source code, see Table 4-125 . For details about the artifact software package, see Table 4-126 .

Table 4-125 code spec parameters

Parameter	Mandatory	Type	Description
repo_type	No	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
repo_url	No	String	Code repository URL. Example: https://github.com/example/demo.git .
repo_ref	No	String	Code branch or tag. Default value: master.
repo_auth	No	String	Authorization name, which can be obtained from the authorization list.

Table 4-126 artifact spec parameters

Parameter	Mandatory	Type	Description
storage	No	String	Storage mode. Value: swr or obs.
type	No	String	Type. Value: package.
url	No	String	Address of the software package/source code.

Parameter	Mandatory	Type	Description
auth	No	String	Authentication mode. Value: iam or none. Default value: iam.
properties	No	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-127 .
webUrl	No	String	Address of a software repository.

Table 4-127 artifact spec properties parameters

Parameter	Mandatory	Type	Description
endpoint	No	String	OBS endpoint address. Example: https://obs.region_id.external_domain_name.com.
bucket	No	String	Name of the OBS bucket where the software package is stored.
key	No	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-128 build parameters

Parameter	Mandatory	Type	Description
parameters	No	Map<String, Object>	See Table 4-129 . This parameter is provided only when no ID is available during build creation.

Table 4-129 parameters description

Parameter	Mandatory	Type	Description
build_cmd	No	String	Compilation command. By default: <ol style="list-style-type: none"> When build.sh exists in the root directory, the command is ./build.sh. When build.sh does not exist in the root directory, the command varies depending on the OS. Example: <ul style="list-style-type: none"> Java and Tomcat: mvn clean package Node.js: npm build
dockerfile_path	No	String	Address of the Docker file. By default, the Docker file is in the root directory (.).
artifact_namespace	No	String	Build archive organization. Default value: cas_{project_id}.
cluster_id	Yes	String	ID of the cluster to be built.
node_label_selector	No	Map<String, String>	key indicates the key of the tag, and value indicates the value of the tag.
cluster_name	No	String	Name of the cluster to be built.
use_public_cluster	No	Boolean	Whether to use the public cluster: <ul style="list-style-type: none"> true false
cluster_type	No	String	Type of the cluster to be built.

Response

Table 4-130 Response parameters

Parameter	Type	Description
id	String	Application component ID.
name	String	Application component name.
status	Integer	Value: 0 or 1. 0: Normal. 1: Being deleted.
runtime	String	Runtime system.

Parameter	Type	Description
category	String	Application component category. Example: Webapp, MicroService, or Common.
sub_category	String	Application component sub-category. Webapp sub-categories include Web. MicroService sub-categories include Java Chassis, and Spring Cloud. Common sub-category can be empty.
description	String	Description.
project_id	String	Project ID.
application_id	String	Application ID.
source	Object	Source of the code/software package. See Table 4-131 .
build	Object	Build. See Table 4-135 .
create_time	Integer	Creation time.
update_time	Integer	Update time.
creator	String	Creator.

Table 4-131 source parameters

Parameter	Type	Description
kind	String	Type.
spec	Object	For details about the source code, see Table 4-132 . For details about the artifact software package, see Table 4-133 .

Table 4-132 code spec parameters

Parameter	Type	Description
repo_type	String	Code repository type. Value: github, bitbucket, or gitlab.
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.
repo_ref	String	Code branch or tag. Default value: master.
repo_auth	String	Authorization name, which can be obtained from the authorization list.

Table 4-133 spec parameters

Parameter	Type	Description
storage	String	Storage mode.
type	String	Type.
url	String	Address of the software package/source code.
auth	String	Authentication mode.
properties	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-134 .

Table 4-134 artifact spec properties parameters

Parameter	Type	Description
endpoint	String	OBS endpoint address. Example: https://obs.region_id.external_domain_name.com.
bucket	String	Name of the OBS bucket where the software package is stored.
key	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-135 build parameters

Parameter	Type	Description
id	String	Type.
parameters	Map<String, Object>	See Table 4-136 .

Table 4-136 parameters description

Parameter	Type	Description
build_cmd	String	Compilation command.
dockerfile_path	String	Address of the Docker file.
artifact_name_space	String	Build archive organization.

Parameter	Type	Description
cluster_id	String	ID of the cluster to be built.
node_label_selector	Map<String, String>	key indicates the key of the tag, and value indicates the value of the tag.

Example Request

Modify the component information corresponding to the specified component ID as follows: Change the component name to **test-micro**, set the component software package source to the **fusionweather-1.0.0.jar** package in the test-soft bucket of OBS, and specify the cce-test cluster as the component to build a cluster.

```
{
  "name": "test-micro",
  "description": "",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://test-soft/fusionweather-1.0.0.jar",
      "auth": "iam",
      "webUrl": "",
      "properties": {
        "bucket": "test-soft",
        "endpoint": "https://obs.region_id.development.com:443",
        "key": "fusionweather-1.0.0.jar"
      }
    }
  },
  "build": {
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": false,
      "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "cluster_name": "cce-test",
      "cluster_type": "VirtualMachine"
    }
  },
}
```

Example Response

```
{
  "id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
  "name": "test-micro",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://test-soft/fusionweather-1.0.0.jar",
      "auth": "iam",
      "properties": {
        "bucket": "test-soft",

```

```
        "endpoint": "https://obs.region_id.development.com:443",
        "key": "fusionweather-1.0.0.jar"
      }
    },
    "build": {
      "id": "zpdkj42b3rnko8bmyojzt1ng828ubychw13j8flv",
      "parameters": {
        "artifact_namespace": "ns",
        "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2"
      }
    },
    "status": 0,

    "creator": "ss-test",
    "create_time": 1610331760105,
    "update_time": 1610519881943
  }
}
```

Status Code

Table 4-137 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.4.4 Deleting a Component Based on the Component ID

Function

This API is used to delete a component based on the component ID.

URI

DELETE /v2/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 4-138 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Table 4-139 Query parameters

Parameter	Mandatory	Type	Description
force	No	boolean	Whether to forcibly delete the component.

Request

Table 4-140 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 4-141 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found
409	Conflict
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.4.5 Obtaining Component Information Based on the Component ID

Function

This API is used to obtain component information based on the component ID.

URI

GET /v2/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 4-142 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 4-143 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-144 Response parameters

Parameter	Type	Description
id	String	Application component ID.
name	String	Application component name.
status	Integer	Value: 0 or 1. 0: Normal. 1: Being deleted.
runtime	String	Runtime system.
category	String	Application component category. Example: Webapp, MicroService, or Common.
sub_category	String	Application component sub-category. Webapp sub-categories include Web. MicroService sub-categories include Java Chassis, and Spring Cloud. Common sub-category can be empty.
description	String	Description.
project_id	String	Project ID.
application_id	String	Application ID.
source	Object	Source of the code/software package. See Table 4-145 .
build	Object	Build. See Table 4-149 .
create_time	Integer	Creation time.

Parameter	Type	Description
update_time	Integer	Update time.
creator	String	Creator.
pipeline_ids	Array of String	Pipeline ID.

Table 4-145 source parameters

Parameter	Type	Description
kind	String	Type.
spec	Object	For details about the source code, see Table 4-146 . See Table 4-147 .

Table 4-146 code spec parameters

Parameter	Type	Description
repo_type	String	Code repository type. Value: github, bitbucket, or gitlab.
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.
repo_ref	String	Code branch or tag. Default value: master.
repo_auth	String	Authorization name, which can be obtained from the authorization list.

Table 4-147 spec parameters

Parameter	Type	Description
storage	String	Storage mode.
type	String	Type.
url	String	Address of the software package/source code.
auth	String	Authentication mode.
webUrl	String	Address of the software/source code repository.
properties	Object	Other attributes of the software package. You need to add these attributes only when you set storage to obs . See Table 4-148 .

Table 4-148 artifact spec properties parameters

Parameter	Type	Description
endpoint	String	OBS endpoint address. Example: https://obs.region_id.external_domain_name.com.
bucket	String	Name of the OBS bucket where the software package is stored.
key	String	Object in the OBS bucket, which is usually the name of the software package. If there is a folder, the path of the folder must be added. Example: test.jar or demo/test.jar.

Table 4-149 build parameters

Parameter	Type	Description
id	String	Job ID.
parameters	Map<String,String>	See Table 4-150 .

Table 4-150 parameters description

Parameter	Type	Description
build_cmd	String	Compilation command.
dockerfile_path	String	Address of the Docker file.
artifact_namepace	String	Build archive organization.
cluster_id	String	ID of the cluster to be built.
node_label_selector	Map<String,String>	key indicates the key of the tag, and value indicates the value of the tag.

Example Request

None

Example Response

```
{
  "id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
  "name": "test-micro",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": ""
```

```
"project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
"application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
"source": {
  "kind": "artifact",
  "spec": {
    "storage": "obs",
    "type": "package",
    "url": "obs://test-soft/fusionweather-1.0.0.jar",
    "auth": "iam",
    "webUrl": "",
    "properties": {
      "bucket": "test-soft",
      "endpoint": "https://obs.region_id.development.com:443",
      "key": "fusionweather-1.0.0.jar"
    }
  }
},
"build": {
  "id": "zpdkj42b3rnko8bmyojzt1ng828ubychw13j8flv",
  "parameters": {
    "artifact_namespace": "ns",
    "use_public_cluster": false,
    "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
    "cluster_name": "cce-test",
    "cluster_type": "VirtualMachine"
  }
},
"status": 0,
"creator": "ss-test",
"create_time": 1610331760105,
"update_time": 1610519881943
}
```

Status Code

Table 4-151 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5 Instance

4.5.1 Creating an Application Component Instance

Function

This API is used to create a component instance.

URI

POST /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances

Table 4-152 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 4-153 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-154 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Component instance name. The value can contain 2 to 63 characters, including lowercase letters, digits, and hyphens (-). It must start with a lowercase letter and end with a lowercase letter or digit.
environment_id	Yes	String	Environment ID.
replica	Yes	Integer	Number of instance replicas.
flavor_id	Yes	String	Resource specifications, which can be obtained by using the API in Obtaining All Supported Flavors of Application Resources . If you need to customize resource specifications, the format is CUSTOM-xxG:xxm-xxm:xxGi-xxGi. Where: <ul style="list-style-type: none">xxG: storage capacity allocated to a component instance. It is a reserved field. You can set it to a fixed number.xxm-xxm: the maximum and minimum number of CPU cores allocated to a component instance.xxGi-xxGi: the maximum and minimum memory allocated to a component instance. For example, CUSTOM-10G:0.5m-0.25m:1.6Gi-0.8Gi indicates that the maximum number of CPU cores allocated to a component instance is 0.5, the minimum number of CPU cores is 0.25, the maximum memory is 1.6 Gi, and the minimum memory is 0.8 Gi.

Parameter	Mandatory	Type	Description
artifacts	Yes	Map<String, Object>	Artifact. key indicates the component name. In the Docker container scenario, key indicates the container name. See Table 4-172 . If the source parameters of a component specify the software package source, this parameter is optional, and the software package source of the component is inherited by default. Otherwise, this parameter is mandatory.
version	Yes	String	Application component version that meets version semantics. Example: 1.0.0.
configuration	No	Object	Configuration parameters, such as environment variables, deployment configurations, and O&M monitoring. By default, this parameter is left blank. See Table 4-155 .
description	No	String	Description. The value can contain up to 128 characters.
external_accesses	No	Array of objects	External network access. See Table 4-173 .
refer_resources	Yes	Array of objects	Deployed resources. See Table 4-174 .

Table 4-155 configuration parameters

Parameter	Mandatory	Type	Description
env	No	Array of objects	Environment variable. See Table 4-156 .
storage	No	Array of objects	Data storage configuration. See Table 4-157 .
strategy	No	Object	Upgrade strategy. See Table 4-158 .
lifecycle	No	Object	Lifecycle. See Table 4-159 .
scheduler	No	Object	Scheduling strategy. See Table 4-160 .
probes	No	Object	Health check. See Table 4-161 .

Parameter	Mandatory	Type	Description
container_spec	No	Object	Container information. See Table 4-175 .

Table 4-156 environment parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Environment variable name. The value contains 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and dots (.), and cannot start with a digit.
value	Yes	String	Environment variable value.

Table 4-157 storage parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Storage type. Value: <ul style="list-style-type: none">• HostPath: host path mounting.• EmptyDir: temporary directory mounting.• ConfigMap: configuration item mounting.• Secret: secret volume mounting.• PersistentVolumeClaim: cloud storage mounting.
parameters	Yes	Object	Storage parameter. See Table 4-170 .
mounts	Yes	Array of objects	Directory mounted to the container. See Table 4-171 .

Table 4-158 strategy parameter

Parameter	Mandatory	Type	Description
upgrade	No	String	Upgrade policy. Value: Recreate or RollingUpdate (default). The former indicates in-place upgrade while the latter indicates rolling upgrade.

Table 4-159 lifecycle parameters

Parameter	Mandatory	Type	Description
entrypoint	No	Object	Startup command. See Table 4-167 .
post-start	No	Object	Post-start processing. See Table 4-168 .
pre-stop	No	Object	Pre-stop processing. See Table 4-168 .

Table 4-160 scheduler parameters

Parameter	Mandatory	Type	Description
affinity	No	Object	Affinity. See Table 4-166 .
anti-affinity	No	Object	Anti-affinity. See Table 4-166 .

Table 4-161 probes parameters

Parameter	Mandatory	Type	Description
livenessProbe	No	Object	Component liveness probe. See Table 4-162 .
readinessProbe	No	Object	Component service probe. See Table 4-162 .

Table 4-162 probes_object parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Value: http, tcp, or command. The check methods are HTTP request check, TCP port check, and command execution check, respectively.
parameters	Yes	Object	Parameters. <ul style="list-style-type: none">• If type is set to http, see Table 4-163.• If type is set to tcp, see Table 4-165.• If type is set to command, see Table 4-164.
delay	No	Integer	Interval between the startup and detection.
timeout	No	Integer	Detection timeout interval.

Table 4-163 probes_param_http parameters

Parameter	Mandatory	Type	Description
scheme	No	String	Value: HTTP or HTTPS.
host	No	String	Pod IP address (default). You can specify an IP address.
port	No	Integer	Port.
path	No	String	Request path.

Table 4-164 probes_param_command parameters

Parameter	Mandatory	Type	Description
command	No	Array of String	Command list.

Table 4-165 probes_param_tcp parameters

Parameter	Mandatory	Type	Description
port	No	Integer	Port

Table 4-166 scheduler_affinity parameters

Parameter	Mandatory	Type	Description
az	No	Array of String	AZ list.
node	No	Array of String	Node private IP address list.
application	No	Array of String	Component instance list.

Table 4-167 lifecycle_entrypoint parameters

Parameter	Mandatory	Type	Description
command	No	Array of String	Command that can be executed.
args	No	Array of String	Running parameters.

Table 4-168 lifecycle_process parameters

Parameter	Mandatory	Type	Description
type	No	String	Value: command or http. The former indicates command execution and the latter indicates HTTP requests.
parameters	No	Object	Post-start or pre-stop processing parameters. See Table 4-169 .

Table 4-169 lifecycle_process_parameters parameters

Parameter	Mandatory	Type	Description
command	No	Array of String	Command parameters, for example, ["sleep", "1"]. This parameter is applicable to the command type.
host	No	String	Pod IP address (default). You can also specify an IP address. This parameter is applicable to the http type.

Parameter	Mandatory	Type	Description
port	No	Integer	Port number. This parameter is applicable to the http type.
path	No	String	Request URL. This parameter is applicable to the http type.

Table 4-170 storage_parameters parameters

Parameter	Mandatory	Type	Description
path	No	String	Host path. This parameter is applicable to the HostPath storage type.
name	No	String	Name of a configuration item or secret. This parameter is applicable to the ConfigMap and Secret storage type.
defaultMode	No	Integer	Permission to be mounted. The value is in decimal format, for example, 384.
medium	No	String	This parameter is applicable to EmptyDir storage. If it is not specified, the default disk medium is used. If it is set to memory , memory storage is enabled.

Table 4-171 storage_mounts parameters

Parameter	Mandatory	Type	Description
path	Yes	String	Path of the mounted disk.
subPath	No	String	Sub-path of the mounted disk.
readOnly	Yes	boolean	Permission of the mounted disk, which can be read-only or read/write.

Table 4-172 artifact parameters

Parameter	Mandatory	Type	Description
storage	Yes	String	Storage mode. Value: swr or obs.

Parameter	Mandatory	Type	Description
type	Yes	String	Type. Value: package (VM-based deployment) or image (container-based deployment).
url	Yes	String	Software package or image address. For a component deployed on a VM, this parameter is the software package address. For a component deployed based on a container, this parameter is the image address or component name:v\${index} . The latter indicates that the component source code or the image automatically built using the software package will be used.
auth	Yes	String	Authentication mode. Value: iam or none. Default value: iam.
version	No	String	Version number.
properties	No	Map<String, Object>	Property information.

Table 4-173 external_accesses parameters

Parameter	Mandatory	Type	Description
protocol	Yes	String	Protocol. Value: http or https.
address	Yes	String	Access address.
forward_port	Yes	Integer	Port number.

Table 4-174 refer_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID. If type is set to ecs , the value of this parameter must be Default .
type	Yes	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.

Parameter	Mandatory	Type	Description
refer_alias	No	String	Application alias, which is provided only in DCS scenario. Value: "distributed_session", "distributed_cache", "distributed_session, distributed_cache". Default value: "distributed_session, distributed_cache".
parameters	No	Map<String, Object>	Reference resource parameter. NOTICE <ul style="list-style-type: none"> When type is set to cce, this parameter is mandatory. You need to specify the namespace of the cluster where the component is to be deployed. Example: {"namespace": "default"}. When type is set to ecs, this parameter is mandatory. You need to specify the hosts where the component is to be deployed. Example: {"hosts": ["04d9f887-9860-4029-91d1-7d3102903a69", "04d9f887-9860-4029-91d1-7d3102903a70"] }.

Table 4-175 container_spec parameters

Parameter	Mandatory	Type	Description
containers	No	Array of objects	Container information. See Table 4-176 .
type	No	String	Workload type. <ul style="list-style-type: none"> deployment statefulset

Table 4-176 container parameters

Parameter	Mandatory	Type	Description
name	No	String	Container name.
size	No	Object	Container size. See Table 4-177 .
env	No	Array of objects	Environment variable. See Table 4-156 .

Parameter	Mandatory	Type	Description
storage	No	Array of objects	Data storage configuration. See Table 4-157 .
lifecycle	No	Object	Lifecycle. See Table 4-159 .
probes	No	Object	Health check. See Table 4-161 .

Table 4-177 size parameter

Parameter	Mandatory	Type	Description
id	No	String	<p>Resource specifications, which can be obtained by using the API in Obtaining All Supported Flavors of Application Resources.</p> <p>If you need to customize resource specifications, the format is CUSTOM-xxG:xxC-xxC:xxGi-xxGi. Where:</p> <ul style="list-style-type: none"> xxG: storage capacity allocated to a component instance. It is a reserved field. You can set it to a fixed number. xxC-xxC: the maximum and minimum number of CPU cores allocated to a component instance. xxGi-xxGi: the maximum and minimum memory allocated to a component instance. <p>For example, CUSTOM-10G:0.5C-0.25C:1.6Gi-0.8Gi indicates that the maximum number of CPU cores allocated to a component instance is 0.5, the minimum number of CPU cores is 0.25, the maximum memory is 1.6 Gi, and the minimum memory is 0.8 Gi.</p>

Response

Table 4-178 Response parameters

Parameter	Type	Description
job_id	String	Job ID, which is used to query information about the created job.
instance_id	String	Component instance ID.

Example Request

Create component instance **component-instance-name**. Select the environment whose ID is **6e763000-9128-4a9d-adea-34c42cc5344d** as the running environment. Select the **demo.jar** package in the myapp bucket of OBS as the deployment component. Select the CCE resource whose ID is **b6862a62-d916-11e9-bdf1-0255ac101fd9** and the ELB resource whose ID is **8c0a45cc-626f-4d65-8257-507ee059aa9a** as the deployment resources.

```
{
  "name": "component-instance-name",
  "environment_id": "6e763000-9128-4a9d-adea-34c42cc5344d",
  "flavor_id": "MICRO-5G:0.5C:1G",
  "replica": 1,
  "artifacts": {
    "container-name": {
      "storage": "obs",
      "type": "package",
      "url": "obs://myapp/demo.jar",
      "properties": {
        "bucket": "myapp",
        "key": "demo.jar",
        "endpoint": "https://obs.region_id.external_domain_name.com"
      }
    },
    "auth": "iam"
  }
},
"version": 1,
"description": "instance desc",
"configuration": {
  "env": [
    {
      "name": "log-level",
      "value": "warn"
    }
  ]
}
],
"refer_resources": [
  {
    "id": "b6862a62-d916-11e9-bdf1-0255ac101fd9",
    "type": "cce",
    "parameters": {
      "namespace": "default"
    }
  },
  {
    "id": "8c0a45cc-626f-4d65-8257-507ee059aa9a",
    "type": "elb"
  }
]
}
```


Example Response

```
{
  "instance_id": "89f5baf5-efe4-4f12-9c0d-734d2af5a184",
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

Status Code

Table 4-179 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.2 Obtaining All Component Instances

Function

This API is used to obtain all instances of a component.

URI

GET /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances

Table 4-180 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Table 4-181 Query parameters

Parameter	Mandatory	Type	Description
maxWaitTime	No	int	User-defined overall startup time.

Table 4-182 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of records to be displayed. The value is 1000 or ranges from 0 to 100. If the specified value is not within the range, value 10 is assigned. In the non-pagination scenario, the value is 1000 . In the pagination scenario, the value ranges from 0 to 100.
offset	No	Integer	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, version, and update_time.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 4-183 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-184 Response parameters

Parameter	Type	Description
count	Integer	Total number of instances.
instances	Array of objects	Instance parameters. See Table 4-185 .

Table 4-185 instance parameters

Parameter	Type	Description
id	String	Component instance ID.
application_id	String	Application ID.
application_name	String	Application name.
component_id	String	Component ID.
component_name	String	Component name.
name	String	Component instance name.
environment_id	String	Component environment ID.
environment_name	String	Environment name.
platform_type	String	Platform type. Value: cce or vmapp.
artifacts	Map<String, Object>	Artifact. key indicates the component name. In the Docker container scenario, key indicates the container name. See Table 4-186 .
creator	String	Creator.
version	String	Component version.
create_time	Integer	Creation time.
update_time	Integer	Update time.
external_accesses	Array of objects	Access mode. See Table 4-187 .
status_detail	Object	Status details. See Table 4-188 .

Table 4-186 artifact parameters

Parameter	Type	Description
storage	String	Storage mode. Value: swr or obs.
type	String	Type. Value: package (VM-based deployment) or image (container-based deployment).
url	String	Software package or image address.
auth	String	Authentication mode. Value: iam or none. Default value: iam.
version	String	Version number.
properties	Map<String,String>	Property information.

Table 4-187 external_accesses parameters

Parameter	Type	Description
id	String	Access type ID.
protocol	String	Protocol.
address	String	Access address.
forward_port	Integer	Port for listening to an application component process.
type	String	Type.
status	String	Status.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-188 status_detail parameters

Parameter	Type	Description
enterprise_project_id	String	Enterprise project ID.
status	String	Instance status.
available_replica	Integer	Number of normal instance replicas.
replica	Integer	Number of instance replicas.
fail_detail	String	Failure description.

Parameter	Type	Description
last_job_id	String	Latest job ID.
last_job_status	String	Latest job status.

Example Request

None

Example Response

```
{
  "instances": [
    {
      "id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
      "application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
      "application_name": "test-app",
      "component_id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
      "component_name": "test-com",
      "name": "test-micro-test-env-7iafjk",
      "environment_id": "61b81021-21d5-42f3-b80e-0b6bd10dbf7d",
      "environment_name": "test-env",
      "platform_type": "cce",
      "version": "1.0.0",
      "artifacts": {
        "test-micro": {
          "storage": "obs",
          "type": "package",
          "url": "obs://myapp/demo.jar",
          "properties": {
            "bucket": "myapp",
            "key": "demo.jar",
            "endpoint": "https://obs.region_id.external_domain_name.com"
          },
          "auth": "iam",
          "version": "1.0.0",
        }
      },
      "create_time": 1610331819996,
      "update_time": 1610331830398,
      "creator": "liuwei",
      "status_detail": {
        "status": "RUNNING",
        "replica": 1,
        "available_replica": 1,
        "fail_detail": null,
        "last_job_id": "JOB7bc0366c-dc43-41f9-b65f-8d6078038488",
        "last_job_status": "SUCCEEDED",
        "enterprise_project_id": "0"
      }
    }
  ],
  "count": 1
}
```

Status Code

Table 4-189 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.3 Querying the Operations Performed on a Component Instance

Function

This API is used to query the operations performed on a component instance.

URI

POST /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances/{instance_id}/action

Table 4-190 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .
instance_id	Yes	String	Component instance ID. See Obtaining All Component Instances .

Request

Table 4-191 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-192 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Operation. Value: start, stop, restart, scale, or rollback. <ul style="list-style-type: none"> • start • stop • restart • scale • rollback
parameters	No	Object	Operation parameter, which is mandatory when action is set to scale or rollback . See Table 4-193 .

Table 4-193 parameters description

Parameter	Mandatory	Type	Description
replica	No	Integer	Number of instances, which is mandatory when action is set to scale .
hosts	No	Array	ECS ID list. This parameter specifies the ECS deployed during VM scaling, and is mandatory for scaling.

Parameter	Mandatory	Type	Description
version	No	String	Version number, which is mandatory when action is set to rollback . You can obtain the version number by using the API in Obtaining Component Instance Snapshots .

Response

Table 4-194 Response parameters

Parameter	Type	Description
job_id	String	Job ID.

Example Request

Scale component instances and change the number of component instances to 4.

```
{
  "action": "scale",
  "parameters": {
    "replica": 4
  }
}
```

Example Response

```
{
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

Status Code

Table 4-195 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.4 Modifying a Component Instance

Function

This API is used to modify a component instance.

URI

PUT /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances/{instance_id}

Table 4-196 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .
instance_id	Yes	String	Component instance ID. See Obtaining All Component Instances .

Request

Table 4-197 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 4-198 Request body parameters

Parameter	Mandatory	Type	Description
version	Yes	String	Application component version that meets version semantics. Example: 1.0.1.
flavor_id	No	String	<p>Resource specifications, which can be obtained by using the API in Obtaining All Supported Flavors of Application Resources.</p> <p>If you need to customize resource specifications, the format is CUSTOM-xxG:xxm-xxm:xxGi-xxGi. Where:</p> <ul style="list-style-type: none"> xxG: storage capacity allocated to a component instance. It is a reserved field. You can set it to a fixed number. xxm-xxm: the maximum and minimum number of CPU cores allocated to a component instance. xxGi-xxGi: the maximum and minimum memory allocated to a component instance. <p>For example, CUSTOM-10G:0.5m-0.25m:1.6Gi-0.8Gi indicates that the maximum number of CPU cores allocated to a component instance is 0.5, the minimum number of CPU cores is 0.25, the maximum memory is 1.6 Gi, and the minimum memory is 0.8 Gi.</p>
artifacts	No	Map<String, Object>	Artifact. key indicates the component name. In the Docker container scenario, key indicates the container name. See Table 4-199 .
configuration	No	Object	Configuration parameters, such as environment variables, deployment configurations, and O&M monitoring. By default, this parameter is left blank. See Table 4-155 in Creating an Application Component Instance .
description	No	String	Description. The value can contain up to 128 characters.
external_accesses	No	Array of objects	Access mode. See Table 4-200 .
refer_resources	No	Array of objects	Deployed resources. See Table 4-201 .

Table 4-199 artifact parameters

Parameter	Mandatory	Type	Description
storage	Yes	String	Storage mode. Value: swr or obs.
type	Yes	String	Type. Value: package (VM-based deployment) or image (container-based deployment).
url	Yes	String	Software package or image address. For a component deployed on a VM, this parameter is the software package address. For a component deployed based on a container, this parameter is the image address or component name:v\${index} . The latter indicates that the component source code or the image automatically built using the software package will be used.
auth	Yes	String	Authentication mode. Value: iam or none. Default value: iam.
version	No	String	Version number.
properties	No	Map<String,String>	Property information.

Table 4-200 external_accesses parameters

Parameter	Mandatory	Type	Description
id	No	String	Access type ID.
protocol	Yes	String	Protocol. Value: http or https.
address	Yes	String	Access address. Example: www.example.com.
forward_port	Yes	Integer	Port for listening to an application component process.
type	No	String	Type.
status	No	String	Status.
create_time	No	Integer	Creation time.
update_time	No	Integer	Update time.

Table 4-201 refer_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID.
type	Yes	String	Resource type. Example: rds or dcs.
refer_alias	No	String	Application alias. This parameter is available only when type is set to dcs . Value: "distributed_session", "distributed_cache", or "distributed_session, distributed_cache" (default).
parameters	No	Map<String, Object>	Reference resource parameter.

Response

Table 4-202 Response parameters

Parameter	Type	Description
job_id	String	Job ID, which is used to query information about the created job.

Example Request

Modify the application component instance. Change the artifact to the **test-micro** image package in the SWR image repository **swr.roma-dev-1.roma.development.com/ns1/examples:v2**. Change the deployment resources to the CCE resource whose ID is **523498f1-36c4-11eb-ae36-0255ac1000c2** and the professional microservice engine **cse**.

```
{
  "description": "",
  "artifacts": {
    "test-micro": {
      "storage": "swr",
      "type": "image",
      "url": "swr.roma-dev-1.roma.development.com/ns1/examples:v2",
      "auth": "iam",
      "version": "1.0.0",
      "properties": {}
    }
  },
  "refer_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "parameters": {
        "namespace": "default"
      }
    }
  ],
}
```

```
{
  "id": "default",
  "type": "cse"
},
"version": "1.0.2"
}
```

Example Response

```
{
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

Status Code

Table 4-203 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.5 Deleting a Component Instance

Function

This API is used to delete a component instance.

URI

DELETE /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances/{instance_id}

Table 4-204 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Parameter	Mandatory	Type	Description
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .
instance_id	Yes	String	Component instance ID. See Obtaining All Component Instances .

Table 4-205 Query parameters

Parameter	Mandatory	Type	Description
force	No	boolean	Whether to forcibly delete the component.

Request

Table 4-206 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-207 Response parameters

Parameter	Type	Description
job_id	String	Job ID

Example Request

None

Example Response

```
{
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

Status Code

Table 4-208 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.6 Querying Instance Details Based on the Instance ID

Function

This API is used to query instance details based on the instance ID.

URI

GET /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances/{instance_id}

Table 4-209 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .
instance_id	Yes	String	Component instance ID. See Obtaining All Component Instances .

Request

Table 4-210 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-211 Response parameters

Parameter	Type	Description
id	String	Component instance ID.
name	String	Component instance name.
environment_id	String	Component environment ID.
platform_type	String	Platform type. Value: cce or vmapp.
description	String	Instance description.
flavor_id	String	Resource flavor.
artifacts	Map<String, Object>	Artifact. key indicates the component name. In the Docker container scenario, key indicates the container name. See Table 4-212 .
version	String	Component version.
configuration	Map<String, Object>	Component configurations, such as environment variables.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
external_accesses	Array of objects	Access mode. See Table 4-213 .

Parameter	Type	Description
refer_resources	Array of objects	Deployed resources. See Table 4-214 .
status_detail	Object	Status details. See Table 4-215 .

Table 4-212 artifact parameters

Parameter	Type	Description
storage	String	Storage mode. Value: swr or obs.
type	String	Type. Value: package (VM-based deployment) or image (container-based deployment).
url	String	Software package or image address.
auth	String	Authentication mode. Value: iam or none. Default value: iam.
version	String	Version number.
properties	Map<String,String>	Property information.

Table 4-213 external_accesses parameters

Parameter	Type	Description
id	String	Access type ID.
protocol	String	Protocol.
address	String	Access address.
forward_port	Integer	Port for listening to an application component process.
type	String	Type.
status	String	Status.
create_time	Integer	Creation time.
update_time	Integer	Update time.

Table 4-214 refer_resources parameters

Parameter	Type	Description
id	String	Resource ID.

Parameter	Type	Description
type	String	Basic resources: cce, as, and ecs. Optional resources: rds, dcs, elb, and other services.
refer_alias	String	Application alias, which is provided only in DCS scenario. Value: "distributed_session", "distributed_cache", "distributed_session, distributed_cache". Default value: "distributed_session, distributed_cache".
parameters	Map<String, Object>	Reference resource parameter.

Table 4-215 status_detail parameters

Parameter	Type	Description
enterprise_project_id	String	Enterprise project ID.
status	String	Instance status. <ul style="list-style-type: none"> INITIALIZING UPGRADING FAILED RUNNING DELETING STARTING STOPPING STOPPED STARTING PENDING PARTIALLY_FAILED
available_replica	Integer	Number of normal instance replicas.
replica	Integer	Number of instance replicas.
fail_detail	String	Failure description.
last_job_id	String	Latest job ID.
last_job_statuses	String	Latest job status.

Example Request

None

Example Response

```
{
  "id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
  "name": "test-micro-test-env-7iafjk",
  "environment_id": "61b81021-21d5-42f3-b80e-0b6bd10dbf7d",
  "platform_type": "cce",
  "flavor_id": "CUSTOM-10G:250m-250m:0.5Gi-0.5Gi",
  "artifacts": {
    "test-micro": {
      "storage": "swr",
      "type": "image",
      "url": "swr.region_id.development.com/ns/examples:v2",
      "auth": "iam",
      "version": "1.0.0",
      "properties": {}
    }
  },
  "version": "1.0.2",
  "description": "",
  "configuration": {
    "env": [
      {
        "name": "PAAS_PROJECT_ID",
        "value": "e7d2e9c589e5445e808a8ff0d1235aca",
        "internal": true
      }
    ],
    "lifecycle": {
      "post-start": null,
      "pre-stop": null
    },
    "scheduler": {
      "affinity": {
        "az": [],
        "node": [],
        "application": []
      },
      "anti-affinity": {
        "az": [],
        "node": [],
        "application": []
      },
      "isSynchronized": true
    },
    "strategy": {
      "upgrade": "RollingUpdate",
      "spec": {
        "maxSurge": 0,
        "maxUnavailable": 1
      }
    }
  },
  "creator": null,
  "create_time": 1610331819996,
  "update_time": 1610522676792,
  "external_accesses": null,
  "refer_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "parameters": {
        "namespace": "default"
      }
    }
  ],
}
```

```
{
  "id": "default",
  "type": "cse"
},
"status_detail": {
  "status": "RUNNING",
  "replica": 1,
  "available_replica": 1,
  "fail_detail": null,
  "last_job_id": "JOB16c33466-41e6-4c5b-a9ac-2059253a1e5c",
  "last_job_status": "SUCCEEDED",
  "enterprise_project_id": "0"
}
```

Status Code

Table 4-216 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.5.7 Obtaining Component Instance Snapshots

Function

This API is used to obtain component instance snapshots.

Instance snapshots refer to the instance information about the source version. They can be used for version rollback.

URI

GET /v2/{project_id}/cas/applications/{application_id}/components/{component_id}/instances/{instance_id}/snapshots

Table 4-217 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .
instance_id	Yes	String	Component instance ID. See Obtaining All Component Instances .

Table 4-218 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of records to be displayed. The value is 1000 or ranges from 0 to 100. If the specified value is not within the range, value 10 is assigned. In the non-pagination scenario, the value is 1000 . In the pagination scenario, the value ranges from 0 to 100.
offset	No	Integer	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 4-219 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-220 Response parameters

Parameter	Type	Description
count	Integer	Number of snapshots.
snapshots	Array of objects	Snapshot parameters. See Table 4-221 .

Table 4-221 snapshot parameters

Parameter	Type	Description
create_time	Integer	Creation time.
description	String	Description.
instance_id	String	Component instance ID.
version	String	Version number.

Example Request

None

Example Response

```
{
  "snapshots": [
    {
      "version": "1.0.0",
      "description": "{\"action\":\"New Version: 1.0.0\"}",
      "instance_id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
      "create_time": 1610331830841
    }
  ],
  "count": 1
}
```

Status Code

Table 4-222 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

4.6 Deployment Jobs

4.6.1 Obtaining Job Details

Function

This API is used to obtain job details.

URI

GET /v2/{project_id}/cas/jobs/{job_id}

Table 4-223 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
job_id	Yes	String	Deployment job ID, which is job_id in the response parameter in Creating an Application Component Instance .

Table 4-224 Query parameters

Parameter	Mandatory	Type	Description
instance_id	No	String	Component instance ID. See Obtaining All Component Instances .
limit	No	Integer	Number of specified records.
offset	No	Integer	Offset.
desc	No	String	Whether query results are displayed in descending order. Value: true (descending order) or false (ascending order).

Request

Table 4-225 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 4-226 Parameters

Parameter	Type	Description
task_count	Integer	Number of tasks.
job	Object	Job parameters. See Table 4-227 .
tasks	Array<Object>	Task parameters. See Table 4-228 .

Table 4-227 job parameters

Parameter	Type	Description
CREATED_BY	String	Creator.
EXECUTION_STATUS	String	Execution status. <ul style="list-style-type: none"> • SUBMITTED • SCHEDULED • RUNNING • FAILED • SUCCEEDED • CANCELLED • PARTIALLY_FAILED • ROLLBACKING • ROLLBACKED • UNKNOWN
JOB_DESC	String	Job description.
JOB_ID	String	Job ID.
JOB_NAME	String	Job name.
JOB_TYPE	String	Type. <ul style="list-style-type: none"> • PROVISION • DEPROVISION • DAILY_OPERATION • SYSTEM_INNER • UNKNOWN
ORDER_ID	String	Order ID.
PROJECT_ID	String	Tenant's project ID.
SERVICE_INSTANCE_ID	String	Instance ID.

Table 4-228 task parameters

Parameter	Type	Description
CREATED_AT	String	Creation time.
LAST_HEALTH_CHECK	String	Health check time.
MESSAGES	String	Message.
OWNER_ID	String	Creator ID.

Parameter	Type	Description
TASK_ID	String	Task ID.
TASK_INDEX	Integer	Task index.
TASK_NAME	String	Task name.
TASK_STATUS	String	Task status. <ul style="list-style-type: none"> • SUBMITTED • SCHEDULED • RUNNING • FAILED • SUCCEEDED • CANCELLED • PARTIALLY_FAILED • SKIPPED • ROLLBACKING • ROLLBACKED • UNKNOWN
TASK_TYPE	String	Task type. <ul style="list-style-type: none"> • kube: Kubernetes task • restful: RESTful HTTP request task • sleep: sleep waiting task • echo: output task • context • sub-job

Example Request

None

Example Response

```
{
  "job": {
    "JOB_ID": "JOB474362ef-091d-429f-9b3e-c9246b37dee6",
    "JOB_NAME": "provision/provMain",
    "JOB_TYPE": "PROVISION",
    "SERVICE_INSTANCE_ID": "429a35a0-dee8-4648-86f6-d1f8bc74ba52",
    "PROJECT_ID": "063a3158b1a34710b36ad208b9497d00",
    "ORDER_ID": "",
    "JOB_DESC": "provision/provMain",
    "EXECUTION_STATUS": "FAILED",
    "CREATED_BY": "585cc5c16cda4954b28d4eb460f06261"
  },
  "tasks": [
    {
      "MESSAGES": "{}",
      "TASK_ID": "Task-bc64b46c-1688-4787-b7ff-87cb53b346ea",

```

```
"TASK_TYPE": "echo",
"TASK_INDEX": 1,
"TASK_NAME": "SEPARATOR_PREPARE_RESOURCES",
"TASK_STATUS": "SUCCEEDED",
"LAST_HEALTH_CHECK": "2020-02-04T04:01:03.344Z",
"CREATED_AT": "2020-02-04T04:01:03.344Z",
"OWNER_ID": "cas-mgr-69bb87d7c-wpqb:30114"
},
{
  "MESSAGES": "{}",
  "TASK_ID": "Task-15ec252f-bca0-401e-ae5b-798a639df92e",
  "TASK_TYPE": "echo",
  "TASK_INDEX": 4,
  "TASK_NAME": "SEPARATOR_BUILDING",
  "TASK_STATUS": "SUCCEEDED",
  "LAST_HEALTH_CHECK": "2020-02-04T04:01:03.346Z",
  "CREATED_AT": "2020-02-04T04:01:03.345Z",
  "OWNER_ID": "cas-mgr-69bb87d7c-wpqb:30114"
},
{
  "MESSAGES": "{}",
  "TASK_ID": "Task-52a5d1ae-2c54-46ec-8134-53e7b6a45fb7",
  "TASK_TYPE": "restful",
  "TASK_INDEX": 6,
  "TASK_NAME": "TN00091",
  "TASK_STATUS": "SUCCEEDED",
  "LAST_HEALTH_CHECK": "2020-02-04T04:01:08.924Z",
  "CREATED_AT": "2020-02-04T04:01:03.346Z",
  "OWNER_ID": "cas-mgr-69bb87d7c-wpqb:30114"
},
{
  "MESSAGES": "{}",
  "TASK_ID": "Task-a149b585-d31b-4c72-a665-d820efcb0bf9",
  "TASK_TYPE": "restful",
  "TASK_INDEX": 7,
  "TASK_NAME": "TN00094",
  "TASK_STATUS": "SUCCEEDED",
  "LAST_HEALTH_CHECK": "2020-02-04T04:01:08.961Z",
  "CREATED_AT": "2020-02-04T04:01:08.925Z",
  "OWNER_ID": "cas-mgr-69bb87d7c-wpqb:30114"
},
{
  "MESSAGES": "{
    \"Message\": \"ECC00011\",
    \"Detail\": \"Building failed.\"
  }",
  "TASK_ID": "Task-a72fe8d8-8cc1-42b5-a097-0cf9e29692a7",
  "TASK_TYPE": "restful",
  "TASK_INDEX": 9,
  "TASK_NAME": "TN00097",
  "TASK_STATUS": "FAILED",
  "LAST_HEALTH_CHECK": "2020-02-04T04:01:38.975Z",
  "CREATED_AT": "2020-02-04T04:01:08.962Z",
  "OWNER_ID": "cas-mgr-69bb87d7c-wpqb:30114"
}
],
"task_count": 5
}
```

Status Code

Table 4-229 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5 Application Management V3 APIs

5.1 Environment

5.1.1 Creating an Environment

Function

An environment is a collection of compute, storage, and network resources used for deploying and running an application. ServiceStage combines the compute resources (such as CCE clusters and ECSs), network resources (such as ELB instances and EIPs), and middleware (such as DCS instances, RDS instances, and CSE engines) in the same enterprise project and VPC into an environment, such as a development environment, testing environment, pre-production environment, or production environment. The resources within an environment can be networked together. Managing resources and deploying services by environment simplifies O&M.

This API is used to create an environment.

URI

POST /v3/{project_id}/cas/environments

Table 5-1 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 5-2 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-3 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Environment name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
deploy_mode	No	String	Environment type. <ul style="list-style-type: none">● virtualmachine● container: Kubernetes● mixed: VM and Kubernetes.
description	No	String	Environment description. The value can contain up to 128 characters.
enterprise_project_id	No	String	Enterprise project ID.
vpc_id	Yes	String	VPC ID.
labels	No	Array of objects	Label. You can use Tag Management Service (TMS) to filter resources by labels. See Table 5-4 .

Table 5-4 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Response

Table 5-5 Response parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
deploy_mode	String	Environment type.
vpc_id	String	VPC ID.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
labels	Array of objects	Label. You can use TMS to query resources by labels. See Table 5-6 .

Table 5-6 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

Create a VM environment **env-test**. The ID of the VPC where the environment is located is **0c72428b-cd6c-4283-a560-210d3edb49f7**.

```
{
  "name": "env-test",
  "description": "",
  "enterprise_project_id": "0",
  "vpc_id": "0c72428b-cd6c-4283-a560-210d3edb49f7",
  "labels": [
    {
      "key": "aaaaa",
      "value": "a5"
    }
  ],
  "deploy_mode": "virtualmachine"
}
```

Example Response

```
{
  "id": "c93a5313-d141-4b9f-97ce-b4ce2b3988bd",
  "name": "test",
  "description": "",
  "project_id": "13ae26238d724e54947af3e0fcbe9c3",
  "enterprise_project_id": "0",
  "vpc_id": "0c72428b-cd6c-4283-a560-210d3edb49f7",
  "creator": "ss-test",
  "create_time": 1681384236023,
  "update_time": 1681384236023,
  "deploy_mode": "virtualmachine",
  "labels": []
}
```

Status Code

Table 5-7 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.2 Obtaining All Environments

Function

This API is used to obtain all created environments.

URI

GET /v3/{project_id}/cas/environments

Table 5-8 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Table 5-9 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of records to be queried. Value range: 0–100.
offset	No	Integer	Offset, which indicates the result after the Nth data record is queried.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time. If the transferred value is not within the supported enumerated value range, the default sorting field is used.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc
name	No	String	Environment name.
environment_id	No	String	Environment ID.
enterprise_project_id	No	String	Enterprise project ID.

Request

Table 5-10 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-11 Response parameters

Parameter	Type	Description
count	Integer	Total number of environments.
environments	Array of objects	Environment parameters. See Table 5-12 .

Table 5-12 environments parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
deploy_mode	String	Environment type.
vpc_id	String	VPC ID.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-13 .

Table 5-13 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

None

Example Response

```
{
  "environments": [
    {
      "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
      "name": "development-env",
      "description": "",
      "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
      "enterprise_project_id": "0",
      "deploy_mode": "virtualmachine",
      "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
      "creator": "ss-test",
      "create_time": 1610418873730,
      "update_time": 1610418873730,
      "labels": null
    }
  ],
  "count": 1
}
```

Status Code

Table 5-14 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.3 Deleting an Environment Based on the Environment ID

Function

This API is used to delete an environment based on the environment ID.

URI

DELETE /v3/{project_id}/cas/environments/{environment_id}

Table 5-15 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-16 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 5-17 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.4 Modifying an Environment Based on the Environment ID

Function

This API is used to modify an environment based on the environment ID.

URI

PUT /v3/{project_id}/cas/environments/{environment_id}

Table 5-18 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-19 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-20 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Environment name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
description	No	String	Environment description. The value can contain up to 128 characters.
labels	No	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-21 .

Table 5-21 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Response

Table 5-22 Response parameters

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

Parameter	Type	Description
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
deploy_mode	String	Environment type.
vpc_id	String	VPC ID.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-23 .
resources	Array of objects	Resource. See Table 5-24 .

Table 5-23 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Table 5-24 resources parameters

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

Parameter	Type	Description
type	String	Resource type. <ul style="list-style-type: none">• vpc• eip• elb• cce• ecs• as• cse• dcs• rds

Example Request

Modify the environment name to **development-env**, leave the description blank, and add a label whose key-value pair is **"test": "value"** to the environment.

```
{
  "name": "development-env",
  "description": "",
  "labels": [
    {
      "key": "test",
      "value": "value"
    }
  ]
}
```

Example Response

```
{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env",
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "deploy_mode": "virtualmachine",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610418873730,
  "resources": [
    {
      "id": "329b135e-7b31-4ea6-afb9-b69017bd0a80",
      "type": "ecs",
      "name": "ecs-test"
    }
  ],
  "labels": [
    {
      "key": "test",
      "value": "value"
    }
  ]
}
```


Status Code

Table 5-25 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.5 Obtaining Environment Details Based on the Environment ID

Function

This API is used to obtain environment details based on the environment ID.

URI

GET /v3/{project_id}/cas/environments/{environment_id}

Table 5-26 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-27 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-28 Response parameters

Parameter	Type	Description
id	String	Environment ID.
name	String	Environment name.
description	String	Environment description.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
deploy_mode	String	Environment type.
vpc_id	String	VPC ID.
creator	String	Creator.
create_time	Integer	Creation time.
update_time	Integer	Update time.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-29 .
resources	Array of objects	Resource. See Table 5-30 .

Table 5-29 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Table 5-30 resources parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Resource type. <ul style="list-style-type: none"> • vpc • eip • elb • cce • ecs • as • cse • dcs • rds

Example Request

None

Example Response

```
{
  "id": "ea011e01-2eb5-453f-87bf-874e4a855abe",
  "name": "dev-env",
  "description": "develop environment",
  "project_id": "bf8523d898b64e4eb956e3be3555ca16",
  "enterprise_project_id": "0",
  "deploy_mode": "virtualmachine",
  "vpc_id": "234241234124xxvasf2342xxxxxxxx",
  "resources": [
    {
      "id": "329b135e-7b31-4ea6-afb9-b69017bd0a80",
      "type": "ecs",
      "name": "ecs-test"
    }
  ],
  "creator": "ss-test",
  "create_time": 1578984198394,
  "update_time": 1578984198394,
  "labels": null
}
```

Status Code

Table 5-31 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.6 Modifying an Environment Resource Based on the Environment ID

Function

This API is used to modify an environment resource based on environment ID.

URI

PUT /v3/{project_id}/cas/environments/{environment_id}/resources

Table 5-32 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-33 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-34 Request body parameters

Parameter	Mandatory	Type	Description
resources	Yes	Array of objects	Resource to be added. See Table 5-35 .

Table 5-35 resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID. (To ensure that the API can be called properly, enter the correct resource ID.)
name	No	String	Resource name.
type	Yes	String	Resource type. <ul style="list-style-type: none">• eip• elb• cce• ecs• as• cse• dcs• rds

Response

Table 5-36 Response parameters

Parameter	Type	Description
resources	Array of objects	Resource. See Table 5-37 .

Table 5-37 schemas parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Resource type. <ul style="list-style-type: none">• eip• elb• cce• ecs• as• cse• dcs• rds

Example Request

Modify the ECS whose ID is **67835bb3-1235-4cc9-be71-becbb2b4ca0d** and name is **ecs-a5a9**.

```
{
  "resources": [
    {
      "id": "67835bb3-1235-4cc9-be71-becbb2b4ca0d",
      "name": "ecs-a5a9",
      "type": "ecs"
    }
  ]
}
```

Example Response

```
{
  "resources": [
    {
      "id": "67835bb3-1235-4cc9-be71-becbb2b4ca0d",
      "type": "ecs",
      "name": "ecs-a5a9"
    }
  ]
}
```

Status Code

Table 5-38 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.1.7 Querying an Environment Resource Based on the Environment ID

Function

This API is used to query an environment resource based on environment ID.

URI

GET /v3/{project_id}/cas/environments/{environment_id}/resources

Table 5-39 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-40 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-41 Response parameter

Parameter	Type	Description
resources	Array of objects	Resource. See Table 5-42 .

Table 5-42 schemas parameters

Parameter	Type	Description
id	String	Resource ID.
name	String	Resource name.
type	String	Resource type. <ul style="list-style-type: none"> • eip • elb • cce • ecs • as • cse • dcs • rds

Example Request

None

Example Response

```
{
  "resources": [
    {
      "id": "67835bb3-1235-4cc9-be71-becbb2b4ca0d",
      "type": "ecs",
      "name": "ecs-a5a9"
    }
  ]
}
```

Status Code

Table 5-43 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2 Application

5.2.1 Creating an Application

Function

An application is a service system with complete functions and consists of one or more components related to features.

This API is used to create an application.

URI

POST /v3/{project_id}/cas/applications

Table 5-44 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 5-45 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-46 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Application name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.
description	No	String	Application description. The value can contain up to 128 characters.
enterprise_project_id	No	String	Enterprise project ID.
labels	No	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-47 .

Table 5-47 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Response

Table 5-48 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-49 .

Table 5-49 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

Create application **app-xpmtii**, set the description to **test**, and set the label key-value pair to **"key": "var"**.

```
{
  "name": "app-xpmtii",
  "description": "test",
  "labels": [
    {
```

```
    "key": "key",  
    "value": "var"  
  }  
]  
}
```

Example Response

```
{  
  "id": "dae29983-c1e9-437d-a6be-5256076fe605",  
  "name": "app-xpmtii",  
  "description": "test",  
  "project_id": "063a3158b1a34710b36ad208b9497d00",  
  "enterprise_project_id": "0",  
  "creator": "ss-test",  
  "create_time": 1679468852302,  
  "update_time": 1679468852302,  
  "labels": [  
    {  
      "key": "key",  
      "value": "var"  
    }  
  ]  
}
```

Status Code

Table 5-50 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.2 Obtaining All Applications

Function

This API is used to obtain all created applications.

URI

GET /v3/{project_id}/cas/applications

Table 5-51 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. See Obtaining a Project ID .

Table 5-52 Query parameters

Parameter	Mandatory	Type	Description
limit	No	integer	Number of records to be queried. Value range: 0–100.
offset	No	integer	Offset, which indicates the result after the Nth data record is queried.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time. If the transferred value is not within the supported enumerated value range, the default sorting field is used.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 5-53 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-54 Response parameters

Parameter	Type	Description
count	Integer	Total number of applications.
applications	Array of object	Application information. See Table 5-55 .

Table 5-55 applications parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
component_count	Integer	Number of components.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-56 .

Table 5-56 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

None

Example Response

```
{
  "count": 1,
  "applications": [
    {
      "id": "dae29983-c1e9-437d-a6be-5256076fe605",
      "name": "app-xpmtii",
      "description": "test",
      "project_id": "063a3158b1a34710b36ad208b9497d00",
      "enterprise_project_id": "0",
      "creator": "ss-test",
      "create_time": 1679468852302,
      "update_time": 1679468852302,
      "labels": [
        {
          "key": "key",
          "value": "var"
        }
      ],
      "component_count": 0
    }
  ]
}
```

Status Code

Table 5-57 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.3 Modifying Application Information Based on the Application ID

Function

This API is used to modify application information based on the application ID.

URI

PUT /v3/{project_id}/cas/applications/{application_id}

Table 5-58 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 5-59 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-60 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Application name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.
description	No	String	Application description. The value can contain up to 128 characters.
enterprise_project_id	No	String	Enterprise project ID.
labels	No	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-61 .

Table 5-61 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Response

Table 5-62 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
component_count	Integer	Number of components.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-63 .

Table 5-63 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

Modify the application name to **app-xpmtii2**, description to **test2**, and label key-value pair to **"key":"var"**.

```
{  
  "name": "app-xpmtii2",  
  "description": "test2",  
  "labels": [  
    {  
      "key": "var",  
      "value": "test2"  
    }  
  ]  
}
```

```
{
  "key": "key",
  "value": "var"
}
]
```

Example Response

```
{
  "id": "dae29983-c1e9-437d-a6be-5256076fe605",
  "name": "app-xpmtii2",
  "description": "test2",
  "project_id": "063a3158b1a34710b36ad208b9497d00",
  "creator": "ss-test",
  "create_time": 1679468852302,
  "update_time": 1679470928765,
  "component_count": 0,
  "labels": [
    {
      "key": "key",
      "value": "var"
    }
  ]
}
```

Status Code

Table 5-64 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.4 Deleting an Application Based on the Application ID

Function

This API is used to delete an application based on the application ID.

URI

DELETE /v3/{project_id}/cas/applications/{application_id}

Table 5-65 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 5-66 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 5-67 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found

HTTP Status Code	Description
409	Conflict
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.5 Obtaining Application Details Based on the Application ID

Function

This API is used to obtain application details based on the application ID.

URI

GET /v3/{project_id}/cas/applications/{application_id}

Table 5-68 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 5-69 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-70 Response parameters

Parameter	Type	Description
id	String	Application ID.
name	String	Application name.
description	String	Application description.
creator	String	Creator.
project_id	String	Project ID.
enterprise_project_id	String	Enterprise project ID.
create_time	Integer	Creation time.
update_time	Integer	Update time.
component_count	Integer	Number of components.
labels	Array of objects	Label. You can use TMS to filter resources by labels. See Table 5-71 .

Table 5-71 label parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Example Request

None

Example Response

```
{
  "id": "e6129197-f9ca-4d60-b84d-a5e71cd2d317",
  "name": "app-test-0317_03",
  "description": "",
  "project_id": "063a3158b1a34710b36ad208b9497d00",
  "creator": "ss-test",
  "create_time": 1679036601735,
  "update_time": 1679036601735,
  "labels": null,
  "component_count": 0
}
```

Status Code

Table 5-72 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.6 Obtaining Application Configurations Based on the Application ID

Function

This API is used to obtain application configurations based on the application ID.

URI

GET /v3/{project_id}/cas/applications/{application_id}/configuration

Table 5-73 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Parameter	Mandatory	Type	Description
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 5-74 Query parameter

Parameter	Mandatory	Type	Description
environment_id	No	String	Environment ID. See Obtaining All Environments .

Request

Table 5-75 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-76 Response parameter

Parameter	Type	Description
configuration	Array of objects	Configuration list. See Table 5-77 .

Table 5-77 configuration parameters

Parameter	Type	Description
application_id	String	Application ID.

Parameter	Type	Description
environment_id	String	Environment ID.
configuration	Object	Environment variable configuration. See Table 5-78 .

Table 5-78 configurationEnv parameter

Parameter	Type	Description
env	Array of object	Environment variable. See Table 5-79 .

Table 5-79 env parameters

Parameter	Type	Description
name	String	Variable name.
value	String	Variable value.

Example Request

None

Example Response

```
{
  "configuration": [
    {
      "application_id": "fdd7889d-eef8-4db1-befc-7037b02d949e",
      "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
      "configuration": {
        "env": [
          {
            "name": "app-env",
            "value": "env"
          }
        ]
      }
    }
  ]
}
```


Status Code

Table 5-80 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.7 Modifying Application Configurations Based on the Application ID

Function

This API is used to modify application configurations based on the application ID.

URI

PUT /v3/{project_id}/cas/applications/{application_id}/configuration

Table 5-81 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 5-82 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-83 Request body parameters

Parameter	Mandatory	Type	Description
environment_id	Yes	String	Environment ID. See Obtaining All Environments .
configuration	Yes	Object	Environment variable configuration. See Table 5-84 .

Table 5-84 configuration parameter

Parameter	Mandatory	Type	Description
env	Yes	Array of objects	Environment variable. See Table 5-85 .

Table 5-85 env parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Variable name.
value	Yes	String	Variable value.

Response

Table 5-86 Response parameters

Parameter	Type	Description
application_id	String	Application ID.
environment_id	String	Environment ID.
configuration	Object	Environment variable configuration. See Table 5-87 .

Table 5-87 configuration parameter

Parameter	Type	Description
env	Array of object	Environment variable. See Table 5-88 .

Table 5-88 env parameters

Parameter	Type	Description
name	String	Variable name.
value	String	Variable value.

Example Request

Modify the environment whose ID is **0fa75dfe-0d32-4bc1-848b-8008cf3f2567** by changing the name to **app-env** and the value to **env**.

```
{
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {
        "name": "app-env",
        "value": "env"
      }
    ]
  }
}
```

Example Response

```
{
  "application_id": "fdd7889d-eef8-4db1-befc-7037b02d949e",
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {

```

```
    "name": "app-env",  
    "value": "env"  
  }  
]  
}
```

Status Code

Table 5-89 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.2.8 Deleting Application Configurations Based on the Application ID

Function

This API is used to delete application configurations based on the application ID.

URI

DELETE /v3/{project_id}/cas/applications/{application_id}/configuration

Table 5-90 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 5-91 Query parameter

Parameter	Mandatory	Type	Description
environment_id	Yes	String	Environment ID. See Obtaining All Environments .

Request

Table 5-92 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 5-93 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request
404	Not Found
409	Conflict

HTTP Status Code	Description
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3 Component

5.3.1 Creating an Application Component

Function

An application component implements a service feature of an application. It is in the form of code or software packages and can be deployed independently.

This API is used to create an application component.

URI

POST /v3/{project_id}/cas/applications/{application_id}/components

Table 5-94 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Request

Table 5-95 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-96 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Application component name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.
labels	No	Array of objects	Label. For container-based deployment, you can query a workload by label. For VM-based deployment, you can configure a label of the corresponding microservice name to use graceful startup and shutdown. See Table 5-97 .
runtime_stack	Yes	Object	Runtime. See Table 5-123 . The value can be obtained from the response of the API for Querying a Runtime System Stack .
environment_id	No	String	Environment ID.
description	No	String	Description. The value can contain up to 128 characters.
source	Yes	Object	Source of the code or software package. See Table 5-124 .
build	No	Object	Component building. This parameter is mandatory when the component is deployed in containers and no image is used. For details, see Table 5-125 .
limit_cpu	No	Number	Maximum CPU limit. Unit: core.
limit_memory	No	Number	Maximum memory size. Unit: GiB.
request_cpu	No	Number	Applied CPU resources. Unit: core.

Parameter	Mandatory	Type	Description
request_memory	No	Number	Applied memory. Unit: GiB.
version	Yes	String	Component version number. The value can contain up to 32 characters. It needs to meet the following rule: <code>^([0-9]+)(.[0-9]+){2,3}\$</code>
envs	No	Array of objects	Environment variable list. See Table 5-121 .
replica	Yes	Integer	Number of instances.
storages	No	Array of objects	Storage. See Table 5-98 .
deploy_strategy	No	Object	Component deployment. See Table 5-101 .
command	No	Object	Startup command. See Table 5-104 .
post_start	No	Object	Post-start processing. See Table 5-105 .
pre_stop	No	Object	Pre-stop processing. See Table 5-105 .
mesher	No	Object	Multi-language access service mesh. This parameter is mandatory when Node.js is selected for the technology stack and the CSE engine is bound. See Table 5-106 .
timezone	No	String	Time zone where the component runs. Example: Asia/Shanghai.
jvm_opts	No	String	JVM parameter. Example: <code>-Xms256m -Xmx1024m</code> . If there are multiple parameters, separate them by spaces. If the parameter is left blank, the default value is used.
tomcat_opts	No	Object	Tomcat parameter. This parameter is valid when you use the Tomcat technology stack. See Table 5-107 .

Parameter	Mandatory	Type	Description
logs	No	Array of objects	Log collection. See Table 5-113 .
custom_metrics	No	Object	Custom metrics. See Table 5-114 .
affinity	No	Object	Affinity. See Table 5-115 .
anti_affinity	No	Object	Anti-affinity. See Table 5-115 .
liveness_probe	No	Object	Component liveness probe. See Table 5-118 .
readiness_probe	No	Object	Component service probe. See Table 5-118 .
refer_resources	Yes	Array of objects	Associated resource. See Table 5-119 .
external_accesses	No	Array of objects	External access. See Table 5-127 .

Table 5-97 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Table 5-98 storages parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Data storage type. <ul style="list-style-type: none"> • HostPath: host path for local disk mounting. • EmptyDir: temporary directory for local disk mounting. • ConfigMap: configuration item for local disk mounting. • Secret: secrets for local disk mounting. • PersistentVolumeClaim: cloud storage mounting.
name	Yes	String	Name of the storage disk.
parameters	Yes	Object	For details about the parameters of each data storage type, see Table 5-99 .
mounts	Yes	Array of objects	Mount path of data storage. See Table 5-100 .

Table 5-99 component_storage_parameters parameters

Parameter	Mandatory	Type	Description
path	No	String	Host path, which is applicable to HostPath storage. This parameter is mandatory when type is set to HostPath .
name	No	String	Name of a configuration item, secret, or PVC, which is applicable to ConfigMap, Secret, or PersistentVolumeClaim storage, respectively. This parameter is mandatory when type is set to ConfigMap , Secret , or PersistentVolumeClaim .
default_mode	No	Integer	Mounting permission. The value is in decimal format. Example: 384. This parameter is applicable to ConfigMap and secret storage.
medium	No	String	This parameter is applicable to EmptyDir storage. If it is not specified, the default disk medium is used. If it is set to memory , memory storage is enabled.

Table 5-100 mounts parameters

Parameter	Mandatory	Type	Description
path	Yes	String	Mount path.
sub_path	Yes	String	Subpath of the mount path.
read_only	Yes	Boolean	Read only or not.

Table 5-101 deploy_strategy parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Deployment type. <ul style="list-style-type: none">• OneBatchRelease: single-batch upgrade.• RollingRelease: rolling deployment and upgrade.• GrayRelease: dark launch upgrade.
rolling_release	No	Object	Rolling deployment parameter, which is mandatory when type is set to RollingRelease . For details, see Table 5-102 .
gray_release	No	Object	Dark launch upgrade parameter, which is mandatory when type is set to GrayRelease . For details, see Table 5-103 .

Table 5-102 rolling_release parameter

Parameter	Mandatory	Type	Description
batches	Yes	Integer	Deployment batches.

Table 5-103 gray_release parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Dark launch policy. weight : The upgrade is performed based on dark launch traffic.
first_batch_weight	Yes	Integer	Proportion of first-batch dark launch traffic.

Parameter	Mandatory	Type	Description
first_batch_replica	Yes	Integer	Number of first-batch dark launch instances.
remaining_batches	Yes	Integer	Number of batches for remaining instance deployment. After the first batch is completed, the remaining instances will be deployed in specified batches. Example: If there are 5 instances remaining to be deployed in 3 batches, these 5 instances will be upgraded based on 2:2:1.

Table 5-104 Command parameters

Parameter	Mandatory	Type	Description
command	No	Array of String	Command for controlling container running.
args	No	Array of String	Parameter for controlling container running. Example: -port=8080. If there are multiple parameters, separate them by line breaks.

Table 5-105 component_lifecycle parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Processing method. <ul style="list-style-type: none"> • http • command
scheme	No	String	HTTP request type, which can be HTTP or HTTPS. This parameter is valid only when type is set to http .
host	No	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	No	Integer	Port number. This parameter is valid only when type is set to http .

Parameter	Mandatory	Type	Description
path	No	String	Request path. This parameter is valid only when type is set to http .
command	No	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-106 mesher parameters

Parameter	Mandatory	Type	Description
port	Yes	Integer	Process listening port.

Table 5-107 tomcat_opt parameter

Parameter	Mandatory	Type	Description
server_xml	Yes	String	<ol style="list-style-type: none">1. Listening port in the server.xml file will take effect even if it has been configured in public network access.2. The custom Tomcat configuration may conflict with the existing port. Check whether the custom port is occupied.

Table 5-108 host_aliases parameters

Parameter	Mandatory	Type	Description
ip	Yes	String	IP address.
hostname	Yes	Array of String	Host alias.

Table 5-109 dns_config parameters

Parameter	Mandatory	Type	Description
nameservers	No	Array of String	IP address list of the DNS server of the pod.
searches	No	Array of String	(Optional) List of DNS search domains used to search for host names in the pod.
options	No	Array of objects	(Optional) List of objects. Each object may have a name (mandatory) and a value (optional). The contents of the objects here will be merged into the options generated from the specified DNS policy. Duplicate entries will be deleted. See Table 5-110 .

Table 5-110 options parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Operation name.
value	No	String	Operation value.

Table 5-111 security_context parameters

Parameter	Mandatory	Type	Description
run_as_user	No	Integer	User for running containers. For example, to run containers as user root , set the user ID to 0 .
run_as_group	No	Integer	Owner group specified for running container processes.
capabilities	No	Object	Capability set. See Table 5-112 .

Table 5-112 capabilities parameters

Parameter	Mandatory	Type	Description
add	No	Array of String	Grant the Linux permission to the running user.
drop	No	Array of String	Remove the Linux permission of the running user.

Table 5-113 logs parameters

Parameter	Mandatory	Type	Description
log_path	Yes	String	Container log path.
rotate	Yes	String	Interval for dumping logs.
host_path	Yes	String	Mounted host path.
host_extend_path	Yes	String	Host extension path. By extending the host path, you can distinguish mounting from different containers in the same host path. <ul style="list-style-type: none">• None: the extended path is not used.• PodUID: extend the host path based on the pod ID.• PodName: extend the host path based on the pod name.• PodUID/ContainerName: extend the host path based on the pod ID and container name.• PodName/ContainerName: extend the host path based on the pod name and container name.

Table 5-114 custom_metric parameters

Parameter	Mandatory	Type	Description
path	Yes	String	Collection path. Example: /metrics.
port	Yes	Integer	Collection port. Example: 9090.
dimensions	Yes	String	Monitoring dimension. Example: cpu_usage or mem_usage.

Table 5-115 component_affinity parameters

Parameter	Mandatory	Type	Description
az	No	Array of string	List of AZs.
node	No	Array of string	Node list.
component	No	Array of objects	Component list. See Table 5-116 .

Table 5-116 affinity_component parameters

Parameter	Mandatory	Type	Description
displayName	No	String	Workload name.
name	No	String	Component name.

Table 5-117 match_expressions parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Condition key.
value	Yes	String	Condition value.
operation	Yes	String	Operation

Table 5-118 component_probe parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Type. Value: http, tcp, or command.
delay	Yes	Integer	Interval between the startup and detection.
timeout	Yes	Integer	Detection timeout interval.
scheme	No	String	Request type, which is HTTP or HTTPS. This parameter is valid only when type is set to http .

Parameter	Mandatory	Type	Description
host	No	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	No	Integer	Port number. This parameter is valid only when type is set to http or tcp .
path	No	String	Request path. This parameter is valid only when type is set to http .
command	No	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-119 refer_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID.
type	Yes	String	Resource type. <ul style="list-style-type: none">• vpc• eip• elb• cce• ecs• as• cse• dcs• rds
parameters	No	Object	Resource parameters. See Table 5-120 .

Table 5-120 refer_resource_parameter parameter

Parameter	Mandatory	Type	Description
namespace	No	String	Namespace.

Table 5-121 env parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Variable name.
value	No	String	Variable value.
value_from	No	Object	Variable reference. See Table 5-122 .

Table 5-122 value_from parameters

Parameter	Mandatory	Type	Description
reference_type	Yes	String	Reference type. <ul style="list-style-type: none"> configMapKey secretKey
name	Yes	String	Name of the configuration item or secret.
key	No	String	Key of the configuration item or secret.
optional	No	Boolean	Whether the key of the configuration item or secret is mandatory.

Table 5-123 runtime_stack parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Technology stack name.
type	Yes	String	Technology stack type. Value: Java, Tomcat, Nodejs, Php, Docker, or Python. When deploy_mode is set to virtualmachine , only Java, Tomcat, and Node.js are supported. When deploy_mode is set to container , all of the preceding types are supported.
version	Yes	String	Technology stack version.
deploy_mode	Yes	String	Deployment mode. <ul style="list-style-type: none"> container virtualmachine

Table 5-124 source parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	Component source kind. <ul style="list-style-type: none">codepackageimage
version	No	String	Version number.
url	No	String	Package address. This parameter is mandatory when kind is set to package or image .
storage	No	String	Storage mode. Value: swr, codearts, or obs. This parameter is mandatory when kind is set to package or image .
auth	No	String	Authentication mode. Value: iam or none. Default value: iam.
repo_auth	No	String	Authorization name, which can be obtained from the authorization list.
repo_namespace	No	String	Namespace of the code repository.
repo_ref	No	String	Code branch or tag. Default value: master.
repo_type	No	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
web_url	No	String	Code redirection URL. Example: https://github.com/example/demo.git.
repo_url	No	String	Code repository URL. Example: https://github.com/example/demo.git.

Table 5-125 build parameter

Parameter	Mandatory	Type	Description
parameters	No	Map<String, Object>	This parameter is provided only when no ID is available during build creation. See Table 5-126 .

Table 5-126 parameters

Parameter	Mandatory	Type	Description
build_cmd	No	String	Compilation command. By default: 1. When build.sh exists in the root directory, the command is ./build.sh . 2. When build.sh does not exist in the root directory, the command varies depending on the OS. Example: <ul style="list-style-type: none">• Java and Tomcat: mvn clean package• Node.js: npm build
dockerfile_path	No	String	Address of the Docker file. By default, the Docker file is in the root directory (./).
artifact_namespace	No	String	Build archive organization. Default value: cas_{project_id}.
cluster_id	Yes	String	ID of the cluster to be built.
node_label_selector	No	Map<String, String>	key : key of the tag. value : value of the tag.
environment_id	No	String	Environment ID.

Table 5-127 external_accesses parameters

Parameter	Mandatory	Type	Description
protocol	No	String	External access type. Value: http or https.
address	No	String	External access address.
forward_port	No	Integer	External access port.

Response

Table 5-128 Response parameters

Parameter	Type	Description
job_id	String	Task ID.
component_id	String	Application component ID.

Example Request

- Scenario 1: Create a component in the CCE container environment

Create a component named **mycomponent**. The technology stack is Java 1.2.0. The component source is the **weather-1.0.0.jar** package stored in the **bucket0001** bucket of OBS. The resource used for deployment is the CCE whose ID is **fd92bfab-73b3-11ee-90f0-0255ac1001b3**.

```
{
  "name": "mycomponent",
  "description": "",
  "labels": [
    {
      "key": "com-key",
      "value": "com-value"
    }
  ],
  "version": "2023.1102.17105",
  "environment_id": "b489f7b4-2539-45a1-ba71-4728961434ca",
  "runtime_stack": {
    "name": "OpenJDK8",
    "version": "1.2.0",
    "type": "Java",
    "deploy_mode": "container"
  },
  "source": {
    "kind": "package",
    "url": "obs://bucket0001/weather-1.0.0.jar",
    "version": "",
    "storage": "obs"
  },
  "tomcat_opts": {
    "server_xml": ""
  },
  "refer_resources": [
    {
      "id": "fd92bfab-73b3-11ee-90f0-0255ac1001b3",
      "type": "cce",
    }
  ],
  "replica": 1,
  "limit_cpu": 0.25,
  "limit_memory": 0.5,
  "request_cpu": 0.25,
  "request_memory": 0.5
}
```

- Scenario 2: Create a component in the ECS VM environment

Create a component named **mycomponent**. The technology stack is Java 1.2.0. The component source is the **weather-1.0.0.jar** package stored in the **bucket0001** bucket of OBS. The resource used for deployment is the ECS whose ID is **67835bb3-1235-4cc9-be71-becbb2b4ca0d**.

```
{
  "name": "mycomponent",
  "description": "",
  "labels": [
    {
      "key": "com-key",
      "value": "com-value"
    }
  ],
  "version": "2023.1102.17540",
  "environment_id": "9a075df5-104f-4e21-9dbf-ffc3572effdf",
  "runtime_stack": {
```

```
{
  "name": "OpenJDK8",
  "version": "1.2.0",
  "type": "Java",
  "deploy_mode": "virtualmachine"
},
"source": {
  "kind": "package",
  "url": "obs://bucket0001/weather-1.0.0.jar",
  "version": "",
  "storage": "obs"
},
"tomcat_opts": {
  "server_xml": ""
},
"refer_resources": [
  {
    "id": "23598aa9-7a6b-43c7-a53a-39ca5e7290fb",
    "type": "ecs"
  }
],
"replica": 1,
"external_accesses": [
  {
    "protocol": "http",
    "address": "192.168.0.169",
    "forward_port": 80
  }
]
}
```

Example Response

```
{
  "job_id": "JOB8be53b2a-af9a-4e2b-8505-096688a52237",
  "component_id": "b8702b0f-94d3-4822-98a1-56815632a0a0"
}
```

Status Code

Table 5-129 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.2 Obtaining All Components of an Application

Function

This API is used to obtain all components of an application.

URI

GET /v3/{project_id}/cas/applications/{application_id}/components

Table 5-130 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .

Table 5-131 Query parameters

Parameter	Mandatory	Type	Description
limit	No	integer	Number of records to be queried. Value range: 0–100.
offset	No	integer	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 5-132 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-133 Response parameters

Parameter	Type	Description
count	Integer	Total number of components.
components	Array of objects	Component parameters. See Table 5-134 .

Table 5-134 components parameters

Parameter	Type	Description
id	String	Component ID.
name	String	Component name.
labels	Array of objects	Label. For container-based deployment, you can query a workload by label. For VM-based deployment, you can configure a label of the corresponding microservice name to use graceful startup and shutdown. See Table 5-139 .
status	Object	Component status. See Table 5-137 .
runtime_stack	Object	Technology stack. See Table 5-136 .
application_name	String	Application name.
application_id	String	Application ID.

Parameter	Type	Description
environment_name	String	Environment name.
environment_id	String	Environment ID.
version	String	Version number.
source	Object	Source of the code or software package. See Table 5-135 .
creator	String	Creator.
platform_type	String	Deployment type.
external_accesses	Array of objects	External access. See Table 5-140 .

Table 5-135 source parameters

Parameter	Type	Description
kind	String	Component source kind. <ul style="list-style-type: none">• code• package• image
version	String	Version number.
url	String	Package address. This parameter is mandatory when kind is set to package or image .
storage	String	Storage mode. Value: swr, codearts, or obs. This parameter is mandatory when kind is set to package or image .
auth	String	Authentication mode. Value: iam or none. Default value: iam.
repo_auth	String	Authorization name, which can be obtained from the authorization list.
repo_namespace	String	Namespace of the code repository.
repo_ref	String	Code branch or tag. Default value: master.
repo_type	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
web_url	String	Code redirection URL. Example: https://github.com/example/demo.git .

Parameter	Type	Description
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.

Table 5-136 runtime_stack parameters

Parameter	Type	Description
name	String	Technology stack name. Example: OpenJDK8.
type	String	Technology stack type. Example: Java.
version	String	Technology stack version. Example: 1.1.4.
deploy_mode	String	Deployment mode. <ul style="list-style-type: none"> • container • virtualmachine

Table 5-137 status parameters

Parameter	Type	Description
component_status	String	Component status. <ul style="list-style-type: none"> • INITIALIZING • UPGRADING • FAILED • RUNNING • DOWN • DELETING • DELETED • RESERVED • STARTING • STOPPING • STOPPED • RESTARTING • PENDING • UNKNOWN • PARTIALLY_FAILED
available_replica	Integer	Available instances.
replica	Integer	Total instances.

Parameter	Type	Description
fail_detail	String	Failure cause. <ul style="list-style-type: none">• cluster_deleted• cluster_unavailable• cluster_inaccessible• namespace_deleted• namespace_unavailable• namespace_inaccessible• resource_deleted• create_failed• delete_failed
last_job_id	String	ID of the last executed task.
create_time	Integer	Creation time.
update_time	Integer	Update time.
creator	String	Creator.
artifact	Object	Artifact information. See Table 5-138 .

Table 5-138 artifact parameters

Parameter	Type	Description
type	String	Component type. package : VM-based deployment. image : other deployment modes.
url	String	Software package or image address.

Table 5-139 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Table 5-140 external_accesses parameters

Parameter	Type	Description
protocol	String	External access type. Value: http or https.

Parameter	Type	Description
address	String	External access address.
forward_port	Integer	External access port.

Example Request

None

Example Response

```
{
  "components": [
    {
      "source": {
        "kind": "package",
        "url": "obs://bucket0001/weather-1.0.0.jar",
        "storage": "obs"
      },
      "name": "mycomponent",
      "id": "b8702b0f-94d3-4822-98a1-56815632a0a0",
      "labels": null,
      "runtime_stack": {
        "name": "OpenJDK8",
        "deploy_mode": "virtualmachine",
        "version": "1.1.1"
      },
      "external_accesses": null,
      "status": {
        "component_status": "RUNNING",
        "available_replica": 1,
        "replica": 1,
        "fail_detail": null,
        "last_job_id": "JOB8be53b2a-af9a-4e2b-8505-096688a52237",
        "creator": "ss-test",
        "create_time": 1679556221028,
        "update_time": 1679556318779
      },
      "environment_name": "sudo",
      "environment_id": "4d084044-0b80-4641-963c-b9c9f4092a4f",
      "application_name": "app-39mbbn",
      "application_id": "fc092465-a5fb-4a52-bc65-b735f18366d8",
      "creator": "ss-test",
      "platform_type": "cce",
      "version": "2023.0323.15181"
    }
  ],
  "count": 1
}
```

Status Code

Table 5-141 Status codes

HTTP Status Code	Description
200	OK

HTTP Status Code	Description
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.3 Obtaining All Components

Function

This API is used to obtain all components.

URI

GET /v3/{project_id}/cas/components

Table 5-142 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Table 5-143 Query parameters

Parameter	Mandatory	Type	Description
limit	No	integer	Number of records to be queried. Value range: 0–100.
offset	No	integer	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time.

Parameter	Mandatory	Type	Description
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc
application_name	No	String	Application name.
component_name	No	String	Component name.

Request

Table 5-144 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-145 Response parameters

Parameter	Type	Description
count	Integer	Total number of components.
components	Array of objects	Component parameters. See Table 5-146 .

Table 5-146 components parameters

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

Parameter	Type	Description
labels	Array of objects	Label. For container-based deployment, you can query a workload by label. For VM-based deployment, you can configure a label of the corresponding microservice name to use graceful startup and shutdown. See Table 5-151 .
status	Object	Component status. See Table 5-149 .
runtime_stack	Object	Technology stack. See Table 5-148 .
application_name	String	Application name.
application_id	String	Application ID.
environment_name	String	Environment name.
environment_id	String	Environment ID.
version	String	Version number.
source	Object	Source of the code or software package. See Table 5-147 .
creator	String	Creator.
platform_type	String	Deployment type.
external_accesses	Array of Objects	External access. See Table 5-152 .

Table 5-147 source parameters

Parameter	Type	Description
kind	String	Component source kind. <ul style="list-style-type: none">• code• package• image
version	String	Version number.
url	String	Package address. This parameter is mandatory when kind is set to package or image .
storage	String	Storage mode, for example, obs.
auth	String	Authentication mode, for example, iam.
repo_auth	String	Authorization name.

Parameter	Type	Description
repo_namespace	String	Namespace of the code repository.
repo_ref	String	Code branch or tag, for example, master.
repo_type	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
web_url	String	Code redirection URL. Example: https://github.com/example/demo.git.
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.

Table 5-148 runtime_stack parameters

Parameter	Type	Description
name	String	Technology stack name.
type	String	Technology stack type.
version	String	Technology stack version.
deploy_mode	String	Deployment mode.

Table 5-149 status parameters

Parameter	Type	Description
component_status	String	Component status. <ul style="list-style-type: none">INITIALIZINGUPGRADINGFAILEDRUNNINGDOWNDELETINGDELETEDRESERVEDSTARTINGSTOPPINGSTOPPEDRESTARTINGPENDINGUNKNOWNPARTIALLY_FAILED
available_replica	Integer	Available instances.
replica	Integer	Total instances.
fail_detail	String	Failure cause. <ul style="list-style-type: none">cluster_deletedcluster_unavailablecluster_inaccessiblenamespace_deletednamespace_unavailablenamespace_inaccessibleresource_deletedcreate_faileddelete_failed
last_job_id	String	ID of the last executed task.
create_time	Integer	Creation time.
update_time	Integer	Update time.
creator	String	Creator.
artifact	Object	Artifact information. See Table 5-150 .

Table 5-150 artifact parameters

Parameter	Type	Description
type	String	Component type. package : VM-based deployment. image : other deployment modes.
url	String	Software package or image address.

Table 5-151 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Table 5-152 external_accesses parameters

Parameter	Type	Description
protocol	String	External access type. Value: http or https.
address	String	External access address.
forward_port	Integer	External access port.

Example Request

None

Example Response

```
{
  "components": [
    {
      "source": {
        "kind": "package",
        "url": "obs://bucket0001/weather-1.0.0.jar",
        "storage": "obs"
      },
      "name": "mycomponent",
      "id": "b8702b0f-94d3-4822-98a1-56815632a0a0",
      "labels": null,
      "runtime_stack": {
        "name": "OpenJDK8",
        "deploy_mode": "virtualmachine",
        "version": "1.1.1"
      },
      "external_accesses": null,
      "status": {
        "component_status": "RUNNING",
        "available_replica": 1,
        "replica": 1,
        "fail_detail": null,
        "last_job_id": "JOB8be53b2a-af9a-4e2b-8505-096688a52237",
      }
    }
  ]
}
```

```
    "creator": "ss-test",
    "create_time": 1679556221028,
    "update_time": 1679556318779
  },
  "environment_name": "sudo",
  "environment_id": "4d084044-0b80-4641-963c-b9c9f4092a4f",
  "application_name": "app-39mbbn",
  "application_id": "fc092465-a5fb-4a52-bc65-b735f18366d8",
  "creator": "ss-test",
  "platform_type": "cce",
  "version": "2023.0323.15181"
}
],
"count": 1
}
```

Status Code

Table 5-153 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.4 Modifying Component Information by Component ID

Function

This API is used to modify component information by component ID.

URI

PUT /v3/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 5-154 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 5-155 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-156 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Application component name. This value cannot be modified. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit.

Parameter	Mandatory	Type	Description
labels	No	Array of objects	Label. For container-based deployment, you can query a workload by label. For VM-based deployment, you can configure a label of the corresponding microservice name to use graceful startup and shutdown. See Table 5-157 .
runtime_stack	Yes	Object	Runtime. This value cannot be modified. See Table 5-183 . The value can be obtained from the response of the API for Querying a Runtime System Stack .
description	No	String	Description. The value can contain up to 128 characters.
source	Yes	Object	Source of the code or software package. See Table 5-184 .
build	No	Object	Component build. See Table 5-185 .
limit_cpu	No	Number	Maximum CPU limit. Unit: core.
limit_memory	No	Number	Maximum memory size. Unit: GiB.
request_cpu	No	Number	CPU resources in application. Unit: core.
request_memory	No	Number	Applied memory. Unit: GiB.
version	Yes	String	Component version number. The value can contain up to 32 characters. It needs to meet the following rule: $^([0-9]+)(.[0-9]+){2,3}$$
envs	No	Array of objects	Environment variable list. See Table 5-181 .
replica	Yes	Integer	Number of instances. This parameter can be modified only when the scaling function is used.
storages	No	Array of objects	Storage. See Table 5-158 .

Parameter	Mandatory	Type	Description
deploy_strategy	No	Object	Component deployment. See Table 5-161 .
command	No	Object	Startup command. See Table 5-164 .
post_start	No	Object	Post-start processing. See Table 5-165 .
pre_stop	No	Object	Pre-stop processing. See Table 5-165 .
mesher	No	Object	Multi-language access service mesh. This parameter is mandatory when Node.js is selected for the technology stack and the CSE engine is bound. See Table 5-166 .
timezone	No	String	Time zone where the component runs. Example: Asia/Shanghai.
jvm_opts	No	String	JVM parameter. Example: -Xms256m -Xmx1024m. If there are multiple parameters, separate them by spaces. If the parameter is left blank, the default value is used.
tomcat_opts	No	Object	Tomcat parameter. This parameter is valid when you use the Tomcat technology stack. See Table 5-167 .
logs	No	Array of objects	Log collection. See Table 5-173 .
custom_metric	No	Object	Custom metrics. See Table 5-174 .
affinity	No	Object	Affinity. See Table 5-175 .
anti_affinity	No	Object	Anti-affinity. See Table 5-175 .
liveness_probe	No	Object	Component liveness probe. See Table 5-178 .
readiness_probe	No	Object	Component service probe. See Table 5-178 .
refer_resources	No	Array of objects	Associated resource. See Table 5-179 .

Parameter	Mandatory	Type	Description
external_accesses	No	Array of objects	External access. See Table 5-187 .

Table 5-157 labels parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Label name.
value	Yes	String	Label value.

Table 5-158 storages parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Data storage type. <ul style="list-style-type: none"> • HostPath: host path for local disk mounting. • EmptyDir: temporary directory for local disk mounting. • ConfigMap: configuration item for local disk mounting. • Secret: secrets for local disk mounting. • PersistentVolumeClaim: cloud storage mounting.
name	Yes	String	Name of the storage disk.
parameters	Yes	Object	For details about the parameters of each data storage type, see Table 5-159 .
mounts	Yes	Array of objects	Mount path of data storage. See Table 5-160 .

Table 5-159 component_storage_parameters parameters

Parameter	Mandatory	Type	Description
path	No	String	Host path. This parameter is applicable to the HostPath storage type. This parameter is mandatory when type is set to HostPath .
name	No	String	Name of a configuration item, secret, or PVC, which is applicable to ConfigMap, Secret, or PersistentVolumeClaim storage, respectively. This parameter is mandatory when type is set to ConfigMap , Secret , or PersistentVolumeClaim .
default_mode	No	Integer	Mounting permission. The value is in decimal format. Example: 384. This parameter is applicable to ConfigMap and secret storage.
medium	No	String	This parameter is applicable to EmptyDir storage. If it is not specified, the default disk medium is used. If it is set to memory , memory storage is enabled.

Table 5-160 mounts parameters

Parameter	Mandatory	Type	Description
path	Yes	String	Mount path.
sub_path	Yes	String	Subpath of the mount path.
read_only	Yes	Boolean	Read only or not.

Table 5-161 deploy_strategy parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Deployment type. <ul style="list-style-type: none"> • OneBatchRelease: single-batch upgrade. • RollingRelease: rolling deployment and upgrade. • GrayRelease: dark launch upgrade.

Parameter	Mandatory	Type	Description
rolling_release	No	Object	Rolling deployment parameter, which is mandatory when type is set to RollingRelease . For details, see Table 5-162 .
gray_release	No	Object	Dark launch upgrade parameter, which is mandatory when type is set to GrayRelease . For details, see Table 5-163 .

Table 5-162 rolling_release parameter

Parameter	Mandatory	Type	Description
batches	Yes	Integer	Deployment batches.

Table 5-163 gray_release parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Dark launch policy. weight : The upgrade is performed based on dark launch traffic.
first_batch_weight	Yes	Integer	Proportion of first-batch dark launch traffic.
first_batch_replica	Yes	Integer	Number of first-batch dark launch instances.
remaining_batch	Yes	Integer	Number of batches for remaining instance deployment. After the first batch is completed, the remaining instances will be deployed in specified batches. Example: If there are 5 instances remaining to be deployed in 3 batches, these 5 instances will be upgraded based on 2:2:1.

Table 5-164 Command parameters

Parameter	Mandatory	Type	Description
command	No	Array of String	Command for controlling container running.
args	No	Array of String	Parameter for controlling container running. Example: -port=8080. If there are multiple parameters, separate them by line breaks.

Table 5-165 component_lifecycle parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Processing method. <ul style="list-style-type: none"> • http • command
scheme	No	String	HTTP request type. Value: HTTP or HTTPS. This parameter is valid only when type is set to http .
host	No	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	No	Integer	Port. This parameter is valid only when type is set to http .
path	No	String	Request path. This parameter is valid only when type is set to http .
command	No	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-166 mesher parameter

Parameter	Mandatory	Type	Description
port	Yes	Integer	Process listening port.

Table 5-167 tomcat_opt parameter

Parameter	Mandatory	Type	Description
server_xml	Yes	String	<ol style="list-style-type: none">1. Listening port in the server.xml file will take effect even if it has been configured in public network access.2. The custom Tomcat configuration may conflict with the existing port. Check whether the custom port is occupied.

Table 5-168 host_aliases parameters

Parameter	Mandatory	Type	Description
ip	Yes	String	IP address.
hostname	Yes	Array of String	Host alias.

Table 5-169 dns_config parameters

Parameter	Mandatory	Type	Description
nameservers	No	Array of String	IP address list of the DNS server of the pod.
searches	No	Array of String	(Optional) List of DNS search domains used to search for host names in the pod.
options	No	Array of objects	An optional list of objects where each object may have a name property (required) and a value property (optional). This property is merged into the options generated from the specified DNS policy. Duplicate entries will be deleted. See Table 5-170 .

Table 5-170 options parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Operation name.
value	No	String	Operation value.

Table 5-171 security_context parameters

Parameter	Mandatory	Type	Description
run_as_user	No	Integer	User for running containers. For example, to run containers as user root , set the user ID to 0 .
run_as_group	No	Integer	Owner group specified for running container processes.
capabilities	No	Object	Capability set. See Table 5-172 .

Table 5-172 capabilities parameters

Parameter	Mandatory	Type	Description
add	No	Array of String	Grant the Linux permission to the running user.
drop	No	Array of String	Remove the Linux permission of the running user.

Table 5-173 logs parameters

Parameter	Mandatory	Type	Description
log_path	Yes	String	Container log path.
rotate	Yes	String	Interval for dumping logs.
host_path	Yes	String	Mounted host path.

Parameter	Mandatory	Type	Description
host_extend_path	Yes	String	<p>Host extension path. By extending the host path, you can distinguish mounting from different containers in the same host path.</p> <ul style="list-style-type: none"> • None: the extended path is not used. • PodUID: extend the host path based on the pod ID. • PodName: extend the host path based on the pod name. • PodUID/ContainerName: extend the host path based on the pod ID and container name. • PodName/ContainerName: extend the host path based on the pod name and container name.

Table 5-174 custom_metric parameters

Parameter	Mandatory	Type	Description
path	Yes	String	Collection path. Example: /metrics.
port	Yes	Integer	Collection port. Example: 9090.
dimensions	Yes	String	Monitoring dimension. Example: cpu_usage,mem_usage.

Table 5-175 component_affinity parameters

Parameter	Mandatory	Type	Description
az	No	Array of String	List of AZs.
node	No	Array of String	Node list.
component	No	Array of objects	Component list. See Table 5-176 .

Table 5-176 affinity_component parameters

Parameter	Mandatory	Type	Description
displayName	No	String	Workload name.
name	No	String	Component name.

Table 5-177 match_expressions parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Condition key.
value	Yes	String	Condition value.
operation	Yes	String	Operation

Table 5-178 component_probe parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Type. Value: http, tcp, or command.
delay	Yes	Integer	Interval between the startup and detection.
timeout	Yes	Integer	Detection timeout interval.
scheme	No	String	Request type, which is HTTP or HTTPS . This parameter is valid only when type is set to http .
host	No	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	No	Integer	Port number. This parameter is valid only when type is set to http or tcp .
path	No	String	Request path. This parameter is valid only when type is set to http .
command	No	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-179 refer_resources parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Resource ID.
type	Yes	String	Resource type. <ul style="list-style-type: none"> • vpc • eip • elb • cce • ecs • as • cse • dcs • rds
parameters	No	Object	Resource parameter, which is mandatory when type is cce . See Table 5-180 .

Table 5-180 refer_resource_parameter parameter

Parameter	Mandatory	Type	Description
namespace	No	String	Namespace.

Table 5-181 env parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Variable name.
value	No	String	Variable value.
value_from	No	Object	Variable reference. See Table 5-182 .

Table 5-182 value_from parameters

Parameter	Mandatory	Type	Description
reference_type	Yes	String	Reference type. <ul style="list-style-type: none">• configMapKey• secretKey
name	Yes	String	Name of the configuration item or secret.
key	No	String	Key of the configuration item or secret.
optional	No	Boolean	Whether the key of the configuration item or secret is mandatory. Value: true and false.

Table 5-183 runtime_stack parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Technology stack name.
type	Yes	String	Technology stack type. Value: Java, Tomcat, Nodejs, Php, Docker, or Python. When deploy_mode is set to virtualmachine , only Java, Tomcat, and Node.js are supported. When deploy_mode is set to container , all of the preceding types are supported.
version	Yes	String	Technology stack version.
deploy_mode	Yes	String	Deployment mode. <ul style="list-style-type: none">• container• virtualmachine

Table 5-184 source parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	Component source kind. <ul style="list-style-type: none">• code• package• image
version	No	String	Version number.
url	No	String	Package address. This parameter is mandatory when kind is set to package or image .

Parameter	Mandatory	Type	Description
storage	No	String	Storage mode. Value: swr, codearts, or obs. This parameter is mandatory when kind is set to package or image .
auth	No	String	Authentication mode. Value: iam or none. Default value: iam.
repo_auth	No	String	Authorization name, which can be obtained from the authorization list.
repo_namespace	No	String	Namespace of the code repository.
repo_ref	No	String	Code branch or tag. Default value: master.
repo_type	No	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
web_url	No	String	Code redirection URL. Example: https://github.com/example/demo.git.
repo_url	No	String	Code repository URL. Example: https://github.com/example/demo.git.

Table 5-185 build parameter

Parameter	Mandatory	Type	Description
parameters	No	Map<String, Object>	This parameter is provided only when no ID is available during build creation. See Table 5-186 .

Table 5-186 parameters

Parameter	Mandatory	Type	Description
build_cmd	No	String	Compilation command. By default: <ol style="list-style-type: none">When build.sh exists in the root directory, the command is ./build.sh.When build.sh does not exist in the root directory, the command varies depending on the OS. Example:<ul style="list-style-type: none">Java and Tomcat: mvn clean packageNodejs: npm build

Parameter	Mandatory	Type	Description
dockerfile_path	No	String	Address of the Docker file. By default, the Docker file is in the root directory (./).
artifact_namespace	No	String	Build archive organization. Default value: cas_{project_id}.
cluster_id	Yes	String	ID of the cluster to be built.
node_label_selector	No	Map<String, String>	key : key of the tag. value : value of the tag.
environment_id	No	String	Environment ID.

Table 5-187 external_accesses parameters

Parameter	Mandatory	Type	Description
protocol	No	String	External access type. Value: http or https.
address	No	String	External access address.
forward_port	No	Integer	External access port.

Response

Table 5-188 Response parameters

Parameter	Type	Description
job_id	String	Task ID.

Example Request

Modify the component named **mycomponet**. The component source is the **weather-1.0.0.jar** package stored in OBS **bucket0001**.

```
{
  "name": "mycomponet",
  "description": "",
  "labels": [
    {
      "key": "com-key",
      "value": "com-value"
    }
  ],
}
```

```
"version": "2023.0323.15181",
"runtime_stack": {
  "name": "OpenJDK8",
  "version": "1.1.1",
  "type": "Java",
  "deploy_mode": "virtualmachine"
},
"source": {
  "kind": "package",
  "url": "obs://bucket0001/weather-1.0.0.jar",
  "version": "",
  "storage": "obs"
},
"tomcat_opts": {
  "server_xml": ""
},
"refer_resources": [
  {
    "id": "67835bb3-1235-4cc9-be71-becbb2b4ca0d",
    "type": "ecs"
  }
],
"replica": 1,
"external_accesses": []
}
```

Example Response

```
{
  "job_id": "JOB1c8e20ec-1b30-4ee4-9a36-35a18b5e8b7e"
}
```

Status Code

Table 5-189 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.5 Deleting a Component Based on the Component ID

Function

This API is used to delete a component based on the component ID.

URI

DELETE /v3/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 5-190 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 5-191 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 5-192 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
409	Conflict
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.6 Obtaining Component Information Based on the Component ID

Function

This API is used to obtain component information based on the component ID.

URI

GET /v3/{project_id}/cas/applications/{application_id}/components/{component_id}

Table 5-193 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 5-194 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-195 Response parameters

Parameter	Type	Description
name	String	Application component name. The value contains 2 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It starts with a letter and ends with a letter or digit. Letters are case insensitive.
labels	Array of objects	Label. For container-based deployment, you can query a workload by label. For VM-based deployment, you can configure a label of the corresponding microservice name to use graceful startup and shutdown. See Table 5-196 .
runtime_stack	Object	Runtime. See Table 5-222 . The value can be obtained from the response of the API for Querying a Runtime System Stack .
environment_id	String	Environment ID.
application_id	String	Application ID.
description	String	Description. The value can contain up to 128 characters.
source	Object	Source of the code or software package. See Table 5-223 .
build	Object	Component build. See Table 5-224 .
limit_cpu	Number	Maximum CPU limit. Unit: core.

Parameter	Type	Description
limit_memory	Number	Maximum memory size. Unit: GiB.
request_cpu	Number	Applied CPU resources. Unit: core.
request_memory	Number	Applied memory. Unit: GiB.
version	String	Component version number. The value can contain up to 32 characters. It needs to meet the following rule: $^([0-9]+)(.[0-9]+){2,3}$$
envs	Array of objects	Environment variable list. See Table 5-220 .
replica	Integer	Number of instances.
storages	Array of objects	Storage. See Table 5-197 .
deploy_strategy	Object	Component deployment. See Table 5-200 .
command	Object	Startup command. See Table 5-203 .
post_start	Object	Post-start processing. See Table 5-204 .
pre_stop	Object	Pre-stop processing. See Table 5-204 .
mesher	Object	Multi-language access service mesh. This parameter is mandatory when Node.js is selected for the technology stack and the CSE engine is bound. See Table 5-205 .
timezone	String	Time zone where the component runs. Example: Asia/Shanghai.
jvm_opts	String	JVM parameter. Example: -Xms256m -Xmx1024m. If there are multiple parameters, separate them by spaces. If the parameter is left blank, the default value is used.
tomcat_opts	Object	Tomcat parameter. This parameter is valid when you use the Tomcat technology stack. See Table 5-206 .
logs	Array of objects	Log collection. See Table 5-212 .
custom_metrics	Object	Custom metrics. See Table 5-213 .
affinity	Object	Affinity. See Table 5-214 .
anti_affinity	Object	Anti-affinity. See Table 5-214 .

Parameter	Type	Description
liveness_probe	Object	Component liveness probe. See Table 5-217 .
readiness_probe	Object	Component service probe. See Table 5-217 .
refer_resources	Array of objects	Associated resource. See Table 5-218 .
status	Object	Component status. See Table 5-226 .

Table 5-196 labels parameters

Parameter	Type	Description
key	String	Label name.
value	String	Label value.

Table 5-197 storages parameters

Parameter	Type	Description
type	String	Data storage type. <ul style="list-style-type: none">• HostPath: host path for local disk mounting.• EmptyDir: temporary directory for local disk mounting.• ConfigMap: configuration item for local disk mounting.• Secret: secrets for local disk mounting.• PersistentVolumeClaim: cloud storage mounting.
name	String	Name of the storage disk.
parameters	Object	For details about the parameters of each data storage type, see Table 5-198 .
mounts	Array of objects	Mount path of data storage. See Table 5-199 .

Table 5-198 component_storage_parameters parameters

Parameter	Type	Description
path	String	Host path. This parameter is applicable to the HostPath storage type.
name	String	Name of a configuration item, secret, or PVC, which is applicable to ConfigMap, Secret, or PersistentVolumeClaim storage, respectively.
default_mode	Integer	Mounting permission. The value is in decimal format. Example: 384. This parameter is applicable to ConfigMap and secret storage.
medium	String	This parameter is applicable to the EmptyDir storage type. Example: memory.

Table 5-199 mounts parameters

Parameter	Type	Description
path	String	Mount path.
sub_path	String	Subpath of the mount path.
read_only	Boolean	Read only or not.

Table 5-200 deploy_strategy parameters

Parameter	Type	Description
type	String	Deployment type. <ul style="list-style-type: none">• OneBatchRelease: single-batch upgrade.• RollingRelease: rolling deployment and upgrade.• GrayRelease: dark launch upgrade.
rolling_release	Object	Rolling deployment parameter. See Table 5-201 .
gray_release	Object	Dark launch upgrade. See Table 5-202 .

Table 5-201 rolling_release parameter

Parameter	Type	Description
batches	Integer	Deployment batches.

Table 5-202 gray_release parameters

Parameter	Type	Description
type	String	Dark launch policy. weight : The upgrade is performed based on dark launch traffic.
first_batch_weight	Integer	Proportion of first-batch dark launch traffic.
first_batch_replica	Integer	Number of first-batch dark launch instances.
remaining_batches	Integer	Number of batches for remaining instance deployment. After the first batch is completed, the remaining instances will be deployed in specified batches. Example: If there are 5 instances remaining to be deployed in 3 batches, these 5 instances will be upgraded based on 2:2:1.

Table 5-203 Command parameters

Parameter	Type	Description
command	Array of String	Command for controlling container running.
args	Array of String	Parameter for controlling container running. Example: -port=8080. If there are multiple parameters, separate them by line breaks.

Table 5-204 component_lifecycle parameters

Parameter	Type	Description
type	String	Processing method. <ul style="list-style-type: none">• http• command
scheme	String	HTTP request type, which can be HTTP or HTTPS. This parameter is valid only when type is set to http .
host	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	Integer	Port number. This parameter is valid only when type is set to http .

Parameter	Type	Description
path	String	Request path. This parameter is valid only when type is set to http .
command	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-205 mesher parameter

Parameter	Type	Description
port	Integer	Process listening port.

Table 5-206 tomcat_opt parameter

Parameter	Type	Description
server_xml	String	<ol style="list-style-type: none">1. Listening port in the server.xml file will take effect even if it has been configured in public network access.2. The custom Tomcat configuration may conflict with the existing port. Check whether the custom port is occupied.

Table 5-207 host_aliases parameters

Parameter	Type	Description
ip	String	IP address.
hostname	Array of String	Host alias.

Table 5-208 dns_config parameters

Parameter	Type	Description
nameservers	Array of String	IP address list of the DNS server of the pod.
searches	Array of String	List of DNS search domains used to search for host names in the pod.

Parameter	Type	Description
options	Array of objects	An optional list of objects where each object may have a name property (required) and a value property (optional). This property is merged into the options generated from the specified DNS policy. Duplicate entries will be deleted. See Table 5-209 .

Table 5-209 options parameters

Parameter	Type	Description
name	String	Operation name.
value	String	Operation value.

Table 5-210 security_context parameters

Parameter	Type	Description
run_as_user	Integer	User for running containers. Example: To run containers as user root , set the user ID to 0 .
run_as_group	Integer	Owner group specified for running container processes.
capabilities	Object	Capability set. See Table 5-211 .

Table 5-211 capabilities parameters

Parameter	Type	Description
add	Array of String	Grant the Linux permission to the running user.
drop	Array of String	Remove the Linux permission of the running user.

Table 5-212 logs parameters

Parameter	Type	Description
log_path	String	Container log path.
rotate	String	Interval for dumping logs.
host_path	String	Mounted host path.

Parameter	Type	Description
host_extend_path	String	Host extension path. By extending the host path, you can distinguish mounting from different containers in the same host path. <ul style="list-style-type: none">• None: the extended path is not used.• PodUID: extend the host path based on the pod ID.• PodName: extend the host path based on the pod name.• PodUID/ContainerName: extend the host path based on the pod ID and container name.• PodName/ContainerName: extend the host path based on the pod name and container name.

Table 5-213 custom_metric parameters

Parameter	Type	Description
path	String	Collection path. Example: /metrics.
port	Integer	Collection port. Example: 9090.
dimensions	String	Monitoring dimension. Example: cpu_usage or mem_usage.

Table 5-214 component_affinity parameters

Parameter	Type	Description
az	Array of String	List of AZs.
node	Array of String	Node list.
component	Array of objects	Component list. See Table 5-215 .

Table 5-215 affinity_component parameters

Parameter	Type	Description
displayName	String	Workload name.
name	String	Component name.

Table 5-216 match_expressions parameters

Parameter	Type	Description
key	String	Condition key.
value	String	Condition value.
operation	String	Operation.

Table 5-217 component_probe parameters

Parameter	Type	Description
type	String	Type. Value: http, tcp, or command.
delay	Integer	Interval between the startup and detection.
timeout	Integer	Detection timeout interval.
scheme	String	Request type, which is HTTP or HTTPS . This parameter is valid only when type is set to http .
host	String	Custom IP address. If this parameter is left blank, the pod IP address is used. This parameter is valid only when type is set to http .
port	Integer	Port number. This parameter is valid only when type is set to http or tcp .
path	String	Request path. This parameter is valid only when type is set to http .
command	Array of String	Command list. This parameter is valid only when type is set to command .

Table 5-218 refer_resources parameters

Parameter	Type	Description
id	String	Resource ID.

Parameter	Type	Description
type	String	Resource type. <ul style="list-style-type: none">• vpc• eip• elb• cce• ecs• as• cse• dcs• rds
parameters	Object	Resource parameters. See Table 5-219 .

Table 5-219 refer_resource_parameter parameter

Parameter	Type	Description
namespace	String	Namespace.

Table 5-220 env parameters

Parameter	Type	Description
name	String	Variable name.
value	String	Variable value.
value_from	Object	Variable reference. See Table 5-221 .

Table 5-221 value_from parameters

Parameter	Type	Description
reference_type	String	Reference type. <ul style="list-style-type: none">• configMapKey• secretKey
name	String	Name of the configuration item or secret.
key	String	Key of the configuration item or secret.
optional	Boolean	Whether the key of the configuration item or secret is mandatory.

Table 5-222 runtime_stack parameters

Parameter	Type	Description
name	String	Technology stack name.
type	String	Technology stack type. Value: Java, Tomcat, Nodejs, Php, Docker, or Python. When deploy_mode is set to virtualmachine , only Java, Tomcat, and Node.js are supported. When deploy_mode is set to container , all of the preceding types are supported.
version	String	Technology stack version.
deploy_mode	String	Deployment mode. <ul style="list-style-type: none">• container• virtualmachine

Table 5-223 source parameters

Parameter	Type	Description
kind	String	Component source kind. <ul style="list-style-type: none">• code• package• image
version	String	Version number.
url	String	Package address. This parameter is mandatory when kind is set to package or image .
storage	String	Storage mode. Value: sw, codearts, or obs. This parameter is mandatory when kind is set to package or image .
auth	String	Authentication mode. Value: iam or none. Default value: iam.
repo_auth	String	Authorization name, which can be obtained from the authorization list.
repo_namespace	String	Namespace of the code repository.
repo_ref	String	Code branch or tag. Default value: master.
repo_type	String	Code repository type. Value: GitHub, BitBucket, or GitLab.
web_url	String	Code redirection URL. Example: https://github.com/example/demo.git.

Parameter	Type	Description
repo_url	String	Code repository URL. Example: https://github.com/example/demo.git.

Table 5-224 build parameter

Parameter	Type	Description
parameters	Map<String, Object>	This parameter is provided only when no ID is available during build creation. See Table 5-225 .

Table 5-225 parameters

Parameter	Type	Description
build_cmd	String	Compilation command. By default: <ol style="list-style-type: none">When build.sh exists in the root directory, the command is ./build.sh.When build.sh does not exist in the root directory, the command varies depending on the OS. Example:<ul style="list-style-type: none">Java and Tomcat: mvn clean packageNodejs: npm build
dockerfile_path	String	Address of the Docker file. By default, the Docker file is in the root directory (./).
artifact_name_space	String	Build archive organization. Default value: cas_{project_id} .
cluster_id	String	ID of the cluster to be built.
node_label_selector	Map<String, String>	key : key of the tag. value : value of the tag.
environment_id	String	Environment ID.

Table 5-226 status parameters

Parameter	Type	Description
component_status	String	Component status. <ul style="list-style-type: none">INITIALIZINGUPGRADINGFAILEDRUNNINGDOWNDELETINGDELETEDRESERVEDSTARTINGSTOPPINGSTOPPEDRESTARTINGPENDINGUNKNOWNPARTIALLY_FAILED
available_replica	Integer	Available instances.
replica	Integer	Total instances.
fail_detail	String	Failure cause. <ul style="list-style-type: none">cluster_deletedcluster_unavailablecluster_inaccessiblenamespace_deletednamespace_unavailablenamespace_inaccessibleresource_deletedcreate_faileddelete_failed
last_job_id	String	ID of the last executed task.
create_time	Integer	Creation time.
update_time	Integer	Update time.
creator	String	Creator.
artifact	Object	Artifact information. See Table 5-227 .

Table 5-227 artifact parameters

Parameter	Type	Description
type	String	Component type. package : VM-based deployment. image : other deployment modes.
url	String	Software package or image address.

Example Request

None

Example Response

```
{
  "name": "test-component",
  "description": "",
  "labels": [],
  "runtime_stack": {
    "type": "Java",
    "name": "OpenJDK8",
    "deploy_mode": "virtualmachine",
    "version": "1.1.1"
  },
  "id": "b8702b0f-94d3-4822-98a1-56815632a0a0",
  "source": {
    "kind": "package",
    "url": "obs://bucket0001/weather-1.0.0.jar",
    "storage": "obs"
  },
  "environment_id": "4d084044-0b80-4641-963c-b9c9f4092a4f",
  "application_id": "fc092465-a5fb-4a52-bc65-b735f18366d8",
  "replica": 1,
  "version": "2023.0323.15181",
  "envs": [],
  "tomcat_opts": {
    "server_xml": "",
    "http_port": 0,
    "context_path": null
  },
  "refer_resources": [
    {
      "id": "Default",
      "type": "ecs",
      "parameters": {
        "hosts": [
          "67835bb3-1235-4cc9-be71-becbb2b4ca0d"
        ]
      }
    }
  ],
  "status": {
    "component_status": "RUNNING",
    "available_replica": 1,
    "replica": 1,
    "fail_detail": null,
    "last_job_id": "JOB1c8e20ec-1b30-4ee4-9a36-35a18b5e8b7e",
    "creator": null,
    "create_time": 1679556221933,
    "update_time": 1679579448500,
  },
  "workload_kind": "deployment"
}
```

Status Code

Table 5-228 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID]. Example: SVCSTG.00100400.
For details, see [ServiceStage Error Codes](#).

5.3.7 Delivering Component Tasks by Component ID

Function

This API is used to deliver component tasks by component ID.

URI

POST /v3/{project_id}/cas/applications/{application_id}/components/
{component_id}/action

Table 5-229 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Request

Table 5-230 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 5-231 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Component task type. <ul style="list-style-type: none">• start• stop• restart• scale• rollback• continue_deploy• check_gray_release• modify_gray_rule
parameters	No	Object	Task parameters. See Table 5-232 .

Table 5-232 parameters

Parameter	Mandatory	Type	Description
replica	No	Integer	Number of component instances.
version	No	String	Version number.
hosts	No	Array of String	Host ID list, which is mandatory when component scaling is deployed on a VM.

Response

Table 5-233 Response parameters

Parameter	Type	Description
job_id	String	Task ID.
result	String	Task status.

Example Request

```
{
  "action": "start",
  "parameters": {
    "replica": 0,
    "hosts": [
      "string"
    ],
    "version": "string"
  }
}
```

Example Response

```
{
  "job_id": "string",
  "result": "succeeded"
}
```

Status Code

Table 5-234 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.3.8 Obtaining Records by Component ID

Function

This API is used to obtain records by component ID.

URI

GET /v3/{project_id}/cas/applications/{application_id}/components/{component_id}/records

Table 5-235 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
application_id	Yes	String	Application ID. See Obtaining All Applications .
component_id	Yes	String	Component ID. See Obtaining All Components of an Application .

Table 5-236 Query parameter

Parameter	Mandatory	Type	Description
limit	No	String	Number of records to be queried. Value range: 0–100.
offset	No	String	Offset.
order_by	No	String	Sorting field. By default, query results are sorted by creation time. Enumerated values: create_time, name, and update_time.
order	No	String	Sorting order. <ul style="list-style-type: none">• desc (default)• asc

Request

Table 5-237 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-238 Response parameters

Parameter	Type	Description
count	Integer	Total number of records.
records	Array of objects	Record list. See Table 5-239 .

Table 5-239 records parameters

Parameter	Type	Description
begin_time	String	Start time.
end_time	String	End time.
description	Object	Description.
instance_id	String	Instance ID.
version	String	Version number.
current_used	Boolean	Currently used parameter.
status	String	Status.
deploy_type	String	Deployment type. <ul style="list-style-type: none">• OneBatchRelease• RollingRelease• GrayRelease

Parameter	Type	Description
jobs	Array of objects	Task list. See Table 5-240 .

Table 5-240 jobs parameters

Parameter	Type	Description
sequence	Integer	Execution sequence.
deploy_type	String	Deployment type. <ul style="list-style-type: none"> OneBatchRelease RollingRelease GrayRelease
job_id	String	Task ID.
job_info	Object	Task information. See Table 5-241 .

Table 5-241 job_info parameters

Parameter	Type	Description
deploy_type	String	Task type.
source_url	String	Component source.
first_batch_weight	Integer	First execution weight.
first_batch_replica	Integer	Component instances deployed first time.
replica	Integer	Total instances.
remaining_batch	Integer	Remaining batch.

Example Request

None

Example Response

```
{
  "count": 10,
  "records": [
    {
      "begin_time": "string",
      "end_time": "string",
      "description": null,

```

```
"instance_id": "string",
"version": "string",
"current_used": true,
"status": "string",
"deploy_type": "string",
"jobs": [
  {
    "sequence": 0,
    "deploy_type": "string",
    "job_id": "string",
    "job_info": {
      "deploy_type": "string",
      "source_url": "string",
      "first_batch_weight": 0,
      "first_batch_replica": 0,
      "replica": 0,
      "remaining_batch": 0
    }
  }
]
}
```

Status Code

Table 5-242 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

5.4 Runtime System

5.4.1 Querying a Runtime System Stack

Function

This API is used to obtain runtime system information.

URI

GET /v3/{project_id}/cas/runtimestacks

Table 5-243 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 5-244 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 5-245 Response parameters

Parameter	Type	Description
runtimestacks	Array of objects	Runtime system stack. See Table 5-246 .

Table 5-246 runtimestacks parameters

Parameter	Type	Description
name	String	Runtime system name.
deploy_mode	String	Deployment mode. <ul style="list-style-type: none">• container• virtualmachine
version	String	Version number.
type	String	Technology stack type. Value: Nodejs, Java, Tomcat, Python, Docker, or Php.

Parameter	Type	Description
status	String	Technology stack status. <ul style="list-style-type: none"> Supported Deprecated

Example Request

None

Example Response

```
{
  "runtimestacks": [
    {
      "name": "runtimes-test",
      "deploy_mode": "container",
      "version": "string",
      "type": "Nodejs",
      "status": "Supported"
    }
  ]
}
```

Status Code

Table 5-247 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request
404	Not Found
409	Conflict
500	Internal Server Error

Error Code

The error code format is SVCSTG.00100.[Error_ID], for example, SVCSTG.00100400. For details, see [ServiceStage Error Codes](#).

6 Git Repository Access APIs

6.1 Obtaining a Git Repository Authorization List

Function

This API is used to obtain a Git repository authorization list.

URI

GET /v1/{project_id}/git/auths

[Table 6-1](#) describes the parameters.

Table 6-1 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 6-2 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 6-3 Response parameters

Parameter	Type	Description
authorizations	Array<Object>	Authorization structure. See Table 6-4 .
count	Integer	Number of authorized repositories.

Table 6-4 authorizations parameters

Parameter	Type	Description
name	String	Authorized repository.
repo_type	String	Repository type. Value: GitHub, BitBucket, or GitLab.
repo_host	String	Repository address.
repo_home	String	Repository homepage.
repo_user	String	Repository username.
avatar	String	Repository avatar.
token_type	String	Repository authorization type.
create_time	Long	Repository authorization creation timestamp.
update_time	Long	Repository authorization update timestamp.
status	Integer	Repository authorization status. 0: normal 1: overdue 2: unauthenticated
tag	String	Type of the authorized site. Default value: null.

Example Request

None

Example Response

```
{
  "authorizations": [
    {
      "name": "test",
      "repo_type": "github",
      "repo_host": "https://api.github.com",
      "repo_home": "https://github.com/tom-repo",
      "repo_user": "tom-repo",
      "avatar": "https://avatars.githubusercontent.com/u/73919200?v=4",
      "token_type": "oauth",
      "create_time": 1649731565642,
      "update_time": 1649731565642,
      "tag": null,
      "status": 0
    }
  ],
  "count": 1
}
```

Status Code

Table 6-5 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.2 Obtaining an Authorization Redirection URL

Function

This API is used to obtain an authorization redirection URL.

URI

GET /v1/{project_id}/git/auths/{repo_type}/redirect

[Table 6-6](#) describes the parameters.

Table 6-6 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
repo_type	Yes	String	Repository type. Value: GitHub, BitBucket, or GitLab.

Table 6-7 Query parameters

Parameter	Mandatory	Type	Description
tag	No	String	Site label. Example: ?tag=intl, which indicates an international site.

Request

Table 6-8 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

Table 6-9 Response parameters

Parameter	Type	Description
url	String	Authorization redirection URL.

Example Request

None

Example Response

```
{
  "url": "https://github.com/login/oauth/authorize?client_id=2993bccf255673ba****&redirect_uri=https%3A%2F%2Fconsole.huaweicloud.com%2Fservicestage%2Foauth%3Frepo_type%3Dgithub&state=%242a%2410%24%2FMOymg0euuGH%2FJNWJI3FeSVZhTUVj%2FF7r52mqpUKcYXz7f6.****&scope=read%3Auser%20repo%20write%3Arepo_hook"
}
```

Status Code

Table 6-10 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.3 Creating OAuth Authorization

Function

This API is used to create OAuth authorization.

URI

POST /v1/{project_id}/git/auths/{repo_type}/oauth

[Table 6-11](#) describes the parameters.

Table 6-11 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
repo_type	Yes	String	Repository type. Value: GitHub, BitBucket, or GitLab.

Table 6-12 Query parameters

Parameter	Mandatory	Type	Description
tag	No	String	Site label. Example: ?tag=intl, which indicates an international site.

Request

Table 6-13 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 6-14 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Authorization name.
code	Yes	String	code parameter in the redirection URL after Git repository authorization is performed.
state	Yes	String	One-off authentication code and random string received after Git repository authorization is performed.

 NOTE

To obtain the values of **code** and **state**, perform the following steps:

1. Enter the URL obtained in [Obtaining an Authorization Redirection URL](#) in the address box of the browser and press **Enter**.
2. On the login page, enter your code repository account and password.
If you have logged in to the code repository, skip this step.
3. On the login page of the Huawei Cloud console, enter the account name and password, and click **Log In**.
If you have logged in to the Huawei Cloud console, skip this step.
4. After the system displays a message indicating that the authorization fails to be created, copy the URL in the address box.

For example:

```
https://console.huaweicloud.com/servicestage/oauth?  
repo_type=github&code=04bed2df009f3fc9***c&state=%242a  
%2410%24.uQYThj5yKFs51w9s9ajeuQ0NSGshPH7IUfXoZLPlz7J%2FoAJ8B***
```

5. Decode the obtained URL to obtain the decoded URL.

For example:

```
https://console.huaweicloud.com/servicestage/oauth?  
repo_type=github&code=04bed2df009f3fc9***c&state=$2a  
$10$.uQYThj5yKFs51w9s9ajeuQ0NSGshPH7IUfXoZLPlz7J/oAJ8B***
```

Obtain the values of **code** and **state**.

The value of the obtained **state** is a one-off authentication code and random string, which can be used only once. If you need to call this API again to create OAuth authorization, follow the steps in [Obtaining an Authorization Redirection URL](#) again to obtain the URL and obtain the values of **code** and **state**.

Response

Table 6-15 Response parameters

Parameter	Type	Description
authorization	Object	Authorization structure. See Table 6-16 .

Table 6-16 authorization parameters

Parameter	Type	Description
name	String	Authorization name.
repo_type	String	Repository type. Value: github, gitlab, , or bitbucket.
repo_host	String	Repository address.
repo_home	String	Repository homepage.
repo_user	String	Repository username.
avartar	String	Repository avatar.

Parameter	Type	Description
token_type	String	Repository authorization type.
create_time	Long	Repository authorization creation timestamp.
update_time	Long	Repository authorization update timestamp.
status	Integer	Repository authorization status. 0: normal 1: overdue 2: unauthenticated

Example Request

Create OAuth authorization **auth1**.

```
{
  "name": "auth1",
  "code": "04bed2df009f3fc9***c",
  "state": "$2a$10$.uQYThj5yKFs51w9s9ajeuQ0NSGshPH7IUfXoZLPlz7J/oAJ8B****"
}
```

Example Response

```
{
  "authorization": {
    "name": "auth1",
    "repo_type": "github",
    "repo_host": "https://api.github.com",
    "repo_home": "https://github.com/tom-repo",
    "repo_user": "repo",
    "avatar": "https://avatars.githubusercontent.com/u/73919200?v=4",
    "token_type": "oauth",
    "create_time": 1649758626858,
    "update_time": 1649758626858,
    "status": 0
  }
}
```

Status Code

Table 6-17 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.4 Creating Private Token Authorization

Function

This API is used to create private token authorization.

URI

POST /v1/{project_id}/git/auths/{repo_type}/personal

[Table 6-18](#) describes the parameters.

Table 6-18 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
repo_type	Yes	String	Repository type. Value: github, or gitlab.

Request

Table 6-19 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 6-20 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Authorization name.
token	Yes	String	Repository token.
host	No	String	Repository address.

Response

Table 6-21 Response parameters

Parameter	Type	Description
authorization	Object	Authorization structure. See Table 6-22 .

Table 6-22 authorization parameters

Parameter	Type	Description
name	String	Authorization name.
repo_type	String	Repository type. Value: github, or gitlab.
repo_host	String	Repository address.
repo_home	String	Repository homepage.
repo_user	String	Repository username.
avatar	String	Repository avatar.
token_type	String	Repository authorization type.
create_time	Long	Repository authorization creation timestamp.
update_time	Long	Repository authorization update timestamp.
status	Integer	Repository authorization status. 0: normal 1: overdue 2: unauthenticated

Example Request

Create private token authorization **token1**.

```
{
  "name": "token1",
  "token": "ghp_7QoYcNLFahSXSbhhuT8R5xHLKe*****6",
  "host": "https://github.com/tom-repo"
}
```

Example Response

```
{
  "authorization": {
    "name": "token1",
    "repo_type": "github",
    "repo_host": "https://api.github.com",
    "repo_home": "https://github.com/tom-repo",
    "repo_user": "tom-repo",
    "avatar": "https://avatars.githubusercontent.com/u/73919264?v=4",
    "token_type": "personal",
    "create_time": 1649762172340,
    "update_time": 1649762172340,
    "status": 0
  }
}
```

Status Code

Table 6-23 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.5 Creating Password Authorization

Function

This API is used to create password authorization for a Git repository.

URI

POST /v1/{project_id}/git/auths/{repo_type}/password

[Table 6-24](#) describes the parameters.

Table 6-24 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
repo_type	Yes	String	Repository type. Value: bitbucket.

Request

Table 6-25 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Table 6-26 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Authorization name.
user	Yes	String	Repository username.
password	Yes	String	Repository password.

Response

Table 6-27 Response parameters

Parameter	Type	Description
authorization	Object	Authorization structure. See Table 6-28 .

Table 6-28 authorization parameters

Parameter	Type	Description
name	String	Authorization name.
repo_type	String	Repository type. Value: bitbucket.
repo_host	String	Repository address.
repo_home	String	Repository homepage.
repo_user	String	Repository username.
avatar	String	Avatar.
token_type	String	Authorization mode.
create_time	Long	Creation time.
update_time	Long	Update time.
status	Integer	Repository authorization status. 0: normal 1: overdue 2: unauthenticated

Example Request

Create password authorization **pass1**.

```
{  
  "name": "pass1",  
  "user": "bucket-ljn",  
  "password": "*****"  
}
```

Example Response

```
{  
  "authorization": {  
    "name": "pass1",  
    "repo_type": "bitbucket",  
    "repo_host": "https://api.bitbucket.org",  
    "repo_home": "https://bitbucket.org/%7B63cc78de-*****-9505c0335f96%7D/",  
    "repo_user": "bucket-ljn",  
    "avatar": "https://secure.gravatar.com/avatar/05a7b4a2302750*****cf64?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FB-2.png",  
    "token_type": "password",  
    "create_time": 1634119075119,  
    "update_time": 1634119075119,  
    "status": 0  
  }  
}
```

Status Code

Table 6-29 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.6 Deleting Repository Authorization

Function

This API is used to delete repository authorization based on the name.

URI

DELETE /v1/{project_id}/git/auths/{name}

[Table 6-30](#) describes the parameters.

Table 6-30 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
name	Yes	String	Authorization name.

Request

Table 6-31 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Response

None

Example Request

None

Example Response

None

Status Code

Table 6-32 Status codes

HTTP Status Code	Description
204	OK
500	Internal Server Error

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.7 Obtaining a Repository Namespace

Function

This API is used to obtain a repository namespace.

URI

GET /v1/{project_id}/git/repos/namespaces

[Table 6-33](#) describes the parameters.

Table 6-33 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Request

Table 6-34 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-35 Response parameters

Parameter	Type	Description
namespaces	Array<Object>	Namespace structure. See Table 6-36 .

Table 6-36 namespaces parameters

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

Example Request

None

Example Response

```
{
  "namespaces": [
    {
      "id": "zmg",
      "name": "zmg"
    }
  ]
}
```

Status Code

Table 6-37 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.8 Obtaining Repository Information Based on the Clone URL

Function

This API is used to obtain repository information based on the clone URL.

URI

GET /v1/{project_id}/git/repos/project-info

[Table 6-38](#) describes the parameters.

Table 6-38 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .

Table 6-39 Query parameters

Parameter	Mandatory	Type	Description
clone_url	Yes	String	Clone URL of a repository. See Obtaining All Projects in a Namespace .

Request

Table 6-40 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-41 Response parameters

Parameter	Type	Description
namespace_id	String	Namespace ID.
namespace	String	Namespace.
project_id	String	Repository project ID.
project	String	Repository project.

Example Request

Obtain repository information based on clone URL **http://xxx.gitlab.com:8090/xxx/springcloud-xxbbon.git**.

```
https://Endpoint/v1/7c6a62ddb2bf45a3920d8053d3cb8fbf/git/repos/project-info?clone_url=http://xxx.gitlab.com:8090/xxx/springcloud-xxbbon.git
```

Example Response

```
{
  "project": "gradle_custom_gradle_dir",
  "namespace_id": "zmg",
  "namespace": "zmg",
  "project_id": "79",
}
```

Status Code

Table 6-42 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.9 Obtaining All Projects in a Namespace

Function

This API is used to obtain all projects in a namespace.

URI

GET /v1/{project_id}/git/repos/{namespace}/projects

[Table 6-43](#) describes the parameters.

Table 6-43 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID. See Obtaining a Repository Namespace .

Request

Table 6-44 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-45 Response parameters

Parameter	Type	Description
projects	Array<Object>	Project structure. See Table 6-46 .

Table 6-46 projects parameters

Parameter	Type	Description
id	String	Project ID.
name	String	Project name.
clone_url	String	Clone URL.

Example Request

None

Example Response

```
{
  "projects": [
    {
      "id": "79",
      "name": "gradle_custom_gradle_dir",
      "clone_url": "http://***@cpe.gitlab.com:8090/zmg/gradle_custom_gradle_dir"
    }
  ]
}
```



```
]
}
```

Status Code

Table 6-47 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.10 Creating a Software Repository Project

Function

This API is used to create a software repository project.

URI

POST /v1/{project_id}/git/repos/{namespace}/projects

[Table 6-48](#) describes the parameters.

Table 6-48 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID. See Obtaining a Repository Namespace .

Request

Table 6-49 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Table 6-50 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Project name.

Response

Table 6-51 Response parameters

Parameter	Type	Description
id	String	Project ID.
name	String	Project name.
clone_url	String	Clone URL.

Example Request

Create software repository project **string**.

```
{  
  "name": "string"  
}
```

Example Response

```
{  
  "id": "79",  
  "name": "gradle_custom_gradle_dir",  
}
```

```
"clone_url": "http://***@cpe.gitlab.com:8090/zmg/gradle_custom_gradledir.git"
}
```

Status Code

Table 6-52 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.11 Obtaining a Project Branch

Function

This API is used to obtain a project branch.

URI

GET /v1/{project_id}/git/repos/{namespace}/{project}/branches

[Table 6-53](#) describes the parameters.

Table 6-53 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Request

Table 6-54 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-55 Response parameters

Parameter	Type	Description
branches	Array<String>	Project branch.

Example Request

None

Example Response

```
{
  "branches": [
    "main"
  ]
}
```

Status Code

Table 6-56 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.12 Obtaining a Project Tag

Function

This API is used to obtain a project tag.

URI

GET /v1/{project_id}/git/repos/{namespace}/{project}/tags

[Table 6-57](#) describes the parameters.

Table 6-57 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Request

Table 6-58 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .

Parameter	Mandatory	Type	Description
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-59 Response parameters

Parameter	Type	Description
tags	Array<String>	Project tag.

Example Request

None

Example Response

```
{
  "tags": [
    "test1"
  ]
}
```

Status Code

Table 6-60 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.13 Creating a Project Tag

Function

This API is used to create a project tag.

URI

POST /v1/{project_id}/git/repos/{namespace}/{project}/tags

Table 6-61 describes the parameters.

Table 6-61 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Table 6-62 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.

Request

Table 6-63 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Table 6-64 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Tag name.
description	Yes	String	Description.

Response

Table 6-65 Response parameters

Parameter	Type	Description
name	String	Project tag name.

Example Request

Create project tag **test1** and set the tag description to **test version 1**.

```
{
  "name": "test1",
  "description": "test version 1"
}
```

Example Response

```
{
  "name": "test1"
}
```

Status Code

Table 6-66 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.14 Deleting a Project Tag

Function

This API is used to delete a project tag.

URI

DELETE /v1/{project_id}/git/repos/{namespace}/{project}/tags/{tag_name}

[Table 6-67](#) describes the parameters.

Table 6-67 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .
tag_name	Yes	String	Tag name. See Obtaining a Project Tag .

Request

Table 6-68 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

None

Example Request

None

Example Response

None

Status Code

Table 6-69 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.15 Obtaining Project Commits

Function

This API is used to obtain the latest ten project commits.

URI

GET /v1/{project_id}/git/repos/{namespace}/{project}/commits

[Table 6-70](#) describes the parameters.

Table 6-70 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .

Parameter	Mandatory	Type	Description
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Table 6-71 Query parameters

Parameter	Mandatory	Type	Description
ref	No	String	Branch or tag name. If it is not provided, use the default branch.

Request

Table 6-72 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-73 Response parameters

Parameter	Type	Description
commits	Array<Object>	Project commits structure. See Table 6-74 .

Table 6-74 commits parameters

Parameter	Type	Description
sha	String	Commit SHA value.
message	String	Commit description.
authored_date	String	Import time.

Example Request

None

Example Response

```
{
  "commits": [
    {
      "sha": "6dcb09b5b57875f334f61aebd695e2e4193db5e",
      "message": "Fix all the bugs",
      "authored_date": "Fix all the bugs"
    }
  ]
}
```

Status Code

Table 6-75 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.16 Obtaining a Project Hook

Function

This API is used to obtain a project hook.

URI

GET /v1/{project_id}/git/repos/{namespace}/{project}/hooks

[Table 6-76](#) describes the parameters.

Table 6-76 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Request

Table 6-77 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-78 Response parameters

Parameter	Type	Description
hooks	Array of objects	Project hooks. See Table 6-79 .

Table 6-79 hooks parameters

Parameter	Type	Description
id	String	Hook ID.
type	String	Hook type.
callback_url	String	Callback URL.

Example Request

None

Example Response

```
{
  "hooks": [
    {
      "id": "1753",
      "type": "github",
      "callback_url": "https://example.com/webhook"
    }
  ]
}
```

Status Code

Table 6-80 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.17 Creating a Project Hook

Function

This API is used to create a project hook.

URI

POST /v1/{project_id}/git/repos/{namespace}/{project}/hooks

Table 6-81 describes the parameters.

Table 6-81 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Request

Table 6-82 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Table 6-83 Request body parameters

Parameter	Mandatory	Type	Description
secret	Yes	String	Character string used to verify received payloads.
url	Yes	String	URL that is called back when a hook is triggered.

Response

Table 6-84 Response parameters

Parameter	Type	Description
id	String	Hook ID.
type	String	Hook type.
callback_url	String	Callback URL.

Example Request

Create a project hook, verify the **secrete** parameter of the received payloads string, and set the callback URL to **https://example.com/webhook** when the hook is triggered.

```
{
  "secret": "*****",
  "url": "https://example.com/webhook"
}
```

Example Response

```
{
  "id": "1573",
  "type": "github",
  "callback_url": "https://example.com/webhook"
}
```

Status Code

Table 6-85 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.18 Deleting a Project Hook

Function

This API is used to delete a project hook.

URI

DELETE /v1/{project_id}/git/repos/{namespace}/{project}/hooks/{hook_id}

[Table 6-86](#) describes the parameters.

Table 6-86 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .
hook_id	Yes	String	Hook ID. See Obtaining a Project Hook .

Request

Table 6-87 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

None

Example Request

None

Example Response

None

Status Code

Table 6-88 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.19 Obtaining a Repository File Directory

Function

This API is used to obtain a repository file directory.

URI

GET /v1/{project_id}/git/files/{namespace}/{project}/trees

[Table 6-89](#) describes the parameters.

Table 6-89 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Table 6-90 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.

Request

Table 6-91 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-92 Response parameters

Parameter	Type	Description
paths	Array<String>	File path.

Example Request

None

Example Response

```
{
  "paths": [
    "files/whitespace"
  ]
}
```

Status Code

Table 6-93 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.20 Obtaining Repository File Contents

Function

This API is used to obtain repository file contents.

URI

GET /v1/{project_id}/git/files/{namespace}/{project}/{path}

[Table 6-94](#) describes the parameters.

Table 6-94 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .
path	Yes	String	File path. If the value contains a slash (/), replace it with a colon (:). See Obtaining a Repository File Directory .

Table 6-95 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.

Request

Table 6-96 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

Table 6-97 Response parameters

Parameter	Type	Description
path	String	File path.
sha	String	Commit SHA value.
encoding	String	Encoding mode. Value: base64 or text/plain.
content	String	File contents.

Example Request

None

Example Response

```
{  
  "path": "app/models/key.rb",
```

```
"sha": "4c294617b60715c1d218e61164a3abd4808a4284cbc30e6728a01ad9aada4481",  
"encoding": "base64",  
"content": "lyA9PSBTY2hlbWEgSW5mb3..."  
}
```

Status Code

Table 6-98 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.21 Creating a Repository File

Function

This API is used to create a repository file.

URI

POST /v1/{project_id}/git/files/{namespace}/{project}/{path}

[Table 6-99](#) describes the parameters.

Table 6-99 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .

Parameter	Mandatory	Type	Description
path	Yes	String	File path. If the value contains a slash (/), replace it with a colon (:). See Obtaining a Repository File Directory .

Table 6-100 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.

Request

Table 6-101 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Table 6-102 Request body parameters

Parameter	Mandatory	Type	Description
message	Yes	String	Description.
content	Yes	String	File content after Base64 encoding.

Response

Table 6-103 Response parameters

Parameter	Type	Description
path	String	File path.

Example Request

Create a repository file, set **message** to **create a new file**, and set **content** to the content encoded using Base64.

```
{
  "message": "create a new file",
  "content": "IyA9PSBTY2h1bWEgSW5mb3..."
}
```

Example Response

```
{
  "path": "app/project.rb"
}
```

Status Code

Table 6-104 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.22 Modifying Repository File Contents

Function

This API is used to modify repository file contents.

URI

PUT /v1/{project_id}/git/files/{namespace}/{project}/{path}

[Table 6-105](#) describes the parameters.

Table 6-105 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .
path	Yes	String	File path. If the value contains a slash (/), replace it with a colon (:). See Obtaining a Repository File Directory .

Table 6-106 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.

Request

Table 6-107 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Table 6-108 Request body parameters

Parameter	Mandatory	Type	Description
message	Yes	String	Description.
content	Yes	String	File content after Base64 encoding.
sha	Yes	String	Blob SHA of a file, which can be obtained through Obtaining Repository File Contents .

Response

Table 6-109 Response parameters

Parameter	Type	Description
path	String	File path.

Example Request

Modify repository file contents.

```
{
  "message": "update file",
  "content": "IyA9PSBTY2hlcWEgSW5mb3...",
  "sha": "4c294617b60715c1d218e61164a3abd4808a4284cbc30e6728a01ad9aada4481"
}
```

Example Response

```
{
  "path": "app/project.rb"
}
```

Status Code

Table 6-110 Status codes

HTTP Status Code	Description
200	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

6.23 Deleting a Repository File

Function

This API is used to delete a repository file.

URI

DELETE /v1/{project_id}/git/files/{namespace}/{project}/{path}

[Table 6-111](#) describes the parameters.

Table 6-111 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant's project ID. See Obtaining a Project ID .
namespace	Yes	String	Namespace ID or URL code name. See Obtaining a Repository Namespace .
project	Yes	String	Repository project ID or URL code name. If the value contains a slash (/), replace it with a colon (:). See Obtaining All Projects in a Namespace .
path	Yes	String	File path. If the value contains a slash (/), replace it with a colon (:). See Obtaining a Repository File Directory .

Table 6-112 Query parameters

Parameter	Mandatory	Type	Description
ref	Yes	String	Branch name, tag name, or commit SHA value.
message	Yes	String	Commit information.
sha	Yes	String	Blob SHA of a file, which can be obtained through Obtaining Repository File Contents .

Request

Table 6-113 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type or format.
X-Auth-Token	Yes	String	API calling can be authenticated using a token or AK/SK. If you use a token, this parameter is mandatory and must be set to the token. For details about how to obtain a token, see Obtaining a User Token .
X-Repo-Auth	Yes	String	Authorization name. See Obtaining a Git Repository Authorization List .

Response

None

Example Request

None

Example Response

None

Status Code

Table 6-114 Status codes

HTTP Status Code	Description
204	OK
400	Bad Request

Error Code

The error code format is SVCSTG.REPO.[Error_ID], for example, SVCSTG.REPO.0401. For details, see [ServiceStage Error Codes](#).

7 CSE API

7.1 API Calling

CSE provides REST APIs, allowing you to call APIs using HTTPS.

To call the CSE APIs of an exclusive microservice engine, see [Calling APIs](#).

7.2 Dynamic Configuration

7.2.1 Importing Configurations

Function

This API is used to import configurations to the configuration management center.

NOTE

- This API applies to microservice engine 2.x.
- Call the API by referring to [Calling APIs](#). In [Request URI](#), replace **{Endpoint}** with [Endpoints](#).

URL

POST /v1/{project_id}/kie/file

Table 7-1 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .

Table 7-2 Query parameters

Parameter	Mandatory	Type	Description
override	Yes	String	force : Forcible import will overwrite duplicate items. skip : Duplicate items will be skipped. abort : If duplicate items are found, the import stops.
label	No	String	To import a specified label, enter a value in the format of {Label key}:{Label value}. If this parameter is empty, the label of the body is imported.

Request

Table 7-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
x-engine-id	Yes	String	ID of the engine whose configuration is to be uploaded. See Querying the Exclusive Microservice Engine List .
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Table 7-4 Request body parameter

Parameter	Mandatory	Type	Description
upload_file	Yes	File	File to be imported. The file name format is fileName.json. File format: <pre>{ "data": [{ "key": "keyItem1", "value": "valueItem1", "labels": { "1": "1" }, "status": "enabled", "value_type": "text" }] }</pre>

Response

Status code: 200

Table 7-5 Response body parameters

Parameter	Type	Description
success	List<KVDoc>	List of configuration items that are successfully imported.
failure	List<DocFailedOfUpload>	List of configuration items that fail to be imported.

Table 7-6 KVDoc

Parameter	Type	Description
id	String	Configuration ID.
key	String	Configuration key.
value	String	Configuration value.
value_type	String	Type of the configuration value.
status	String	Configuration status.
create_time	Integer	Time when the configuration is created.
update_time	Integer	Time when the configuration is updated.
create_revision	Integer	Version number of the created configuration.

Parameter	Type	Description
update_revision	Integer	Version number of the updated configuration.
labels	Map<String,String>	Configuration labels.

Table 7-7 DocFailedOfUpload

Parameter	Type	Description
key	String	Key of the configuration item that fails to be imported.
labels	Map<String,String>	Labels of the configuration item that fails to be imported.
error_code	String	Error code.
error_message	String	Error message.

Status code: 400

Table 7-8 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-9 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.
detail	String	Location details.

Example Request

Import a configuration file. If the same configurations exist, these configurations will be overwritten based on the body label.

```
POST https://{endpoint}/v1/{project_id}/kie/file/override=force
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="upload_file"; filename="custom_1663124521493.json"
Content-Type: application/json
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

Example Response

Successfully queried.

```
{
  "success": [
    {
      "id": "7e2d346e-d907-4871-bf51-a17fc7e52ad4",
      "key": "2",
      "value": "2",
      "value_type": "text",
      "create_revision": 3,
      "update_revision": 3,
      "status": "enabled",
      "create_time": 1636374809,
      "update_time": 1636374809,
      "labels": {
        "2": "2",
        "environment": "production"
      }
    },
    {
      "id": "907b3891-c691-4ae0-816f-e91eba705e28",
      "key": "1",
      "value": "1",
      "value_type": "text",
      "create_revision": 4,
      "update_revision": 4,
      "status": "enabled",
      "create_time": 1636374809,
      "update_time": 1636374809,
      "labels": {
        "1": "1",
        "environment": "production"
      }
    }
  ],
  "failure": []
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.2.2 Exporting Configurations

Function

This API is used to export configurations from the configuration management center.

NOTE

- This API applies to microservice engine 2.x.
- Call the API by referring to [Calling APIs](#). In [Request URI](#), replace **{Endpoint}** with [Endpoints](#).

URI

POST /v1/{project_id}/kie/download

Table 7-10 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .

Table 7-11 Query parameters

Parameter	Mandatory	Type	Description
label	No	String	Exports data filtered by label. The format is {Label key}: {Label value}.
match	No	String	Matching option of the filtered items. Value exact indicates exact matching, including the same number of labels. If the value is null, inclusive matching is used.

Request

Table 7-12 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
x-engine-id	Yes	String	ID of the engine whose configuration is to be downloaded. See Querying the Exclusive Microservice Engine List .
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Table 7-13 Request body parameter

Parameter	Mandatory	Type	Description
ids	Yes	List<String>	ID list of the configuration items to be exported. See Querying the Configuration List . When ids is empty, this API is called to export all configuration items.

Response

Status code: 200

Table 7-14 Response body parameter

Parameter	Mandatory	Type	Description
metadata	No	Object	Additional information about a configuration item.
data	Yes	List< KVCreateBody >	List of configuration items to be created.

Table 7-15 KVCreateBody

Parameter	Mandatory	Type	Description
id	Yes	String	Configuration ID.
key	Yes	String	Configuration key.
value	No	String	Configuration value.
labels	No	Map<String,String>	Configuration labels.
status	No	String	Configuration status.
value_type	No	String	Type of the configuration value.

Table 7-16 Metadata

Parameter	Mandatory	Type	Description
version	No	String	Version information.
annotations	No	Object	Additional information about a configuration item.

Status code: 400

Table 7-17 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-18 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Parameter	Type	Description
detail	String	Location details.

Example Request

Export the configuration whose ID is **0bdfddb8-c4f9-4a88-8360-07a8a616a804**.

```
POST https://{endpoint}/v1/{project_id}/kie/download
```

```
{
  "ids":[
    "0bdfddb8-c4f9-4a88-8360-07a8a616a804"
  ]
}
```

Example Response

Successfully queried.

```
{
  "metadata": {
    "version": "2.0.0"
  },
  "data": [
    {
      "id": "0bdfddb8-c4f9-4a88-8360-07a8a616a804",
      "key": "2",
      "labels": {
        "2": "2"
      },
      "value": "2",
      "status": "enabled",
      "value_type": "text"
    },
    {
      "id": "34befef9-7f3a-4394-85f9-142622a11b1d",
      "key": "1",
      "labels": {
        "1": "1"
      },
      "value": "1",
      "status": "enabled",
      "value_type": "text"
    }
  ]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3 Engine Management

7.3.1 Querying Flavors Supported by an Exclusive Microservice Engine

Function

This API is used to query the list of flavors supported by an exclusive microservice engine.

URI

GET /v2/{project_id}/enginemgr/flavors

Table 7-19 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .

Table 7-20 query parameter

Parameter	Mandatory	Type	Description
specType	No	String	To query a microservice engine, set it to CSE2 .

Request

Table 7-21 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Response

Status code: 200

Table 7-22 Response body parameters

Parameter	Type	Description
total	Integer	Total number of flavors of an exclusive microservice engine.
data	Array of Flavor objects	Flavor details of an exclusive microservice engine.

Table 7-23 Flavor

Parameter	Type	Description
flavor	String	Flavor of an exclusive microservice engine.
description	String	Flavor description of an exclusive microservice engine.

Status code: 400

Table 7-24 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-25 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query the list of flavors supported by an exclusive microservice engine.

```
GET /v2/{project_id}/enginemgr/flavors
```

Example Response

```
{
  "total": 0,
  "data": [
    {
      "flavor": "string",
      "description": "string"
    }
  ]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3.2 Querying the Exclusive Microservice Engine List

Function

This API is used to query the exclusive microservice engine list.

URI

```
GET /v2/{project_id}/enginemgr/engines
```

Table 7-26 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .

Table 7-27 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset.
limit	No	Integer	Number of data records on each page.
type	No	String	Engine type: CSE or CSE_SHARE.

Request

Table 7-28 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.

Response

Status code: 200

Table 7-29 Response body parameters

Parameter	Type	Description
total	Integer	Total number of exclusive microservice engines.
data	Array of EngineInfo objects	Details about exclusive microservice engines.

Table 7-30 EngineInfo

Parameter	Type	Description
id	String	ID of an exclusive microservice engine.
name	String	Name of an exclusive microservice engine.
enterpriseProjectId	String	ID of the enterprise project to which an exclusive microservice engine belongs.
enterpriseProjectName	String	Name of the enterprise project to which an exclusive microservice engine belongs.
type	String	Engine type: CSE or CSE_SHARE.

Parameter	Type	Description
description	String	Description of an exclusive microservice engine.
flavor	String	Flavor of an exclusive microservice engine.
payment	String	Billing mode of an exclusive microservice engine. 1 indicates pay-per-use, and 2 indicates free of charge.
authType	String	Authentication mode of an exclusive microservice engine. RBAC indicates security authentication, and NONE indicates no authentication.
status	String	Current status of an exclusive microservice engine.
externalAddress	String	Address for accessing an exclusive microservice engine in a VPC on the tenant side.
serviceEndpoint	Object	Address for accessing an exclusive microservice engine component in a VPC on the tenant side.
publicAddress	String	Public network access address of an exclusive microservice engine. You need to enable public network access.
publicServiceEndpoint	Object	Public network access address of an exclusive microservice engine component. You need to enable public network access.
totalInstance	Integer	Total number of instances supported by an exclusive microservice engine.
usedInstance	Integer	Total number of used instances.
availableInstance	Integer	Total number of available instances.
version	String	Current version of an exclusive microservice engine.
latestVersion	String	Latest version of an exclusive microservice engine.
createTime	Integer	Creation time of an exclusive microservice engine.
dueTo	Integer	Expiration time of an exclusive microservice engine.
latestJobId	Integer	ID of the latest job of an exclusive microservice engine.

Parameter	Type	Description
engineAdditionalActions	Array of String	Additional operations allowed by an exclusive microservice engine. Delete ForceDelete Upgrade Retry Configure
specType	String	Deployment type of an exclusive microservice engine. Fixed value: CSE2.
reference	Object	Additional information about an exclusive microservice engine.

Table 7-31 Endpoint

Parameter	Type	Description
kie	Object	Endpoint information of the configuration center.
serviceCenter	Object	Endpoint information of the service center.

Table 7-32 EntryPoint

Parameter	Type	Description
masterEntryPoint	String	Primary IPv4 address for accessing an exclusive microservice engine component in a VPC.
masterEntryPointIpv6	String	Primary IPv6 address for accessing an exclusive microservice engine component in a VPC.
slaveEntryPoint	String	Secondary IPv4 address for accessing an exclusive microservice engine component in a VPC.
slaveEntryPointIpv6	String	Secondary IPv6 address for accessing an exclusive microservice engine component in a VPC.
type	String	Type of an exclusive microservice engine component. CSE

Table 7-33 EngineReference

Parameter	Type	Description
vpc	String	VPC name.
vpcId	String	VPC ID. The value can contain up to 64 characters.
azList	Array of String	List of AZs for exclusive microservice engine deployment.
networkId	String	Subnet network ID of an exclusive microservice engine.
subnetCidr	String	IPv4 subnet division of an exclusive microservice engine.
subnetCidrV6	String	IPv6 subnet division of an exclusive microservice engine.
subnetGateway	String	Subnet gateway of an exclusive microservice engine.
publicIpId	String	Public IP address ID of an exclusive microservice engine. You need to enable public network access.
serviceLimit	Integer	Total number of microservices supported by a package.
instanceLimit	Integer	Total number of instances supported by a package.
inputs	Map <String,String >	Additional parameter of an exclusive microservice engine.

Status code: 400

Table 7-34 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-35 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query the microservice engine list.

```
GET /v2/{project_id}/enginemgr/engines
```

Example Response

```
{
  "total": 1,
  "data": [{
    "id": "891bf21a-4024-4f47-b38c-bd259ca8f10a",
    "name": "test",
    "enterpriseProjectId": "0",
    "enterpriseProjectName": "default",
    "type": "CSE_SHARE",
    "description": "",
    "flavor": "cse.s1.medium2",
    "payment": "0",
    "authType": "RBAC",
    "status": "Available",
    "externalAddress": "192.168.0.169",
    "serviceEndpoint": {
      "kie": {
        "masterEntrypoint": "https://192.168.0.169:30110",
        "masterEntrypointIpv6": "https://[2407:c080:11f0:11:b11d:675c:97ab:65f6]:30110",
        "slaveEntrypoint": null,
        "slaveEntrypointIpv6": null,
        "type": "REGISTRY"
      },
      "serviceCenter": {
        "masterEntrypoint": "https://192.168.0.169:30100",
        "masterEntrypointIpv6": "https://[2407:c080:11f0:11:b11d:675c:97ab:65f6]:30100",
        "slaveEntrypoint": null,
        "slaveEntrypointIpv6": null,
        "type": "REGISTRY"
      }
    },
    "publicAddress": "",
    "publicServiceEndpoint": {
      "kie": {
        "masterEntrypoint": "",
        "masterEntrypointIpv6": "",
        "slaveEntrypoint": "",
        "slaveEntrypointIpv6": "",
        "type": "REGISTRY"
      },
      "serviceCenter": {
        "masterEntrypoint": "",
        "masterEntrypointIpv6": "",
        "slaveEntrypoint": "",
        "slaveEntrypointIpv6": "",
        "type": "REGISTRY"
      }
    }
  ]
}
```

```
"totalInstance": 200,
"usedInstance": 0,
"availableInstance": 200,
"version": "2.3.1",
"latestVersion": "2.3.3",
"createTime": 1635576800332,
"dueTo": 4102415999000,
"latestJobId": 12339,
"engineAdditionalActions": ["Retry"],
"specType": "CSE2",
"reference": {
  "vpc": "vpc-test",
  "vpcId": "09902850-9454-4715-9764-018f0c3701hy",
  "azList": ["test"],
  "networkId": "88550801-e892-4f8e-b21b-f7147f604f69",
  "subnetCidr": "192.168.0.0/24",
  "subnetCidrV6": "2407:c080:11f0:11::/64",
  "subnetGateway": "192.168.0.2",
  "publicIpId": null,
  "serviceLimit": 200,
  "instanceLimit": 200,
  "inputs": {
    "is_arm_cluster": "false",
    "nodeFlavor": "s6.large.2"
  }
}
}]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3.3 Creating an Exclusive Microservice Engine

Function

This API is used to create an exclusive microservice engine.

URI

POST /v2/{project_id}/enginemgr/engines

Table 7-36 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .

Request

Table 7-37 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Table 7-38 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of an exclusive microservice engine. The value contains 3 to 24 characters, including letters, and digits, and starts with a letter but cannot end with a hyphen (-).
description	No	String	Description of an exclusive microservice engine. The value contains 0 to 255 characters.
payment	Yes	String	Billing mode of an exclusive microservice engine. Set it to the fixed value 1 .

Parameter	Mandatory	Type	Description
flavor	Yes	String	Flavor of an exclusive microservice engine. <ul style="list-style-type: none">• cse.s1.small2: HA 100-instance engine• cse.s1.medium2: HA 200-instance engine• cse.s1.large2: HA 500-instance engine• cse.s1.xlarge2: HA 2000-instance engine
azList	Yes	Array of String	List of AZs at the current region.
authType	Yes	String	Authentication mode of an exclusive microservice engine. RBAC indicates security authentication, and NONE indicates no authentication.
vpc	Yes	String	VPC name.
vpcId	Yes	String	VPC ID. The value can contain up to 64 characters.
networkId	Yes	String	Subnet ID of an exclusive microservice engine.
subnetCidr	Yes	String	Subnet division of an exclusive microservice engine.
publicIpId	No	String	ID of the public network address for accessing an exclusive microservice engine.
specType	Yes	String	Deployment type of an exclusive microservice engine. Fixed value: CSE2.
auth_cred	No	object	This parameter is mandatory when security authentication needs to be enabled for an exclusive microservice engine. It contains the authentication information of the engine.
inputs	No	Map<String,String>	Additional parameter of an exclusive microservice engine.

Table 7-39 object

Parameter	Mandatory	Type	Description
pwd	Yes	String	root password for enabling security authentication.

Response

Status code: 200

Table 7-40 Response body parameters

Parameter	Type	Description
id	String	ID of an exclusive microservice engine.
name	String	Name of an exclusive microservice engine.
jobId	Integer	ID of an exclusive microservice engine job.

Status code: 400

Table 7-41 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-42 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Create exclusive microservice engine **test**: billing mode is pay-per-use, flavor is cse.s1.small2, AZ is **test**, security authentication is disabled, and deployment type is CSE2.

```
POST https://{endpoint}/v2/{project_id}/enginemgr/engines
{
  "name": "test",
  "description": "test",
  "payment": "1",
  "flavor": "cse.s1.small2",
  "azList": ["test"],
  "authType": "NONE",
  "vpc": "vpc-demotest",
  "vpcId": "09902850-9454-4715-9764-018f0c37022",
  "networkId": "88550801-e892-4f8e-b21b-f7147f6229",
  "subnetCidr": "192.168.0.5/26",
  "specType": "CSE2",
  "inputs": {
    "nodeFlavor": "s6.large.2"
  }
}
```

Example Response

```
{
  "id": "891bf21a-4024-4f47-b38c-bd259ca8f10a",
  "name": "test",
  "jobId": 17655
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3.4 Querying Details About an Exclusive Microservice Engine

Function

This API is used to query details about an exclusive microservice engine.

URI

GET /v2/{project_id}/enginemgr/engines/{engine_id}

Table 7-43 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .
engine_id	Yes	String	ID of an exclusive microservice engine. See Querying the Exclusive Microservice Engine List .

Request

Table 7-44 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Response

Status code: 200

Table 7-45 Response body parameters

Parameter	Type	Description
id	String	ID of an exclusive microservice engine.
name	String	Name of an exclusive microservice engine.
projectId	String	ID of the project to which an exclusive microservice engine belongs.
enterpriseProjectId	String	ID of the enterprise project to which an exclusive microservice engine belongs.
enterpriseProjectName	String	Name of the enterprise project to which an exclusive microservice engine belongs.
type	String	Engine type. CSE: exclusive microservice engine.

Parameter	Type	Description
description	String	Description of an exclusive microservice engine.
beDefault	Boolean	Whether the exclusive microservice engine is the default engine.
flavor	String	Flavor of an exclusive microservice engine. <ul style="list-style-type: none">• cse.s1.small2: HA 100-instance engine• cse.s1.medium2: HA 200-instance engine• cse.s1.large2: HA 500-instance engine• cse.s1.xlarge2: HA 2000-instance engine
payment	String	Billing mode of an exclusive microservice engine. 1 indicates pay-per-use, and 2 indicates free of charge.
authType	String	Authentication mode of an exclusive microservice engine. RBAC indicates security authentication, and NONE indicates no authentication.
status	String	Current status of an exclusive microservice engine.
cceSpec	Object	CCE specification of an exclusive microservice engine.
externalEntry point	Object	Access address of an exclusive microservice engine.
publicAddress	String	Public network access address of an exclusive microservice engine. You need to enable public network access.
version	String	Current version of an exclusive microservice engine.
latestVersion	String	Latest version of an exclusive microservice engine.
createTime	Integer	Creation time of an exclusive microservice engine.
createUser	String	Creator of an exclusive microservice engine.
latestJobId	Integer	ID of the latest job of an exclusive microservice engine.

Parameter	Type	Description
engineAdditionalActions	Array of String	Additional operations allowed by an exclusive microservice engine. Delete ForceDelete Upgrade Retry
specType	String	Deployment type of an exclusive microservice engine. Fixed value: CSE2.
reference	Object	Additional information about an exclusive microservice engine.
vmIds	Array of String	ID list of VMs used by an exclusive microservice engine on the resource tenant side.

Table 7-46 EngineReference

Parameter	Type	Description
vpc	String	VPC name.
vpcId	String	VPC ID. The value can contain up to 64 characters.
azList	Array of String	List of AZs for exclusive microservice engine deployment.
networkId	String	Subnet network ID of an exclusive microservice engine.
subnetCidr	String	IPv4 subnet division of an exclusive microservice engine.
subnetCidrV6	String	IPv6 subnet division of an exclusive microservice engine.
subnetGateway	String	Subnet gateway of an exclusive microservice engine.
publicIpId	String	Public IP address ID of an exclusive microservice engine. You need to enable public network access.
serviceLimit	Integer	Total number of microservices supported by a package.
instanceLimit	Integer	Total number of instances supported by a package.

Parameter	Type	Description
inputs	Map <String,String >	Additional parameter of an exclusive microservice engine.

Table 7-47 Spec

Parameter	Type	Description
id	Integer	CCE specification ID of an exclusive microservice engine.
engineId	String	ID of an exclusive microservice engine.
specType	String	CCE cluster deployment type of an exclusive microservice engine.
cluster	String	CCE cluster information about an exclusive microservice engine. Currently, the value is null .
clusterId	String	CCE cluster ID of an exclusive microservice engine.
clusterNodes	Object	CCE node list of an exclusive microservice engine.
flavor	String	CCE cluster flavor of an exclusive microservice engine.
region	String	Region where the CCE cluster of an exclusive microservice engine is located.
version	String	CCE cluster version of an exclusive microservice engine.
extendParam	String	Additional parameter of the CCE cluster of an exclusive microservice engine.

Table 7-48 SpecClusterNode

Parameter	Type	Description
clusterNodes	Array of ClusterNode objects	CCE node information of an exclusive microservice engine.

Table 7-49 ClusterNode

Parameter	Type	Description
id	String	CCE node ID of an exclusive microservice engine.
az	String	AZ to which the CCE cluster of an exclusive microservice engine belongs.
ip	String	CCE node IP of an exclusive microservice engine.
label	String	CCE node label of an exclusive microservice engine.
status	String	CCE node label of an exclusive microservice engine.

Table 7-50 EngineExternalEndpoint

Parameter	Type	Description
externalAddress	String	Address for accessing an exclusive microservice engine in a VPC on the tenant side.
publicAddress	String	Public network access address of an exclusive microservice engine. You need to enable public network access.
serviceEndpoint	Object	Address for accessing an exclusive microservice engine component in a VPC on the tenant side.
publicServiceEndpoint	Object	Public network access address of an exclusive microservice engine component. You need to enable public network access.

Table 7-51 Endpoint

Parameter	Type	Description
kie	Object	Endpoint information of the configuration center.
serviceCenter	Object	Endpoint information of the service center.

Table 7-52 EntryPoint

Parameter	Type	Description
masterEntryPoint	String	Primary IPv4 address for accessing an exclusive microservice engine component in a VPC.
masterEntryPointIpv6	String	Primary IPv6 address for accessing an exclusive microservice engine component in a VPC.
slaveEntryPoint	String	Secondary IPv4 address for accessing an exclusive microservice engine component in a VPC.
slaveEntryPointIpv6	String	Secondary IPv6 address for accessing an exclusive microservice engine component in a VPC.
type	String	Type of an exclusive microservice engine component. CSE

Status code: 400

Table 7-53 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-54 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query details about an exclusive engine with specified **engine_id**.

```
GET /v2/{project_id}/enginemgr/engines/{engine_id}
```


Example Response

```
{
  "id": "891bf21a-4024-4f47-b38c-bd259ca8f10a",
  "name": "test",
  "description": "",
  "authType": "RBAC",
  "flavor": "cse.s1.medium2",
  "payment": "0",
  "version": "2.3.1",
  "latestVersion": "2.3.3",
  "status": "Creating",
  "beDefault": true,
  "createUser": "test",
  "createTime": 1635576800332,
  "cceSpec": {
    "id": 7465,
    "engineId": "891bf21a-4024-4f47-b38c-bd259ca8f10a",
    "specType": "CCE",
    "cluster": null,
    "clusterId": "41115a6f-912f-11eb-9af9-0255ac100188",
    "clusterNodes": {
      "clusterNodes": [
        {
          "id": "c13aaf5c-2192-421c-8e03-522e2b9a06b5",
          "az": "test",
          "ip": "172.31.25.277",
          "label": "test",
          "status": "Active"
        }
      ]
    },
    "flavor": null,
    "region": "test",
    "version": "",
    "extendParam": ""
  },
  "externalEntrypoint": {
    "externalAddress": "192.168.0.169",
    "publicAddress": "",
    "serviceEndpoint": {
      "kie": {
        "masterEntrypoint": "https://192.168.0.169:30110",
        "masterEntrypointIpv6": "https://[2407:c080:11f0:11:b11d:675c:97ab:65f6]:30110",
        "slaveEntrypoint": null,
        "slaveEntrypointIpv6": null,
        "type": "REGISTRY"
      },
      "serviceCenter": {
        "masterEntrypoint": "https://192.168.0.169:30100",
        "masterEntrypointIpv6": "https://[2407:c080:11f0:11:b11d:675c:97ab:65f6]:30100",
        "slaveEntrypoint": null,
        "slaveEntrypointIpv6": null,
        "type": "REGISTRY"
      }
    }
  },
  "publicServiceEndpoint": {
    "kie": {
      "masterEntrypoint": "https://192.168.0.169:30110",
      "masterEntrypointIpv6": null,
      "slaveEntrypoint": null,
      "slaveEntrypointIpv6": null,
      "type": "REGISTRY"
    },
    "serviceCenter": {
      "masterEntrypoint": "https://192.168.0.169:30100",
      "masterEntrypointIpv6": null,
      "slaveEntrypoint": null,
      "slaveEntrypointIpv6": null,
      "type": "REGISTRY"
    }
  }
}
```

```
},
"reference": {
  "vpc": "vpc-test",
  "vpcId": "09902850-9454-4715-9764-018f0c3701hy",
  "azList": ["test"],
  "networkId": "88550801-e892-4f8e-b21b-f7147f604f69",
  "subnetCidr": "192.168.0.0/24",
  "subnetCidrV6": "2407:c080:11f0:11::/64",
  "subnetGateway": "192.168.0.2",
  "publicIpId": null,
  "serviceLimit": 200,
  "instanceLimit": 200,
  "inputs": {
    "is_arm_cluster": "false",
    "nodeFlavor": "s6.large.2"
  }
},
"latestJobId": 12339,
"enterpriseProjectId": "0",
"enterpriseProjectName": "default",
"engineAdditionalActions": ["Retry"],
"specType": "CSE2",
"type": "CSE",
"projectId": "string",
"vmIds": [""]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3.5 Deleting an Exclusive Microservice Engine

Function

This API is used to delete an exclusive microservice engine.

URI

DELETE /v2/{project_id}/enginemgr/engines/{engine_id}

Table 7-55 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .
engine_id	Yes	String	ID of an exclusive microservice engine.

Request

Table 7-56 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Response

Status code: 200

Table 7-57 Response body parameters

Parameter	Type	Description
id	String	ID of an exclusive microservice engine.
name	String	Name of an exclusive microservice engine.
jobId	Integer	ID of an exclusive microservice engine job.

Status code: 400

Table 7-58 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.

Parameter	Type	Description
detail	String	Location details.

Status code: 500

Table 7-59 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Delete an engine with specified **engine_id**.

```
DELETE /v2/{project_id}/enginemgr/engines/{engine_id}
```

Example Response

```
{  
  "id": "891bf21a-4024-4f47-b38c-bd259ca8f10a",  
  "name": "test",  
  "jobId": 7256  
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

7.3.6 Querying Details About an Exclusive Microservice Engine Job

Function

This API is used to query details about an exclusive microservice engine job.

URI

GET /v2/{project_id}/enginemgr/engines/{engine_id}/jobs/{job_id}

Table 7-60 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Tenant sub-project ID, which must be unique. The value contains 1 to 64 characters. See Obtaining a Project ID .
engine_id	Yes	String	ID of an exclusive microservice engine. See Querying the Exclusive Microservice Engine List .
job_id	Yes	String	ID of an exclusive microservice engine. See Querying Details About an Exclusive Microservice Engine .

Request

Table 7-61 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token.
X-Enterprise-Project-ID	No	String	Enterprise project ID. If this parameter is not set, the default enterprise project is default and the ID is 0 .

Response

Status code: 200

Table 7-62 Response body parameters

Parameter	Type	Description
id	Integer	Job ID.
engineId	String	ID of an exclusive microservice engine.
type	String	Job type. Create Delete Upgrade Configure
description	String	Job description.
status	String	Current status of a job. Init: initializing Executing Error Timeout Finished
scheduling	Integer	Whether a job is being executed. 0 : no; 1 : yes.
createUser	String	Creator of a job.
startTime	Integer	Start time of a job.
endTime	Integer	End time of a job.
context	String	Job execution context.
tasks	Array of TaskSteps objects	Job phases.

Table 7-63 TaskSteps

Parameter	Type	Description
taskName	String	Name of a phase.
taskNames	Array of String	List of procedures contained in the current phase.
status	String	Status of the phase.
startTime	Integer	Start time of a phase.
endTime	Integer	End time of a phase.

Parameter	Type	Description
taskExecutorBrief	Object	Job metadata.
tasks	Array of Task objects	Sub-jobs in the phase.

Table 7-64 TaskExecutorBrief

Parameter	Type	Description
duration	Integer	Duration of a sub-job.
description	String	Sub-job description.

Table 7-65 Task

Parameter	Type	Description
jobId	Integer	ID of the job to which the sub-job belongs.
id	Integer	Sub-job ID, which is in UUID format.
type	String	Sub-job type.
assigned	String	Executor of a sub-job.
taskName	String	Sub-job name.
engineName	String	Name of the engine to which the sub-job belongs.
taskOrder	Integer	Sequence in which sub-jobs are executed, in ascending order.
status	String	Sub-job status.
startTime	Integer	Start time of a sub-job.
endTime	Integer	End time of a sub-job.
createTime	Integer	Creation time of a sub-job.
updateTime	Integer	Update time of a sub-job.
timeout	Integer	Whether a sub-job times out.
log	String	Sub-job details, which are auxiliary information generated during the execution.
output	String	Sub-job output information.

Parameter	Type	Description
taskExecutorBrief	Object	Sub-job metadata.

Status code: 400

Table 7-66 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 7-67 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query details about a job with specified **job_id** in an exclusive engine with specified **engine_id**.

```
GET /v2/{project_id}/enginemgr/engines/{engine_id}/jobs/{job_id}
```

Example Response

```
{
  "id": 12339,
  "engineId": "ad76972c-a743-4770-859c-c98a680f4d98",
  "type": "Create",
  "description": "",
  "status": "Finished",
  "scheduling": 0,
  "createUser": "test",
  "startTime": 1636540095901,
  "endTime": 1636540460230,
  "context": "",
  "tasks": [{
    "taskName": "resourcePrepare",
    "taskNames": ["TenantTaskExecutor"],
    "status": "Finished",
```



```
"startTime": 1636540100216,
"endTime": 1636540110215,
"taskExecutorBrief": {
  "duration": 60000,
  "description": "Resource preparation"
},
"tasks": [{
  "jobId": 12339,
  "id": 89117,
  "type": "Create",
  "assigned": "",
  "taskName": "TenantTaskExecutor",
  "engineName": "test",
  "taskOrder": 0,
  "status": "Init",
  "startTime": 1636540100216,
  "endTime": 1636540110215,
  "createTime": 1636540095910,
  "updateTime": 1636540110230,
  "timeout": -1,
  "log": "",
  "output": "{}",
  "taskExecutorBrief": {
    "duration": 60000,
    "description": "Tenant processing"
  }
}]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [CSE Error Codes](#).

8 ServiceComb API

8.1 API Calling

CSE provides REST APIs, allowing you to call APIs using HTTPS.

NOTICE

APIs in [Microservice](#), [Schema](#), [Microservice Instance](#), and [Dependency](#) are available only in ME-Riyadh, CN-Hong Kong, and AP-Singapore.

To call the ServiceComb APIs of an exclusive ServiceComb engine, do as follows:

1. Log in to ServiceStage and choose **Cloud Service Engine > Engines**.
2. Select the target microservice engine from the **Microservice Engine** drop-down list in the upper part of the page.

When calling the APIs in [Authentication](#), [Microservice](#), [Schema](#), [Microservice Instance](#), and [Dependency](#), view or click to copy the service center address of the engine.

When calling the APIs in [Configuration Management](#), view or click to copy the configuration center address of the engine.

3. Call the API by referring to [Calling APIs](#). In [Request URI](#), replace **{Endpoint}** with the obtained service center address.

8.2 Authentication

8.2.1 Obtaining the User Token of an Exclusive Microservice Engine

Function

This API is intended only for exclusive microservice engines with security authentication enabled.

Before accessing APIs of an exclusive microservice engine with security authentication enabled, you need to call this API to obtain a user token. This token is the unique credential for accessing the APIs.

The token must be contained in the request header when an access request is sent. The format is as follows:

```
Authorization:Bearer{Token}
```

Restrictions

None

URI

POST /v4/token

Request

Table 8-1 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Account name.
password	Yes	String	Password of the account.

Response

Status code: 200

Table 8-2 Response body parameter

Parameter	Type	Description
token	String	User token, which is valid for 12 hours.

Status code: 401

Table 8-3 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-4 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

```
POST https://{Service center address}/v4/token
```

```
{  
  "name": "root",  
  "password": "*****"  
}
```

Example Response

Status code: 200

OK

```
{  
  "token" :  
  "****bGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50Ijoicm9vdCIsImV4cCI6MTY1MDU5MTcwMSwicm9sZXMiOiSiYWRTaW4iXX0.WKwNAjaYMMCSjNX0qCGCeyh13FJRzLousxoXlThdkMwKH-  
pXEmG51_SguH0LLHOZolc8gNJq-ilQg4bxTo1s0pnQZIS3wma0qvE-  
MzaYnFguTuHM7rxD7eZdwnbUe3dhnw9xRqR1hcd-lTuBbLoL9fbED4U_63loEDyBCI9D_l0F86uGzpUysCvC-  
t6MrJHgi7miUaO7ZZQmSAUNhmbEoN8IIvp-QtP_cWNWtWaFO-  
eoQrmCT2FdIYiB9MCuELr9-5EGM_mFLPgs6E4fyIGiGHY7IwoGUKOCW5w6Jb0L-2JxeUe3eO15Md5kzOIAE_EYU  
ATxCbJ5GmgpSSJf*****"  
}
```

Status Code

Status Code	Description
200	OK
401	Unauthorized
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3 Microservice

8.3.1 Querying Information About a Microservice

Function

This API is used to query the definition information about a microservice based on **service_id**.

Restrictions

None

URI

GET /v4/{project_id}/registry/microservices/{service_id}

Table 8-5 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^[.*\$]</code> . See Querying Information About All Microservices .

Request

Table 8-6 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-7 Response body parameter

Parameter	Type	Description
service	MicroService object	Microservice information.

Table 8-8 MicroService

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: $\wedge.*\$$
environment	String	Microservice environment. Value: development, testing, acceptance, or production.

Parameter	Type	Description
appld	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
serviceName	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
version	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]\$ ^[0-9]+([0-9]+)\$</code>
description	String	Microservice description. The value contains a maximum of 256 characters.
level	String	Microservice level. Value: FRONT, MIDDLE, or BACK.
registerBy	String	Microservice registration mode. Value: SDK, PLATFORM, SIDECAR, or UNKNOWN.
schemas	Array of strings	Microservice schema content. The value must be 1 to 160 bytes long. Only digits, letters, underscores (<code>_</code>), hyphens (<code>-</code>), and periods (<code>.</code>) are allowed. An array contains a maximum of 100 schemas.
status	String	Microservice status. Value: UP or DOWN. Default value: UP. Value: <ul style="list-style-type: none">• UP• DOWN
timestamp	String	Microservice registration time.
modTimestamp	String	Latest modification time (UTC).
framework	Framework object	Development framework.
paths	Array of ServicePath objects	Service path.

Table 8-9 Framework

Parameter	Type	Description
name	String	Microservice development framework. Default value: UNKNOWN.
version	String	Version of the microservice development framework.

Table 8-10 ServicePath

Parameter	Type	Description
Path	String	Route address.
Property	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Status code: 400

Table 8-11 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-12 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query details about the microservice whose ID is **819706e21b7173306797d19922ce4231441c17c5**.


```
GET https://{endpoint}/v4/{project_id}/registry/microservices/  
819706e21b7173306797d19922ce4231441c17c5
```

Example Response

Status code: 200

Successfully queried.

```
{  
  "service": {  
    "serviceId": "819706e21b7173306797d19922ce4231441c17c5",  
    "appId": "default",  
    "serviceName": "SERVICECENTER",  
    "version": "2.4.8",  
    "level": "BACK",  
    "schemas": [  
      "servicecenter.grpc.api.ServiceCtrl",  
      "servicecenter.grpc.api.ServiceInstanceCtrl"  
    ],  
    "status": "UP",  
    "timestamp": "1616426688",  
    "modTimestamp": "1616426688",  
    "environment": "development"  
  }  
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.2 Deleting Definition Information About a Microservice

Function

This API is used to delete definition and related information about a microservice, and deregister all instances of the microservice.

Restrictions

None

URI

DELETE /v4/{project_id}/registry/microservices/{service_id}

Table 8-13 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .

Table 8-14 Query parameter

Parameter	Mandatory	Type	Description
force	No	Boolean	Whether to forcibly delete information about a microservice. true: forcible deletion; false: non-forcible deletion. If you choose forcible deletion, all service instances are automatically deregistered and related service dependencies are deleted. If this parameter is not transferred, the service cannot be deleted when it has an instance. Default value: false.

Request

Table 8-15 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 400

Table 8-16 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-17 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Delete the microservice whose ID is **e0f0da073f2c91e8979a89ff2d7c69t6**.

```
DELETE https://{endpoint}/v4/d9f4da085f2c11e8959a00ff2d7c69b7/registry/microservices/  
e0f0da073f2c91e8979a89ff2d7c69t6
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.3 Querying Information About All Microservices

Function

This API is used to query definition information about microservices that meet filter criteria.

Restrictions

None

URI

GET /v4/{project_id}/registry/microservices

Table 8-18 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none"> For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default. For an exclusive microservice engine, enter the fixed value default.

Request

Table 8-19 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-20 Response body parameter

Parameter	Type	Description
services	Array of MicroService objects	Microservice list.

Table 8-21 MicroService

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^.*\$</code>
environment	String	Microservice environment. Value: development, testing, acceptance, or production. You can use the API for uploading schemas in batches to add or modify an existing schema only when the value is development , testing , or acceptance . Default value: development.
appld	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
serviceName	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
version	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]\$^[0-9]+([0-9]+)\$</code>
description	String	Microservice description. The value contains a maximum of 256 characters.
level	String	Microservice level. Value: FRONT, MIDDLE, or BACK.
registerBy	String	Microservice registration mode. Value: SDK, PLATFORM, SIDECAR, or UNKNOWN.
schemas	Array of strings	Microservice schema content. The value must be 1 to 160 bytes long. Only digits, letters, underscores (<code>_</code>), hyphens (<code>-</code>), and periods (<code>.</code>) are allowed. An array contains a maximum of 100 schemas.
status	String	Microservice status. Value: UP or DOWN. Default value: UP.

Parameter	Type	Description
timestamp	String	Microservice registration time.
modTimestamp	String	Latest modification time (UTC).
framework	Framework object	Development framework.
paths	Array of ServicePath objects	Service path.

Table 8-22 Framework

Parameter	Type	Description
name	String	Microservice development framework. Default value: UNKNOWN.
version	String	Version of the microservice development framework.

Table 8-23 ServicePath

Parameter	Type	Description
Path	String	Route address.
Property	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Status code: 400

Table 8-24 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-25 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query information about all registered microservices.

```
GET https://{endpoint}/v4/{project_id}/registry/microservices
```

Example Response

Status code: 200

Successfully queried.

```
{
  "services": [
    {
      "serviceId": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
      "appld": "default",
      "serviceName": "test",
      "version": "1.0.0",
      "description": "this is a test",
      "level": "BACK",
      "status": "UP",
      "timestamp": "1650543950",
      "modTimestamp": "1650543950"
    },
    {
      "serviceId": "dcc6c1073eab3cadb47cea2e1a874b7883b02a63",
      "appld": "test",
      "serviceName": "test1",
      "version": "1.0.0",
      "level": "BACK",
      "status": "UP",
      "timestamp": "1650544223",
      "modTimestamp": "1650544223"
    }
  ]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.4 Creating Static Information for a Microservice

Function

This API is used to create static information for a microservice before registering a microservice instance. The registered instance is associated with the static information based on **serviceId**. One service corresponds to multiple instances.

serviceId can be customized. If **serviceId** is not customized, the system generates a random service ID.

Restrictions

None

URI

POST /v4/{project_id}/registry/microservices

Table 8-26 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Request

Table 8-27 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-28 Request body parameters

Parameter	Mandatory	Type	Description
service	Yes	MicroService object	Microservice information.
rules	No	Array of Rule objects	Blacklist and whitelist.
instances	No	Array of MicroServiceInstance objects	Instance information.
tags	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Table 8-29 MicroService

Parameter	Mandatory	Type	Description
serviceId	No	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^.*\$</code>
environment	No	String	Microservice environment. Value: development, testing, acceptance, or production. You can use the API for uploading schemas in batches to add or modify an existing schema only when the value is development, testing, or acceptance . Default value: development.
appId	No	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
serviceName	Yes	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
version	No	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]\$^[0-9]+(.[0-9]+)\$</code>
description	No	String	Microservice description. The value contains a maximum of 256 characters.
level	No	String	Microservice level. Value: FRONT, MIDDLE, or BACK.
registerBy	No	String	Microservice registration mode. Value: SDK, PLATFORM, SIDECAR, or UNKNOWN.
schemas	No	Array of strings	Foreign key ID of a microservice access schema. The array length supports a maximum of 100 schemas.

Parameter	Mandatory	Type	Description
status	No	String	Microservice status. Value: UP or DOWN. Default value: UP.
timestamp	No	String	Microservice registration time.
modTimestamp	No	String	Latest modification time (UTC).
framework	No	Framework object	Development framework.
paths	No	Array of ServicePath objects	Service path.

Table 8-30 Framework

Parameter	Mandatory	Type	Description
name	No	String	Microservice development framework. Default value: UNKNOWN .
version	No	String	Version of the microservice development framework.

Table 8-31 ServicePath

Parameter	Mandatory	Type	Description
Path	No	String	Route address.
Property	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Table 8-32 Rule

Parameter	Mandatory	Type	Description
ruleId	No	String	Customized rule ID.
ruleType	No	String	Rule type. Value: WHITE or BLACK.

Parameter	Mandatory	Type	Description
attribute	No	String	If the value starts with tag_XXX , the attributes are filtered by Tag . Otherwise, the attributes are filtered by serviceld , Appld , ServiceName , Version , Description , Level , or Status .
pattern	No	String	Matching rule. The value is a regular expression containing 1 to 64 characters.
description	No	String	Rule description.
timestamp	No	String	Time when a rule is created. This parameter is used only when you query rules.
modTimestamp	No	String	Update time.

Table 8-33 MicroServiceInstance

Parameter	Mandatory	Type	Description
instanceId	No	String	Instance ID, which must be unique. The instance ID is generated by the service center.
serviceld	No	String	Microservice ID, which must be unique. During instance creation, use the service ID in the URL instead of the service ID here.
version	No	String	Microservice version.
hostName	Yes	String	Host information.
endpoints	Yes	Array of strings	Access address information.
status	No	String	Instance status. Value: UP, DOWN, STARTING, or OUTOFSERVICE. Default value: UP.
properties	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Parameter	Mandatory	Type	Description
healthCheck	No	HealthCheck object	Health check information.
dataCenterInfo	No	DataCenterInfo object	Data center information.
timestamp	No	String	Time when an instance is created, which is automatically generated.
modTimestamp	No	String	Update time.

Table 8-34 HealthCheck

Parameter	Mandatory	Type	Description
mode	Yes	String	Heartbeat mode. Value: push or pull.
port	No	Integer	Port.
interval	Yes	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Yes	Integer	Maximum retries.

Table 8-35 DataCenterInfo

Parameter	Mandatory	Type	Description
name	Yes	String	Region name.
region	Yes	String	Region.
availableZone	Yes	String	AZ.

Response

Status code: 200

Table 8-36 Response body parameter

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique.

Status code: 400

Table 8-37 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-38 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Create static information about a microservice: microservice name is **test**, microservice version is **1.0.0**, rule type is whitelist, host information of the instance is **instanceTest**, and access address is **rest:127.0.0.1:8080**.

POST https://{endpoint}/v4/{project_id}/registry/microservices

```
{
  "service": {
    "appld": "default",
    "serviceName": "test",
    "version": "1.0.0",
    "description": "this is a test"
  },
  "rules": [{
    "ruleType": "WHITE",
    "attribute": "tag_123",
    "pattern": "aaa"
  }],
  "instances": [{
    "hostName": "instanceTest",
    "endpoints": ["rest:127.0.0.1:8080"]
  }],
  "tags": {
    "test_tag1": "test_tag1",
    "test_tag2": "test_tag2",
    "test_tag3": "test_tag3"
  }
}
```

Example Response

Status code: 200

Successfully created. **serviceld** indicates the ID generated for the microservice. Subsequent operations are operated based on the microservice ID.

```
{  
  "serviceld" : "8aed80ea052ac04a64dfc79c24f2170224d074f5"  
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.5 Deleting Static Information About Microservices in Batches

Function

This API is used to delete the definitions and related information about microservices in batches, and deregister all instances of the microservices.

Restrictions

None

URI

DELETE /v4/{project_id}/registry/microservices

Table 8-39 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Table 8-40 Query parameter

Parameter	Mandatory	Type	Description
force	No	Boolean	Whether to forcibly delete static information about microservices. true: forcible deletion; false: non-forcible deletion. If you choose forcible deletion, all service instances are automatically deregistered and related service dependencies are deleted. If this parameter is not transferred, the service cannot be deleted when it has an instance. Default value: false.

Request

Table 8-41 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-42 Request body parameter

Parameter	Mandatory	Type	Description
serviceIds	Yes	Array of strings	Service ID list. See Querying Information About All Microservices .

Response

Status code: 200

Table 8-43 Response body parameter

Parameter	Type	Description
services	Array of DelServicesRspInfo objects	List of services to be deleted.

Table 8-44 DelServicesRspInfo

Parameter	Type	Description
serviceId	String	Microservice ID.

Status code: 400

Table 8-45 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-46 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Delete the static information about the microservices whose IDs are **id1** and **id2**.

```
DELETE https://{endpoint}/v4/{project_id}/registry/microservices
{
  "serviceIds" : [ "id1", "id2" ]
}
```

Example Response

Status code: 200

Successfully deleted.

```
{
  "services" : [ {
    "serviceId" : "id1"
  },
  {
    "serviceId" : "id2"
  }
]
```

```
]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.6 Modifying Extended Attributes of a Microservice

Function

This API is used to modify static information about a microservice. To update some fields in the static information, input all static information (including the fields that do not need to be updated) in JSON format.

Restrictions

None

URI

PUT /v4/{project_id}/registry/microservices/{service_id}/properties

Table 8-47 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Parameter	Mandatory	Type	Description
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^[.*\$]</code> . See Querying Information About All Microservices .

Request

Table 8-48 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: <code>Authorization:Bearer {Token}</code></p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-49 Request body parameter

Parameter	Mandatory	Type	Description
properties	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Response

Status code: 400

Table 8-50 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-51 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Modify extended information about a microservice with specified **service_id**.

```
PUT https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/properties
```

```
{  
  "properties": {  
    "a": "test"  
  }  
}
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.3.7 Querying the Unique Service or Schema ID of a Microservice

Function

This API is used to query the unique service or schema ID of a microservice based on filter criteria.

URI

GET /v4/{project_id}/registry/existence

Table 8-52 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Table 8-53 Query parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Request type. <ul style="list-style-type: none">microserviceschema NOTE <ul style="list-style-type: none">If type is set to microservice, env, appId, serviceName, and version are mandatory.If type is set to schema, serviceId and schemaId are mandatory.
env	No	String	Microservice environment. Value: development, testing, acceptance, or production.

Parameter	Mandatory	Type	Description
appld	Yes	String	Application ID, which needs to be transferred when the resource type is microservice . The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
serviceName	Yes	String	Microservice name, which needs to be transferred when the resource type is microservice . The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
version	Yes	String	Microservice version, which needs to be transferred when the resource type is microservice . The value contains 1 to 64 characters. Regular expression: <code>^[0-9]\$ ^[0-9]+(\.[0-9]+)\$</code>
serviceId	Yes	String	Microservice ID, which needs to be transferred when the resource type is schema . The value contains 1 to 64 characters. Regular expression: <code>^\.*\$</code>
schemald	Yes	String	Schema ID, which needs to be transferred when the resource type is schema . The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]{1,160}\$ ^[a-zA-Z0-9][a-zA-Z0-9_-]{0,158}[a-zA-Z0-9]\$</code>

Request

Table 8-54 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-55 Response header parameter

Parameter	Type	Description
X-Schema-Summary	String	Summary of a microservice schema.

Table 8-56 Response body parameters

Parameter	Type	Description
serviceId	String	When a service is queried, a service ID is returned.
schemaId	String	When a schema is queried, a schema ID is returned.

Status code: 400

Table 8-57 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-58 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query **serviceld** based on **microservice** as **type**, **default** as **appld**, **service** as **serviceName**, and **1.0.0** as **version**.

```
GET https://{endpoint}/v4/{project_id}/registry/existence?  
type=microservice&appld=default&serviceName=service&version=1.0.0
```

Example Response

Status code: 200

Successfully queried. **serviceld** or **schemald** is returned.

```
{  
  "serviceld" : "8aed80ea052ac04a64dfc79c24f2170224d074f5"  
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.4 Schema

8.4.1 Querying a Microservice Schema

Function

Query a microservice schema based on **service_id** and **schema_id**.

Restrictions

None

URI

GET /v4/{project_id}/registry/microservices/{service_id}/schemas/{schema_id}

Table 8-59 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .
schema_id	Yes	String	Schema ID. See Querying All Schema Information About a Microservice .

Request

Table 8-60 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-61 Response header parameter

Parameter	Type	Description
X-Schema-Summary	String	Schema summary.

Table 8-62 Response body parameter

Parameter	Type	Description
schema	String	Schema content.

Status code: 400

Table 8-63 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.

Parameter	Type	Description
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-64 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query details about a schema with specified **service_id** and **schema_id**.

```
GET https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/schemas/{schema_id}
```

Example Response

Status code: 200

Successfully queried. If the summary exists, the value of **X-Schema-Summary** in the header is the summary of the schema.

```
{
  "schema" : "---\nswagger: \"2.0\"\ninfo:\n  version: \"1.0.0\"\n  title: \"swagger definition for\ncom.service.provider.controller.ProviderImpl\"\n  x-java-interface:\n    \"cse.gen.springmvc.provider.provider.ProviderImplIntf\"\n  basePath: \"/provider\"\n  consumes:\n    \"application/json\"\n  produces:\n    \"application/json\"\n  paths:\n    /helloworld:\n      get:\n        operationId:\n          \"helloworld\"\n        produces:\n          - \"application/json\"\n        parameters:\n          - name: \"name\"\n            in: \"query\"\n            required: true\n            type: \"string\"\n        responses:\n          200:\n            description:\n              \"response of 200\"\n            schema:\n              type: \"string\"\n}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.4.2 Modifying a Microservice Schema

Function

Modify a microservice schema based on `schema_id`.

Restrictions

In versions earlier than 2.3.33, schema overwriting is not supported in production environment. In version 2.3.33 and later, the environment variable `SCHEMA_EDITABLE` can be used to support schema overwriting.

URI

PUT `/v4/{project_id}/registry/microservices/{service_id}/schemas/{schema_id}`

Table 8-65 Path parameters

Parameter	Mandatory	Type	Description
<code>project_id</code>	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
<code>service_id</code>	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^.*\$</code> . See Querying Information About All Microservices .
<code>schema_id</code>	Yes	String	Microservice schema ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]{1,160}\$ ^[a-zA-Z0-9][a-zA-Z0-9_-.]{0,158}[a-zA-Z0-9]\$</code> . See Querying All Schema Information About a Microservice .

Request

Table 8-66 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required. The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine .

Table 8-67 Request body parameters

Parameter	Mandatory	Type	Description
schema	Yes	String	Schema content.
summary	No	String	Schema summary.

Response

Status code: 400

Table 8-68 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-69 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Modify the content and summary of a schema with specified **service_id** and **schema_id**.

```
PUT https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/schemas/{schema_id}
{
  "schema": "----\nswagger: \"2.0\"\ninfo:\n  version: \"1.0.0\"\n  title: \"swagger definition for\ncom.service.provider.controller.ProviderImpl\"\n  x-java-interface:\n\"cse.gen.springmvc.provider.provider.ProviderImplIntf\"\n  basePath: \"/provider\"\n  consumes:\n- \"application/json\"\n  produces:\n- \"application/json\"\n  paths:\n    /helloworld:\n      get:\n        operationId:\n\"helloworld\"\n        produces:\n- \"application/json\"\n        parameters:\n- name: \"name\"\n  in: \"query\"\n  required: true\n  type: \"string\"\n  responses:\n    200:\n      description:\n\"response of 200\"\n      schema:\n        type: \"string\"\n  \"summary\": \"test\"\n}
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.4.3 Querying All Schema Information About a Microservice

Function

Query all schema information (including **schemald** and **summary**) about a microservice.

URI

GET /v4/{project_id}/registry/microservices/{service_id}/schemas

Table 8-70 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .

Table 8-71 Query parameter

Parameter	Mandatory	Type	Description
withSchema	No	Integer	Whether to query the schema content. Default value: 0. 0: Only schema_id and summary are displayed. 1: schema_id , summary , and schema are displayed.

Request

Table 8-72 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-73 Response body parameter

Parameter	Type	Description
schemas	Array of Schema objects	Schema list.

Table 8-74 Schema

Parameter	Type	Description
schemald	String	Schema ID.
schema	String	Schema content.
summary	String	Schema summary.

Status code: 400

Table 8-75 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-76 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query the schema of a microservice with specified **service_id**, including **schema_id**, **summary**, and **schema**.

```
GET /v4/{project_id}/registry/microservices/{service_id}/schemas?withSchema=1
```

Example Response

Status code: 200

Successfully queried.

```
{
  "schemas": [{
    "schemaId": "xxxxmvc",
    "schema": "---\nswagger: \"2.0\"\ninfo:\n  version: \"1.0.0\"\n  title: \"swagger definition for com.service.provider.controller.ProviderImpl\"\n  x-java-interface: \n  cse.gen.springmvc.provider.provider.ProviderImplIntf\n  basePath: \"/provider\"\n  consumes:\n  - \"application/json\"\n  produces:\n  - \"application/json\"\n  paths:\n    /helloworld:\n      get:\n        operationId: \n        helloworld\n        produces:\n        - \"application/json\"\n        parameters:\n        - name: \"name\"\n        in: \"query\"\n        required: true\n        type: \"string\"\n        responses:\n        200:\n          description: \n          response of 200\n          schema:\n            type: \"string\"\n            xxx\n        summary: \"abcd7b4072ef2d7a5fc9aefccf03e5548029ae31c6cd5fc29da7685d6d9e14adea3\"
  }
}]
```

Status Code

Status Code	Description
200	OK

Status Code	Description
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5 Microservice Instance

8.5.1 Registering a Microservice Instance

Function

This API is used to register a microservice instance after a microservice is created.

Information about the instance must be provided during registration.

instanceid can be customized. If it is customized, the new instance ID will overwrite the original one. If it is not customized, the system automatically generates an ID. If the endpoints are duplicate, the original ID is used.

URI

POST /v4/{project_id}/registry/microservices/{service_id}/instances

Table 8-77 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Parameter	Mandatory	Type	Description
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^[.*\$]</code> . See Querying Information About All Microservices .

Request

Table 8-78 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: <code>Authorization:Bearer {Token}</code></p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-79 Request body parameter

Parameter	Mandatory	Type	Description
instance	Yes	MicroServiceInstance object	Microservice instance information.

Table 8-80 MicroServiceInstance

Parameter	Mandatory	Type	Description
instanceId	No	String	Instance ID, which must be unique. The instance ID is generated by the service center.
serviceId	No	String	Microservice ID, which must be unique. During instance creation, use the microservice ID in the URL instead of the microservice ID specified here.
version	No	String	Microservice version.
hostName	Yes	String	Host information.
endpoints	No	Array of strings	Access address information.
status	No	String	Instance status. Value: UP, DOWN, STARTING, or OUTOFSERVICE. Default value: UP.
properties	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.
healthCheck	No	HealthCheck object	Health check information.
dataCenterInfo	No	DataCenterInfo object	Data center information.
timestamp	No	String	Time when an instance is created, which is automatically generated.
modTimestamp	No	String	Update time.

Table 8-81 HealthCheck

Parameter	Mandatory	Type	Description
mode	Yes	String	Heartbeat mode. Value: push or pull.
port	No	Integer	Port.

Parameter	Mandatory	Type	Description
interval	Yes	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Yes	Integer	Maximum retries.

Table 8-82 DataCenterInfo

Parameter	Mandatory	Type	Description
name	Yes	String	Region name.
region	Yes	String	Region.
availableZone	Yes	String	AZ.

Response

Status code: 200

Table 8-83 Response body parameter

Parameter	Type	Description
instanceId	String	Instance ID.

Status code: 400

Table 8-84 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-85 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Register a microservice instance: host information is **localhost**, heartbeat mode is **push**, heartbeat interval is **30s**, region is **r1**, name is **dc**, and AZ is **az1**.

```
POST https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances
```

```
{
  "instance": {
    "endpoints": [ "grpc://127.0.1.312:9980", "rest://127.0.0.111:8081" ],
    "hostName": "localhost",
    "status": "UP",
    "properties": {
      "_TAGS": "A, B",
      "attr1": "a",
      "nodeIP": "127.0.0.1"
    },
    "dataCenterInfo": {
      "name": "dc",
      "region": "r1",
      "availableZone": "az1"
    },
    "healthCheck": {
      "mode": "push",
      "interval": 30,
      "times": 3
    }
  }
}
```

Example Response

Status code: 200

Successfully registered.

```
{
  "instanceId": "8540bb8b693c4ad1a7fb6a756c415244"
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.2 Querying a Microservice Instance Based on service_id

Function

This API is used to query all instances of a microservice based on **service_id** after the instances are registered.

URI

GET /v4/{project_id}/registry/microservices/{service_id}/instances

Table 8-86 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .

Table 8-87 Query parameter

Parameter	Mandatory	Type	Description
tags	No	String	Tag. When there are multiple tags, separate them using commas (.). Regular expression: ^[a-zA-Z][a-zA-Z0-9_-]{0,63}\$

Request

Table 8-88 Request header parameters

Parameter	Mandatory	Type	Description
X-ConsumerId	No	String	Microservice consumer ID, which must be unique.
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-89 Response body parameter

Parameter	Type	Description
instances	Array of MicroServiceInstance objects	Instance list.

Table 8-90 MicroServiceInstance

Parameter	Type	Description
instanceId	String	Instance ID, which must be unique. The instance ID is generated by the service center.

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique. During instance creation, use the microservice ID in the URL instead of the microservice ID specified here.
version	String	Microservice version.
hostName	String	Host information.
endpoints	Array of strings	Access address information.
status	String	Instance status. Value: UP, DOWN, STARTING, or OUTOFSERVICE. Default value: UP.
properties	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.
healthCheck	HealthCheck object	Health check information.
dataCenterInfo	DataCenterInfo object	Data center information.
timestamp	String	Time when an instance is created, which is automatically generated.
modTimestamp	String	Update time.

Table 8-91 HealthCheck

Parameter	Type	Description
mode	String	Heartbeat mode. Value: push or pull.
port	Integer	Port.
interval	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Integer	Maximum retries.

Table 8-92 DataCenterInfo

Parameter	Type	Description
name	String	Region name.

Parameter	Type	Description
region	String	Region.
availableZone	String	AZ.

Status code: 400

Table 8-93 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-94 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query instance information about a microservice with specified **service_id**.

```
GET https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances
```

Example Response

Status code: 200

Successfully queried.

```
{
  "instances": [
    {
      "instanceId": "8540bb8b693c4ad1a7fb6a756c415244",
      "serviceId": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
      "endpoints": [
        "rest:127.0.0.1:8080"
      ],
      "hostName": "hostNameTest",
      "status": "UP",
      "properties": {
```

```
    "engineID": "30c263e5-2eac-4da1-9c72-5abb9ac94550",
    "engineName": "cse-fkln1-HA"
  },
  "healthCheck": {
    "mode": "push",
    "interval": 30,
    "times": 3
  },
  "timestamp": "1650545035",
  "modTimestamp": "1650545035",
  "version": "1.0.0"
}
]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.3 Deregistering a Microservice Instance

Function

This API is used to deregister an instance based on **instance_id**.

URI

DELETE /v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}

Table 8-95 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Parameter	Mandatory	Type	Description
service_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^[A-Za-z0-9_-]*\$</code> . See Querying a Microservice Instance Based on service_id .
instance_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: <code>^[A-Za-z0-9_-]*\$</code> . See Querying a Microservice Instance Based on service_id .

Request

Table 8-96 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: <code>Authorization:Bearer {Token}</code></p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 400

Table 8-97 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-98 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Deregister an instance with specified **instance_id**.

```
DELETE https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.4 Querying Details About a Microservice Instance

Function

This API is used to query details about an instance based on **service_id** and **instance_id** after the instance is registered.

URI

GET /v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}

Table 8-99 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .
instance_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^[A-Za-z0-9_-]*\$. See Querying a Microservice Instance Based on service_id .

Table 8-100 Query parameter

Parameter	Mandatory	Type	Description
tags	No	String	Tag. When there are multiple tags, separate them using commas (.). Regular expression: ^[a-zA-Z][a-zA-Z0-9_-]{0,63}\$

Request

Table 8-101 Request header parameters

Parameter	Mandatory	Type	Description
X-consumerId	No	String	Microservice consumer ID, which must be unique.
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-102 Response body parameter

Parameter	Type	Description
instance	MicroServiceInstance object	Microservice instance information.

Table 8-103 MicroServiceInstance

Parameter	Type	Description
instanceId	String	Instance ID, which must be unique. The instance ID is generated by the service center.

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique. During instance creation, use the microservice ID in the URL instead of the microservice ID specified here.
version	String	Microservice version.
hostName	String	Host information.
endpoints	Array of strings	Access address information.
status	String	Instance status. Value: UP, DOWN, STARTING, or OUTFSERVICE. Default value: UP.
properties	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.
healthCheck	HealthCheck object	Health check information.
dataCenterInfo	DataCenterInfo object	Data center information.
timestamp	String	Time when an instance is created, which is automatically generated.
modTimestamp	String	Update time.

Table 8-104 HealthCheck

Parameter	Type	Description
mode	String	Heartbeat mode. Value: push or pull.
port	Integer	Port.
interval	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Integer	Maximum retries.

Table 8-105 DataCenterInfo

Parameter	Type	Description
name	String	Region name.

Parameter	Type	Description
region	String	Region.
availableZone	String	AZ.

Status code: 400

Table 8-106 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-107 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query details about an instance with specified **service_id** and **instance_id**.

```
GET https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}
```

Example Response

Status code: 200

Successfully queried.

```
{
  "instance": {
    "instanceId": "4994929d6b8044b29462d4f7daa33c4f",
    "serviceId": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
    "endpoints": [
      "rest:127.0.0.1:8080"
    ],
    "hostName": "hostNameTest",
    "status": "UP",
    "properties": {
      "engineID": "30c263e5-2eac-4da1-9c72-5abb9ac94550",
    }
  }
}
```

```
    "engineName": "cse-fkln1-HA"
  },
  "healthCheck": {
    "mode": "push",
    "interval": 30,
    "times": 3
  },
  "timestamp": "1650545270",
  "modTimestamp": "1650545270",
  "version": "1.0.0"
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.5 Modifying the Extended Information About a Microservice Instance

Function

This API is used to add or update the extended information about a microservice instance based on **instance_id** after the instance is registered.

Restrictions

None

URI

PUT /v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/properties

Table 8-108 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .
instance_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^[A-Za-z0-9_.-]*\$. See Querying a Microservice Instance Based on service_id .

Request

Table 8-109 Request header parameters

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>
X-consumerId	No	String	Microservice consumer ID, which must be unique.

Table 8-110 Request body parameter

Parameter	Mandatory	Type	Description
properties	No	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Response

Status code: 400

Table 8-111 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-112 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Update the extended information of an instance with specified **instance_id** to **"attr1": "b"**.

```
PUT https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/properties
{
  "properties": {
    "attr1": "b"
  }
}
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.6 Modifying Status of a Microservice Instance

Function

This API is used to update the status of a microservice instance based on **instance_id** after the instance is registered.

Restrictions

The TESTING state is supported only in version 2.3.X.

URI

PUT /v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/status

Table 8-113 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .
instance_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^[A-Za-z0-9_-.]*\$. See Querying a Microservice Instance Based on service_id .

Table 8-114 Query parameter

Parameter	Mandatory	Type	Description
value	Yes	String	Instance status. UP: online; OUTOFSERVICE: off-hook; STARTING: starting; DOWN: offline; TESTING: dialing test.

Request

Table 8-115 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 400

Table 8-116 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-117 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Update the status of an instance with specified **instance_id** to **UP**.

```
PUT https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/status?
value=UP
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.7 Sending Heartbeat Information

Function

This API is used to send heartbeat information. Service providers need to send heartbeat information to the service center, so that the center can check whether service instances are normal.

Restrictions

None

URI

```
PUT /v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/
heartbeat
```

Table 8-118 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
service_id	Yes	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$. See Querying Information About All Microservices .
instance_id	Yes	String	Microservice instance ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^[A-Za-z0-9_.-]*\$. See Querying a Microservice Instance Based on service_id .

Request

Table 8-119 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 400

Table 8-120 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-121 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Send heartbeat information to check whether an instance with specified **instance_id** is healthy.

```
PUT https://{endpoint}/v4/{project_id}/registry/microservices/{service_id}/instances/{instance_id}/heartbeat
```

Example Response

None

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.8 Querying a Microservice Instance by Filter Criteria

Function

This API is used to query a microservice instance based on microservice field filter criteria after the instance is registered.

URI

```
GET /v4/{project_id}/registry/instances
```

Table 8-122 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Table 8-123 Query parameters

Parameter	Mandatory	Type	Description
appld	Yes	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
serviceName	Yes	String	Microservice name The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-][a-zA-Z0-9]\$</code>
tags	No	String	Tag. When there are multiple tags, separate them using commas (.). Regular expression: <code>^[a-zA-Z][a-zA-Z0-9_-]{0,63}\$</code>
env	No	String	Environment information about the instance.

Request

Table 8-124 Request header parameters

Parameter	Mandatory	Type	Description
X-ConsumerId	No	String	Microservice consumer ID, which must be unique.
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-125 Response body parameter

Parameter	Type	Description
instances	Array of MicroServiceInstance objects	Instance list.

Table 8-126 MicroServiceInstance

Parameter	Type	Description
instanceId	String	Instance ID, which must be unique. The instance ID is generated by the service center.
serviceId	String	Microservice ID, which must be unique. During instance creation, use the microservice ID in the URL instead of the microservice ID specified here.
version	String	Microservice version.
hostName	String	Host information.
endpoints	Array of strings	Access address information.
status	String	Instance status. Value: UP, DOWN, STARTING, or OUTOFSERVICE. Default value: UP.
properties	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.
healthCheck	HealthCheck object	Health check information.
dataCenterInfo	DataCenterInfo object	Data center information.
timestamp	String	Time when an instance is created, which is automatically generated.
modTimestamp	String	Update time.

Table 8-127 HealthCheck

Parameter	Type	Description
mode	String	Heartbeat mode. Value: push or pull.
port	Integer	Port.
interval	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Integer	Maximum retries.

Table 8-128 DataCenterInfo

Parameter	Type	Description
name	String	Region name.
region	String	Region.
availableZone	String	AZ.

Status code: 400

Table 8-129 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-130 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query the instance of the microservice whose application ID is **default** and microservice name is **test**.

```
GET https://{endpoint}/v4/{project_id}/registry/instances?appId=default&serviceName=test
```

Example Response

Status code: 200

Successfully queried.

```
{
  "instances": [
    {
      "instanceId": "4098483294bb42d8b3c27eee0c166c1c",
      "serviceId": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
      "endpoints": [
        "rest:127.0.0.1:8080"
      ],
      "hostName": "hostNameTest",
      "status": "UP",
      "properties": {
        "engineID": "30c263e5-2eac-4da1-9c72-5abb9ac94550",
        "engineName": "cse-fkln1-HA"
      },
      "healthCheck": {
        "mode": "push",
        "interval": 30,
        "times": 3
      },
      "timestamp": "1650545431",
      "modTimestamp": "1650545442",
      "version": "1.0.0"
    }
  ]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.5.9 Querying Microservice Instances in Batches

Function

This API is used to query microservice instances in batches based on microservice field filter criteria after the instances are registered.

URI

POST /v4/{project_id}/registry/instances/action

Table 8-131 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.

Table 8-132 Query parameter

Parameter	Mandatory	Type	Description
type	Yes	String	Operation type. Currently, only query is supported.

Request

Table 8-133 Request header parameters

Parameter	Mandatory	Type	Description
X-consumerId	No	String	Microservice consumer ID, which must be unique.

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} </p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-134 Request body parameters

Parameter	Mandatory	Type	Description
services	At least one of the two parameters must be specified.	Array of FindService objects	Structure for querying service information.
instances		Array of FindInstance objects	Structure for querying instance information.

Table 8-135 FindService

Parameter	Mandatory	Type	Description
service	Yes	Dependency Key object	Dependency item.

Parameter	Mandatory	Type	Description
rev	No	String	<p>Version number of the client cache, which determines the difference between the local cache and the microservice in the service center.</p> <ul style="list-style-type: none">• Input 0 for the first time.• If the input rev is the same as that of the current microservice, notModified is returned. If they are different, updated is returned.• If rev is required in the next request, use the value of rev in updated in the response body.

Table 8-136 DependencyKey

Parameter	Mandatory	Type	Description
environment	Yes	String	Environment. Value: development, testing, acceptance, or production.
appld	Yes	String	Application ID, which must be unique.
serviceName	Yes	String	Microservice name

Table 8-137 FindInstance

Parameter	Mandatory	Type	Description
instance	No	HeartbeatSet Element object	Request struct of a microservice instance.

Parameter	Mandatory	Type	Description
rev	No	String	Version number cached by the client. Version number of the client cache, which determines the difference between the local cache and the microservice instance in the service center. <ul style="list-style-type: none">• Input 0 for the first time.• If the input rev is the same as that of the current microservice instance, notModified is returned. If they are different, updated is returned.• If rev is required in the next request, use the value of rev in updated in the response body.

Table 8-138 HeartbeatSetElement

Parameter	Mandatory	Type	Description
serviceId	No	String	Microservice ID.
instanceId	No	String	Microservice instance ID.

Response

Status code: 200

Table 8-139 Response body parameters

Parameter	Type	Description
services	BatchFindResult object	Batch query result structure.
instances	BatchFindResult object	Batch query result structure.

Table 8-140 BatchFindResult

Parameter	Type	Description
failed	Array of FindFailedResult objects	Query failure result structure.

Parameter	Type	Description
notModified	Array of integers	Index set corresponding to the request array.
updated	Array of FindResult objects	Query result structure list.

Table 8-141 FindFailedResult

Parameter	Type	Description
indexes	Array of integers	Index set corresponding to the request array.
error	Error object	Error struct.

Table 8-142 Error

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Table 8-143 FindResult

Parameter	Type	Description
index	Integer	Index corresponding to the request array.
rev	String	Versions returned by the server. If the versions are the same as the versions cached on the client, the instances parameter is left empty.
instances	Array of MicroServiceInstance objects	Instance list.

Table 8-144 MicroServiceInstance

Parameter	Type	Description
instanceId	String	Instance ID, which must be unique. The instance ID is generated by the service center.
serviceId	String	Microservice ID, which must be unique. During instance creation, use the microservice ID in the URL instead of the microservice ID specified here.
version	String	Microservice version.
hostName	String	Host information.
endpoints	Array of strings	Access address information.
status	String	Instance status. Value: UP, DOWN, STARTING, or OUTFSERVICE. Default value: UP.
properties	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.
healthCheck	HealthCheck object	Health check information.
dataCenterInfo	DataCenterInfo object	Data center information.
timestamp	String	Time when an instance is created, which is automatically generated.
modTimestamp	String	Update time.

Table 8-145 HealthCheck

Parameter	Type	Description
mode	String	Heartbeat mode. Value: push or pull.
port	Integer	Port.
interval	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s.
times	Integer	Maximum retries.

Table 8-146 DataCenterInfo

Parameter	Type	Description
name	String	Region name.
region	String	Region.
availableZone	String	AZ.

Status code: 400

Table 8-147 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-148 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query all instances of the microservice whose application ID is **default** and microservice name is **test**.

```
POST https://{endpoint}/v4/{project_id}/registry/instances/action?type=query
```

```
{
  "services": [
    {
      "service": {
        "environment": "",
        "appld": "default",
        "serviceName": "test"
      },
      "rev": "0"
    }
  ]
}
```


Example Response

Status code: 200

Successfully queried.

```
{
  "services": {
    "updated": [
      {
        "index": 0,
        "rev": "0feb784798bca7b2fb4de8351578c4437b516c4b",
        "instances": [
          {
            "instanceId": "79cdaf47cacf43a5b2b4185527da2255",
            "serviceId": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
            "endpoints": [
              "rest:127.0.0.1:8080"
            ],
            "hostName": "hostNameTest",
            "status": "UP",
            "properties": {
              "engineID": "30c263e5-2eac-4da1-9c72-5abb9ac94550",
              "engineName": "cse-fkln1-HA"
            },
            "healthCheck": {
              "mode": "push",
              "interval": 30,
              "times": 3
            },
            "timestamp": "1650546183",
            "modTimestamp": "1650546183",
            "version": "1.0.0"
          }
        ]
      }
    ]
  }
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.6 Dependency

8.6.1 Querying All Providers of a Microservice

Function

This API is used to query all providers of a microservice based on **consumer_id**. For example, if service A depends on services B and C, you can query the information about services B and C based on service A ID.

URI

GET /v4/{project_id}/registry/microservices/{consumer_id}/providers

Table 8-149 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none">For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. See . You can also enter the fixed value default.For an exclusive microservice engine, enter the fixed value default.
consumer_id	Yes	String	Consumer service ID.

Table 8-150 Query parameters

Parameter	Mandatory	Type	Description
noSelf	No	Integer	Whether to return the self-dependency relationship. 0: no; 1: yes.
sameDomain	No	Integer	Whether to return the shared-service relationship. 0: no; 1: yes.

Request

Table 8-151 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-152 Response body parameter

Parameter	Type	Description
providers	MicroService object	Microservice information.

Table 8-153 Microservice

Parameter	Type	Description
serviceId	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: ^.*\$
environment	String	Microservice environment. Value: development, testing, acceptance, or production. You can use the API for uploading schemas in batches to add or modify an existing schema only when the value is development , testing , or acceptance . Default value: development.

Parameter	Type	Description
appld	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
serviceName	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]\$ ^[a-zA-Z0-9][a-zA-Z0-9_-.][a-zA-Z0-9]\$</code>
version	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]\$ ^[0-9]+([0-9]+)\$</code>
description	String	Microservice description. The value contains a maximum of 256 characters.
level	String	Microservice level. Value: FRONT, MIDDLE, or BACK.
registerBy	String	Microservice registration mode. Value: SDK, PLATFORM, SIDECAR, or UNKNOWN.
schemas	Array of strings	Microservice schema content. The value must be 1 to 160 bytes long. Only digits, letters, underscores (<code>_</code>), hyphens (<code>-</code>), and periods (<code>.</code>) are allowed. An array contains a maximum of 100 schemas.
status	String	Microservice status. Value: UP or DOWN. Default value: UP.
timestamp	String	Microservice registration time.
modTimestamp	String	Latest modification time (UTC).
framework	Framework object	Development framework.
paths	Array of ServicePath objects	Service path.
properties	Object	Extended attribute. You can customize a key and value.

Table 8-154 Framework

Parameter	Type	Description
name	String	Microservice development framework. Default value: UNKNOWN.
version	String	Version of the microservice development framework.

Table 8-155 ServicePath

Parameter	Type	Description
Path	String	Route address.
Property	Object	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

Status code: 400

Table 8-156 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-157 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Query the providers of a consumer service with specified **consumer_id**.

```
GET https://{endpoint}/v4/{project_id}/registry/microservices/{consumer_id}/providers
```

Example Response

Status code: 200

Successfully queried. **serviceld** or **schemald** is returned.

```
{
  "providers": [
    {
      "serviceld": "8aed80ea052ac04a64dfc79c24f2170224d074f5",
      "appld": "default",
      "serviceName": "test",
      "version": "1.0.0",
      "description": "this is a test",
      "level": "BACK",
      "status": "UP",
      "properties": {
        "test": "test"
      },
      "timestamp": "1650543950",
      "modTimestamp": "1650544411"
    }
  ]
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7 Configuration Management

8.7.1 Creating a Configuration

Function

This API is used to create a configuration.

URI

POST /v1/{project_id}/kie/kv

Table 8-158 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.

Request

Table 8-159 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-160 Request body parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Configuration key.
value	No	String	Configuration value.
labels	No	Map<String,String>	Configuration labels.
status	No	String	Configuration status.
value_type	No	String	Type of the configuration value.

Response

Status code: 200

Table 8-161 Response body parameter

Parameter	Type	Description
kie	SingleKieInfo	Configuration information.

Table 8-162 SingleKieInfo

Parameter	Type	Description
id	String	Configuration ID.
key	String	Configuration key.
value	String	Configuration value.
value_type	String	Type of the configuration value.
status	String	Configuration status.
create_time	String	Time when the configuration is created.
update_time	String	Time when the configuration is updated.
labels	Map<String,String>	Configuration labels.
create_version	Integer	Created version number of the configuration.
update_version	Integer	Updated version number of the configuration.

Status code: 400

Table 8-163 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 500

Table 8-164 Response body parameters

Parameter	Type	Description
error_code	String	Error code.

Parameter	Type	Description
error_message	String	Error message.

Example Request

Create a configuration whose label is **"key1": "value1"** and **"key2": "value2"**.

```
POST https://{endpoint}/v1/{project_id}/kie/kv
{
  "key": "String",
  "value": "String",
  "labels": {
    "key1": "value1",
    "key2": "value2",
  },
  "status": "String",
  "value_type": "String"
}
```

Example Response

Status code: 200

Successfully created. The configuration information is returned.

```
{
  "id": "8a9e6a5d-8d65-48fb-a40c-5150c8479da8",
  "key": "string",
  "labels": {
    "key1": "value1",
    "key2": "value2"
  },
  "value": "string",
  "value_type": "text",
  "status": "enabled",
  "create_time": 1623139038,
  "update_time": 1623139038,
  "create_revision": 13,
  "update_revision": 13,
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
409	Conflict
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7.2 Modifying a Configuration

Function

This API is used to modify a configuration.

URI

PUT /v1/{project_id}/kie/kv/{kv_id}

Table 8-165 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.
kv_id	Yes	String	ID of the configuration to be modified. See Querying the Configuration List .

Request

Table 8-166 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-167 Request body parameters

Parameter	Mandatory	Type	Description
value	Yes	String	Configuration value.
status	No	String	Configuration status.

Response

Status code: 200

Table 8-168 Response body parameter

Parameter	Type	Description
kie	SingleKieInfo	Configuration information.

Status code: 400

Table 8-169 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 500

Table 8-170 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Example Request

Modify the value of the configuration whose ID is **8a9e6a5d-8d65-48fb-a40c-5150c8479da8** to **this is a test** and set the configuration status to **enabled**.

```
PUT https://{endpoint}/v1/{project_id}/kie/kv/8a9e6a5d-8d65-48fb-a40c-5150c8479da8
{
  "value": "this is a test",
```

```
"status": "enabled"
}
```

Example Response

Status code: 200

Successfully modified. The configuration information is returned.

```
{
  "id": "8a9e6a5d-8d65-48fb-a40c-5150c8479da8",
  "key": "string",
  "labels": {
    "key1": "value1",
    "key2": "value2"
  },
  "value": "this is a test",
  "value_type": "text",
  "status": "enabled",
  "create_time": 1623139038,
  "update_time": 1623139038,
  "create_revision": 13,
  "update_revision": 13,
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7.3 Querying a Configuration

Function

This API is used to query a configuration in the configuration center.

URI

GET /v1/{project_id}/kie/kv/{kv_id}

Table 8-171 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.

Parameter	Mandatory	Type	Description
kv_id	Yes	String	ID of the configuration to be queried. See Querying the Configuration List .

Request

Table 8-172 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

Status code: 200

Table 8-173 Response body parameter

Parameter	Type	Description
id	String	Configuration ID.
key	String	Configuration key.
value	String	Configuration value.
value_type	String	Type of the configuration value.
status	String	Configuration status.
create_time	String	Time when the configuration is created.

Parameter	Type	Description
update_time	String	Time when the configuration is updated.
labels	Map<String,String>	Configuration labels.
create_revision	Integer	Created version number of the configuration.
update_revision	Integer	Updated version number of the configuration.

Status code: 400

Table 8-174 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 404

Table 8-175 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 500

Table 8-176 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Example Request

Query the configuration whose configuration ID is **8a9e6a5d-8d65-48fb-a40c-5150c847**.

```
GET https://{endpoint}/v1/{project_id}/kie/kv/8a9e6a5d-8d65-48fb-a40c-5150c8479da8
```

Example Response

Status code: 200.

Successfully queried.

```
{
  "id": "8a9e6a5d-8d65-48fb-a40c-5150c8479da8",
  "key": "string",
  "labels": {
    "key1": "value1",
    "key2": "value2"
  },
  "value": "string",
  "value_type": "text",
  "status": "enabled",
  "create_time": 1623139038,
  "update_time": 1623139038,
  "create_revision": 13,
  "update_revision": 13
}
```

Status Code

Status Code	Description
200	OK
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7.4 Querying the Configuration List

Function

This API is used to query the configuration list in the configuration center.

URI

```
GET /v1/{project_id}/kie/kv
```

Table 8-177 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.

Request

Table 8-178 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 8-179 Query parameters

Parameter	Mandatory	Type	Description
label	No	String	Exports data filtered by label. The format is {Label key}: {Label value}.
match	No	String	Matching option of the filtered items. Value exact indicates exact matching, including the same number of labels. If the value is null, inclusive matching is used.

Parameter	Mandatory	Type	Description
revision	No	Integer	<p>Configuration version number</p> <p>This parameter is used to determine the difference between the local cache configuration and the configuration in the configuration center.</p> <ul style="list-style-type: none">• Input 0 for the first time.• If the input revision value is the same as the current revision value, status code 304 is returned, but no body is returned. If they are inconsistent, both revision values are returned.• If revision is required in the next request, use the value of update_revision in the response body.

Response

Status code: 200

Table 8-180 Response body parameters

Parameter	Type	Description
total	Integer	Number of query results.
data	List< SingleKielInfo >	List of queried kie.

Table 8-181 SingleKielInfo

Parameter	Type	Description
id	String	Configuration ID.
key	String	Configuration key.
value	String	Configuration value.
value_type	String	Type of the configuration value.
status	String	Configuration status.
create_time	String	Time when the configuration is created.

Parameter	Type	Description
update_time	String	Time when the configuration is updated.
labels	Map<String,String>	Configuration labels.
create_revision	Integer	Created version number of the configuration.
update_revision	Integer	Updated version number of the configuration.

Status code: 304

None

Status code: 400

Table 8-182 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 500

Table 8-183 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Example Request

Query the configuration list.

```
GET https://{endpoint}/v1/{project_id}/kie/kv
```

Example Response

Status code: 200

Successfully queried.

```
{
  "total": 1,
  "data": [{
    "id": "8a9e6a5d-8d65-48fb-a40c-5150c8479da8",
    "key": "string",
    "labels": {
      "key1": "value1",
      "key2": "value2"
    },
    "value": "string",
    "value_type": "text",
    "status": "enabled",
    "create_time": 1623139038,
    "update_time": 1623139038,
    "create_revision": 13,
    "update_revision": 13
  }]
}
```

Status Code

Status Code	Description
200	OK
304	Not Modified
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7.5 Deleting a Configuration

Function

This API is used to delete a configuration.

URI

DELETE /v1/{project_id}/kie/kv/{kv_id}

Table 8-184 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.
kv_id	Yes	String	ID of the configuration to be deleted. See Querying the Configuration List .

Request

Table 8-185 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required. The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine .

Response

Status code: 200

Successfully deleted.

Status code: 404

Table 8-186 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Status code: 400

Table 8-187 Response body parameters

Parameter	Type	Description
error_code	String	Error code.

Parameter	Type	Description
error_message	String	Error message.

Status code: 500

Table 8-188 Response body parameters

Parameter	Type	Description
error_code	String	Error code.
error_message	String	Error message.

Example Request

Delete the configuration whose configuration ID is **8a9e6a5d-8d65-48fb-a40c-5150c8479da8**.

```
DELETE https://{endpoint}/v1/{project_id}/kie/kv/8a9e6a5d-8d65-48fb-a40c-5150c8479da8
```

Example Response

Status code: 204

Successfully deleted.

Status Code

Status Code	Description
200	No Content
400	Bad Request
404	Not Found
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

8.7.6 Deleting All Configurations

Function

This API is used to delete all configurations.

URI

DELETE /v1/{project_id}/kie/kv

Table 8-189 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.

Request

Table 8-190 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	No	String	This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required. The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine .

Table 8-191 Request body parameter

Parameter	Mandatory	Type	Description
ids	Yes	List<String>	ID list of the configurations to be deleted. See Querying the Configuration List .

Response

Status code: 204

Successfully deleted.

Status code: 404

Table 8-192 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 8-193 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

Delete all configurations.

```
DELETE https://{endpoint}/v1/{project_id}/kie/kv
```

```
{
  "ids":[
    "id1",
    "id2"
  ]
}
```

Example Response

Status code: 204

Successfully deleted.

Status Code

Status Code	Description
204	No Content
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceComb Error Codes](#).

9 Out-of-Date APIs

9.1 Querying Configurations

Function

This API is used to query configurations in the configuration center.

Call Method

- For a professional microservice engine, see [Calling APIs](#).
- For an exclusive microservice engine, perform the following steps:
 - a. Log in to ServiceStage and choose **Cloud Service Engine > Engines**.
 - b. Select the microservice engine of the API to be called from the **Microservice Engine** drop-down list in the upper part of the page.
 - c. In the **Service Discovery and Configuration** area, view or click to copy the configuration center address.
 - d. Access the API by referring to [Calling APIs](#). In **Request URI**, replace **{Endpoint}** with the obtained configuration center address.

URI

GET /v3/{project_id}/configuration/items

Table 9-1 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	<ul style="list-style-type: none"> For a professional microservice engine, enter the tenant project ID. The value contains 1 to 64 characters. You can also enter the fixed value default. For an exclusive microservice engine, enter the fixed value default.

Table 9-2 Query parameters

Parameter	Mandatory	Type	Description
dimensionsInfo	Yes	String	<p>Service information, which consists of the service name (mandatory), application (mandatory), and version (optional). The combination format is as follows (note that @ and # must be escaped when being added to the HTTP URL. @ is escaped to %40, and # is escaped to %23):</p> <p>{serviceName}@{appId}#{version} or {serviceName}@{appId}</p> <p>The value cannot contain spaces or the following special characters: \$%^&+/\</p>
keyFilter	No	String	<p>key filter criteria. A regular expression is supported. If special characters are contained, escape them.</p>
revision	No	String	<p>Version of a configuration item. This parameter is used to determine the difference between the local cache configuration and the configuration in the configuration center.</p> <ul style="list-style-type: none"> Input 0 for the first time. If the input revision value is the same as the current revision value, status code 304 is returned, but no body is returned. If they are inconsistent, both revision values are returned. If revision is required in the next request, use the value of version in revision in the returned body.

Request

Table 9-3 Request header parameter

Parameter	Mandatory	Type	Description
x-environment	No	String	Environment whose configurations need to be queried. Value: development, testing, acceptance, or production. Other values are regarded as null.
Authorization	Yes	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Response

If the input revision value is the same as the current revision value, no response message is returned.

[Table 9-4](#) describes the response parameters.

Table 9-4 Response parameters

Parameter	Type	Description
{dimension} (The parameter name and number of parameters are not fixed.)	Map<String, String>	Configuration content.
revision	Table 9-5	Revision.

Table 9-5 revision

Parameter	Type	Description
version	String	Version.

Example Request

```
GET /v3/default/configuration/items?dimensionsInfo=service%40app%231.0.0
```

Example Response

```
{
  "service@app": {
    "k1": "v1"
  },
  "service@app#1.0.0": {
    "k2": "v2"
  }
}
```

Status Code

See [Status Codes](#).

Error Code

See [ServiceStage Error Codes](#).

9.2 Creating a Dependency Between Services

Function

This API is used to create a dependency between services. For **consumer**, **version** must be a confirmed version and **serviceName** cannot be *. **consumer** must be an existing service while **provider** can be a service that has not been created.

Call Method

See "Microservice Management" in [API Calling](#).

URI

```
PUT /v4/{project_id}/registry/dependencies
```

Table 9-6 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Fixed value: default.

Request

Table 9-7 Request header parameter

Parameter	Mandatory	Type	Description
Authorization	Yes	String	<p>This parameter is mandatory if security authentication is enabled for an exclusive microservice engine. Otherwise, this parameter is not required.</p> <p>The token of an exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token}</p> <p>For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine.</p>

Table 9-8 Request body parameters

Parameter	Mandatory	Type	Description
dependencies	Yes	Array of objects	Set of rules. See Table 9-9 . The array can contain 1–100 objects.

Table 9-9 MicroServiceDependency parameters

Parameter	Mandatory	Type	Description
consumer	Yes	Object	Consumer microservice. See DependencyMicroService .
providers	No	Array of objects	Provider microservice. See DependencyMicroService .

Response

None

Example Request

```
{
  "dependencies": [
    {
      "consumer": {
        "environment": "",
        "appld": "default",
```

```
    "serviceName": "test1",  
    "version": "1.0.0"  
  },  
  "providers": [  
    {  
      "environment": "",  
      "appld": "default",  
      "serviceName": "test",  
      "version": "1.0.0"  
    }  
  ]  
}
```

Example Response

None

Status Code

See [Status Codes](#).

Error Code

See [ServiceStage Error Codes](#).

9.3 Reporting Service Metrics

Function

This API is used to report service metrics to the dashboard.

Call Method

See [Calling APIs](#).

URI

POST /v2/{project_id}/csemonitor/metric

Table 9-10 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which must be unique. The value contains 1 to 64 characters.

Request

Table 9-11 Request header parameter

Parameter	Mandatory	Type	Description
x-domain-name	Yes	String	Tenant account name.
X-Auth-Token	Yes	String	User token.

Table 9-12 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Name.
appld	No	String	Application ID.
serviceId	No	String	Service ID.
version	No	String	Version.
instanceId	No	String	Instance ID.
instance	No	String	Instance information.
thread	No	Integer	Thread name.
cpu	No	Double	CPU information.
memory	No	Map<String, Long>	Memory information.
interfaces	No	Array of FunctionMetricInfo objects	Monitoring item list.

Table 9-13 FunctionMetricInfo

Parameter	Mandatory	Type	Description
functionName	No	String	Name of the monitoring item.
functionType	No	String	Monitoring item type.
desc	No	String	Monitoring item description.
time	No	Long	Time.
appld	No	String	Application ID.
version	No	String	Version.

Parameter	Mandatory	Type	Description
qps	No	Double	Throughput.
latency	No	Double	Latency.
rate	No	Double	Percentage.
failureRate	No	Double	Failure percentage.
shortCircuited	No	Integer	Number of circuits.
semaphoreRejected	No	Integer	Number of rejected semaphores.
threadPoolRejected	No	Integer	Number of rejected threads.
countTimeout	No	Integer	Timeout duration.
l995	No	Double	99.5% requests have a value smaller than the value of this parameter.
l99	No	Double	99% requests have a value smaller than the value of this parameter.
l90	No	Double	90% requests have a value smaller than the value of this parameter.
l75	No	Double	75% requests have a value smaller than the value of this parameter.
l50	No	Double	50% requests have a value smaller than the value of this parameter.
l25	No	Double	25% requests have a value smaller than the value of this parameter.
l5	No	Double	5% requests have a value smaller than the value of this parameter.

Response

Status code: 400

Table 9-14 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Status code: 500

Table 9-15 Response body parameters

Parameter	Type	Description
errorCode	String	Error code.
errorMessage	String	Error message.
detail	String	Location details.

Example Request

POST https://{endpoint}/v2/{project_id}/csemonitor/metric

```
{
  "name" : "test",
  "appId" : "default",
  "version" : "1.0.0",
  "instance" : "hostNameTest",
  "thread" : 0,
  "cpu" : 0,
  "memory" : {
    "1": "1"
  },
  "interfaces" : [
    {
      "time" : 0,
      "name" : "test",
      "qps" : 0,
      "latency" : 0,
      "rate" : 0,
      "total" : 0,
      "isCircuitBreakerOpen" : true,
      "failure" : 0,
      "shortCircuited" : 0,
      "semaphoreRejected" : 0,
      "threadPoolRejected" : 0,
      "countTimeout" : 0,
      "l995" : 0,
      "l99" : 0,
      "l90" : 0,
      "l75" : 0,
      "l50" : 0,
      "l25" : 0,
      "l5" : 0
    }
  ]
}
```

Example Response

Status code: 400

```
{  
  "errorCode" : "string",  
  "errorMessage" : "string",  
  "detail" : "string"  
}
```

Status code: 500

```
{  
  "errorCode" : "string",  
  "errorMessage" : "string",  
  "detail" : "string"  
}
```

Status Code

Status Code	Description
200	The data is reported successfully.
400	Bad Request
500	Internal Server Error

Error Code

See [ServiceStage Error Codes](#).

10 Data Structure

10.1 Parameter Description

10.1.1 HealthCheck

Table 10-1 Parameters

Parameter	Mandatory	Type	Description
mode	Yes	String	Check mode. Value: push or pull.
port	No	Integer	Port number. The value can contain up to 65,536 characters. Regular expression: $^[0-9]*\$$.
interval	Yes	Integer	Heartbeat interval. Unit: s. If the value is less than 5s, the registration is performed at an interval of 5s. The value can contain up to 4,294,967,296 characters. Regular expression: $^[0-9]+\$$.
times	Yes	Integer	Maximum number of request attempts. The value can contain up to 4,294,967,296 characters. Regular expression: $^[0-9]+\$$.

10.1.2 MicroServiceInstance

Table 10-2 Parameters

Parameter	Mandatory	Type	Description
instanceId	No	String	Instance ID, which must be unique. The instance ID is generated by the service center.
serviceId	No	String	Microservice ID, which must be unique. The microservice ID in the URL will be used.
version	No	String	Microservice version.
hostName	Yes	String	PC name, which can be a domain name or IP address.
endpoints	No	Array of strings	Example: rest:127.0.0.1:8080.
status	Yes	String	Instance status. <ul style="list-style-type: none">• UP• DOWN• STARTING• OUTFSERVICE
properties	No	Map<String, String>	Extended attribute. You can customize a key and value.
healthCheck	No	Health check	Health check mode. Value: push or pull. interval indicates the pull interval (unit: s), and times indicates the maximum request attempts. When in the pull mode, you need to define port to ensure that the registration center actively connects to the port. The default value of healthCheck is push , that of interval is 30 , and that of times is 3 .
timestamp	No	String	Time when an instance is created, which is automatically generated.
modTimestamp	No	String	Update time.
dataCenterInfo	No	DataCenterInfo	Time zone.

10.1.3 HeartbeatSetElement

Table 10-3 Parameters

Parameter	Mandatory	Type	Description
serviceld	Yes	String	Microservice ID. The value contains 1 to 64 characters. Regular expression: $^.*\$$.
instanceId	Yes	String	Microservice instance ID. The value contains 1 to 64 characters. Regular expression: $^[A-Za-z0-9_-.]*\$$.

10.1.4 InstanceHbRst

Table 10-4 Parameters

Parameter	Mandatory	Type	Description
serviceld	No	String	Microservice ID.
instanceId	No	String	Microservice instance ID.
errMessage	No	String	Error information. If the operation is successful, the value is empty. If the operation fails, the error information is returned.

10.1.5 DelServicesResponse

Table 10-5 Parameter

Parameter	Mandatory	Type	Description
serviceld	No	String	Microservice ID.

10.1.6 MetricData

Table 10-6 Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Service name.
appld	Yes	String	Application ID.

Parameter	Mandatory	Type	Description
serviceId	No	String	Service ID.
version	Yes	String	Service version.
instanceId	No	String	Instance ID.
instance	Yes	String	Instance name.
thread	No	String	Number of threads.
cpu	No	String	CPU usage.
memory	No	String	Memory.
interfaces	No	Table 10-7	API data details.

 NOTE

Parameters **serviceId** and **instanceId** must be specified or left blank at the same time.

Table 10-7 items parameters

Parameter	Mandatory	Type	Description
items	No	FunctionMetricInfo	API-level aggregation metric data.

10.1.7 FunctionMetricInfo

Table 10-8 Parameters

Parameter	Mandatory	Type	Description
AggregateMetricInfo	No	Object	Aggregation metric data.
l995	No	Integer	99.5% requests have a value smaller than the value of this parameter.
l99	No	Integer	99% requests have a value smaller than the value of this parameter.
l90	No	Integer	90% requests have a value smaller than the value of this parameter.
l75	No	Integer	75% requests have a value smaller than the value of this parameter.
l50	No	Integer	50% requests have a value smaller than the value of this parameter.

Parameter	Mandatory	Type	Description
l25	No	Integer	25% requests have a value smaller than the value of this parameter.
l5	No	Integer	5% requests have a value smaller than the value of this parameter.

10.1.8 InstanceMetricInfo

Table 10-9 Parameters

Parameter	Mandatory	Type	Description
AggregateMetricInfo	No	Object	Aggregation metric data.
thread	No	Integer	Number of threads.
cpu	No	number	CPU usage.
memory	No	Object	Memory information.

10.2 Common Request Parameters

10.2.1 MicroService

Table 10-10 Parameters

Parameter	Mandatory	Type	Description
serviceId	No	String	Microservice ID, which must be unique. The value contains 1 to 64 characters. Regular expression: $^{\wedge}.*\$.$
environment	No	String	Microservice environment. Value: Empty value, development, testing, acceptance, or production.
appId	No	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: $^{\wedge}[a-zA-Z0-9]*\$\wedge[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$.$

Parameter	Mandatory	Type	Description
serviceName	Yes	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\-\.]*[a-zA-Z0-9]\$</code> .
version	No	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$</code> .
description	No	String	Description of a microservice. The value can contain up to 256 characters.
level	No	String	Microservice level. <ul style="list-style-type: none">• FRONT• MIDDLE• BACK
timestamp	No	String	Microservice registration time.
schemas	No	Array of String	Foreign key ID of a microservice access schema. The array length supports up to 100 schemas.
status	No	String	Microservice status. Value: UP or DOWN. Default value: UP.
paths	No	ServicePath	Service path.
framework	No	Framework	Microservice development framework.
registerBy	No	String	Microservice registration mode. <ul style="list-style-type: none">• SDK• PLATFORM• SIDECAR• UNKNOWN
modTimestamp	No	String	Latest modification time (UTC).
properties	No	Map<String, String>	Extended attribute. You can customize a key and value.
alias	No	String	Microservice alias. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9_\-\.]*\$</code> .

Table 10-11 ServicePath parameters

Parameter	Mandatory	Type	Description
Path	No	String	Route address. The value contains 1 to 160 bytes, including digits, letters, and the following characters:.,?'\/+& %\$#=#~_-@{}
Property	No	Properties	Extended attribute.

Table 10-12 Framework parameters

Parameter	Mandatory	Type	Description
name	No	String	Microservice development framework. Default value: UNKNOWN.
version	No	String	Version of the microservice development framework. The value contains 1 to 64 bytes.

10.2.2 Properties

Table 10-13 Parameters

Parameter	Mandatory	Type	Description
< * >	No	String	Extended attribute. You can customize a key and value. The value must be at least 1 byte long.

10.2.3 DependencyMicroService

Table 10-14 Parameters

Parameter	Mandatory	Type	Description
appld	Yes	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]*\$</code> <code>^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code> .

Parameter	Mandatory	Type	Description
serviceName	Yes	String	Microservice name. For a provider microservice, the value can be *, indicating all services of a tenant. If the value is *, both appId and version can be omitted. For a consumer microservice, the value cannot be *. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]*\$</code> <code>^[a-zA-Z0-9][a-zA-Z0-9_\-\.]*[a-zA-Z0-9]\$</code> .
version	Yes	String	Version rule. This parameter is mandatory for a professional microservice engine. Otherwise, this parameter is not required. <ul style="list-style-type: none"> Exact version matching. Example: 0.0.1. Subsequent version matching. Example: 0.0.1+. Latest version. Example: latest. Version range. Example: 0.1.0-0.2.0.
environment	Yes	String	Microservice environment. <ul style="list-style-type: none"> Empty value development testing acceptance production

10.2.4 Rule

Table 10-15 Parameters

Parameter	Mandatory	Type	Description
ruleId	No	String	Blacklist or whitelist ID.
ruleType	Yes	String	Type. <ul style="list-style-type: none"> WHITE BLACK

Parameter	Mandatory	Type	Description
attribute	Yes	String	If the value starts with tag_XXX , the attributes are filtered by Tag . Otherwise, the attributes are filtered by ServiceId , Appld , ServiceName , Version , Description , Level , or Status .
pattern	Yes	String	Matching rule. The value is a regular expression containing 1 to 64 characters.
timestamp	No	String	Rule creation time. This parameter is used only when you obtain a rule.
description	No	String	Description of a rule. The value can contain up to 256 characters.
modTimestamp	No	String	Latest modification time (UTC).

10.2.5 AddOrUpdateRule

Table 10-16 Parameters

Parameter	Mandatory	Type	Description
ruleType	No	String	Type. <ul style="list-style-type: none">• WHITE• BLACK
attribute	No	String	If the value starts with tag_XXX , the attributes are filtered by Tag . Otherwise, the attributes are filtered by ServiceId , Appld , ServiceName , Version , Description , Level , or Status .
pattern	No	String	Matching rule. The value is a regular expression containing 1 to 64 characters.
description	No	String	Description of a rule. The value can contain up to 256 characters.

10.2.6 DataCenterInfo

Table 10-17 Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Data center name. The value must be 1 to 128 bytes long. Only digits, letters, underscores (_), hyphens (-), and periods (.) are allowed.
region	Yes	String	Region where the data center is located. The value must be 1 to 128 bytes long. Only digits, letters, underscores (_), hyphens (-), and periods (.) are allowed.
availableZone	Yes	String	AZ where the data center is located. The value must be 1 to 128 bytes long. Only digits, letters, underscores (_), hyphens (-), and periods (.) are allowed.

10.2.7 Schema

Table 10-18 Parameters

Parameter	Mandatory	Type	Description
schemald	No	String	Microservice schema ID. The value must be 1 to 160 bytes long. Only digits, letters, underscores (_), hyphens (-), and periods (.) are allowed.
schema	No	String	Microservice schema content. The value must be 1 to 2048 bytes long.
summary	No	String	Microservice schema summary. The value can contain up to 128 bytes. Only digits and letters are allowed.

10.3 Common Response Parameters

10.3.1 WatchMicroServiceKey

Table 10-19 Parameters

Parameter	Mandatory	Type	Description
appld	No	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code> .
serviceName	No	String	Microservice name. For a provider microservice, the value can be *, indicating all services of a tenant. If the value is *, both appld and version can be omitted. For a consumer microservice, the value cannot be *.
version	No	String	Microservice version. For a provider microservice, the value can be the version+, fixed version, or latest version. For example, 1.0.1+ indicates version 1.0.1 or later. For a consumer microservice, the value can only be a fixed version.

10.3.2 SuccessdResponse

Table 10-20 Parameters

Parameter	Mandatory	Type	Description
Result	Yes	String	Returned result.

10.3.3 ServiceInfo

Table 10-21 Parameters

Parameter	Mandatory	Type	Description
serviceId	Yes	String	Microservice ID, which must be unique. The value is a UUID. Regular expression: <code>^\.*\$</code> .

Parameter	Mandatory	Type	Description
appld	Yes	String	Application ID, which must be unique. The value contains 1 to 160 characters. Regular expression: <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code> .
serviceName	Yes	String	Microservice name, which must be unique in an application. The value contains 1 to 128 characters. Regular expression: <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code> .
version	Yes	String	Microservice version. The value contains 1 to 64 characters. Regular expression: <code>^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$</code> .
level	Yes	String	Microservice level. <ul style="list-style-type: none">• FRONT• MIDDLE• BACK
status	Yes	String	Service status. <ul style="list-style-type: none">• UP• DOWN
timestamp	Yes	String	Timestamp.

10.3.4 AggregateMetricInfo

Table 10-22 Parameters

Parameter	Mandatory	Type	Description
time	No	Integer	Report time.
name	No	String	API name.
qps	No	Number	Current Query Per Second (QPS).
latency	No	Number	Average latency.
rate	No	Integer	Success rate.
total	No	Integer	Total number of requests.
isCircuitBreakerOpen	No	Boolean	Whether the circuit breaker is enabled.
failure	No	Integer	Total number of failed requests.

Parameter	Mandatory	Type	Description
shortCircuited	No	Integer	Total number of short circuits.
semaphoreRejected	No	Integer	Total number of rejected semaphores.
threadPoolRejected	No	Integer	Total number of rejected threads.
countTimeout	No	Integer	Total number of timeout requests.

10.3.5 ServiceDetail

Table 10-23 Parameters

Parameter	Mandatory	Type	Description
microService	No	MicroService	Service information.
instances	No	MicroServiceInstance	Instance information.
schemaInfos	No	Schema	Microservice API details.
rules	No	Rule	Blacklist and whitelist information.
providers	No	MicroService	All provider information.
consumers	No	MicroService	All consumer information.
tags	No	Tags	All tag information about a microservice.
microServiceVersions	No	Array. Each item in the array is a string.	All microservice versions.

Table 10-24 tags parameters

Parameter	Mandatory	Type	Description
tags	No	Properties	Request struct of an extended attribute of a microservice.

10.3.6 TenantProject

Table 10-25 TenantProject parameters

Parameter	Located In	Type	Description
tenant	body	String	Tenant name.
projectId	body	String	Project ID.

10.3.7 Error

Table 10-26 Parameters

Parameter	Mandatory	Type	Description
errorCode	Yes	String	Error code.
errorMessage	Yes	String	Error message.
detail	No	String	Location details.

11 Permissions and Supported Actions

11.1 Introduction

This chapter describes fine-grained permissions management for your ServiceStage. If your Huawei Cloud account does not need individual Identity and Access Management (IAM) users, then you may skip over this chapter.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant users permissions using [roles](#) and [policies](#). Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is recommended if you want to allow or deny the access to an API.

Your account has all the permissions required to call all APIs, but IAM users under your account must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query metrics using an API, the user must have been granted permissions that allow the **servicestage:app:get** action.

Supported Actions

There are two kinds of policies: system-defined policies and custom policies. If the permissions preset in the system do not meet your requirements, you can create custom policies and apply these policies to user groups for refined access control. Operations supported by policies are specific to APIs. The following lists common concepts related to policies:

- Permissions: Defined by actions in a custom policy.
- APIs: REST APIs that can be called in a custom policy.
- Actions: specific operations that are allowed or denied.
- IAM projects/Enterprise projects: the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that contain actions only for IAM projects can be used and applied to IAM only. Administrators can check whether an action supports IAM projects or enterprise projects in the action list. For details about the differences between IAM and enterprise projects, see [What Are the Differences Between IAM and Enterprise Management?](#)

ServiceStage supports the following actions in custom policies:

Table 11-1 ServiceStage actions

Permissions	APIs	Actions	IAM Projects	Enterprise Projects
Creating an application	POST /v2/{project_id}/cas/applications	servicestage:app:create	√	√
Deleting an application based on the application ID	DELETE /v2/{project_id}/cas/applications/{application_id}	servicestage:app:delete	√	√
Modifying application information	PUT /v2/{project_id}/cas/applications/{application_id}	servicestage:app:modify	√	√
Obtaining all applications	GET /v2/{project_id}/cas/applications	servicestage:app:list	√	√
Obtaining application details based on the application ID	GET /v2/{project_id}/cas/applications/{application_id}	servicestage:app:get	√	√
Querying static information about a microservice	GET /v4/{project_id}/registry/microservices/{serviceId}	cse:registry:get	√	√

Permissions	APIs	Actions	IA M Proj ects	Enterp rise Projec ts
Querying static information about all microservices	GET /v4/{project_id}/registry/microservices	cse:registry:get	√	√
Creating static information for a microservice	POST /v4/{project_id}/registry/microservices	cse:registry:modify	√	√
Modifying static information about a microservice	PUT /v4/{project_id}/registry/microservices/{serviceId}/properties	cse:registry:modify	√	√
Querying a microservice schema	GET /v4/{project_id}/registry/microservices/{serviceId}/schemas/{schemaId}	cse:registry:get	√	√
Modifying a microservice schema	PUT /v4/{project_id}/registry/microservices/{serviceId}/schemas/{schemaId}	cse:registry:modify	√	√
Creating a dependency between services	PUT /v4/{project_id}/registry/dependencies	cse:registry:modify	√	√
Querying all providers of a microservice	GET /v4/{project_id}/registry/microservices/{serviceId}/providers	cse:registry:get	√	√
Querying the unique service or schema ID of a microservice	GET /v4/{project_id}/registry/existence	cse:registry:get	√	√
Registering a microservice instance	POST /v4/{project_id}/registry/microservices/{serviceId}/instances	cse:registry:modify	√	√
Querying a microservice instance based on service ID	GET /v4/{project_id}/registry/microservices/{serviceId}/instances	cse:registry:get	√	√

Permissions	APIs	Actions	IA M Proj ects	Enterp rise Projec ts
Deregistering a microservice instance	DELETE /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}	cse:registry:modify	√	√
Querying details about a microservice instance	GET /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}	cse:registry:get	√	√
Modifying the extended information about a microservice instance	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/properties	cse:registry:modify	√	√
Changing the status of a microservice instance	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/status	cse:registry:modify	√	√
Sending heartbeat information	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/heartbeat	cse:registry:modify	√	√
Querying a microservice instance by filter criteria	GET /v4/{project_id}/registry/instances	cse:registry:get	√	√
Querying configurations	GET /v3/{project_id}/configuration/items	cse:config:get	√	√
Deleting static information about a microservice	DELETE /v4/{project_id}/registry/microservices/{serviceId}	cse:registry:modify	√	√
Deleting static information about microservices in batches	DELETE /v4/{project_id}/registry/microservices	cse:registry:modify	√	√
Querying microservice instances in batches	POST /v4/{project_id}/registry/instances/action	cse:registry:modify	√	√

Permissions	APIs	Actions	IA M Proj ects	Enterp rise Projec ts
Querying all schema information about a microservice	GET /v4/{project_id}/registry/microservices/{serviceId}/schemas	cse:registry:get	√	√

12 Appendix

12.1 Status Codes

[Table 12-1](#) describes the status codes.

Table 12-1 Status codes

Status Code	Message	Description
200	-	OK
304	Not Modified	The requested resource is not modified.
400	Bad Request	Invalid request. The client should not repeat the request without modifications.
401	Unauthorized	The authentication information is incorrect or invalid.
403	Forbidden	The server refused the request from the client.
404	Not Found	The requested resource cannot be found. The client should not repeat the request without modifications.
422	Unprocessable Entity	The request was well-formed but was unable to be followed due to semantic errors.
500	Internal Server Error	The server is able to receive the request but it could not understand the request.

12.2 ServiceStage Error Codes

If an error occurs after a request is sent to the system, a response containing an error code will be returned.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to [Error Codes](#).

The following lists the error codes.

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0010 0400	Bad Request	Invalid parameter.	Enter a valid value and try again.
400	SVCSTG.0010 0402	Bad Request	The component has been deployed. Uninstall it first.	Uninstall the component and try again.
400	SVCSTG.0010 0414	Bad Request	The job ID cannot be empty.	Specify a job ID.
400	SVCSTG.0010 0418	Bad Request	The component instance operation is not defined.	Specify a defined operation.
400	SVCSTG.0010 0420	Bad Request	The runtime system is not supported.	Specify a supported runtime system.
400	SVCSTG.0010 0422	Bad Request	The basic resource cannot be empty.	Configure basic resources.
400	SVCSTG.0010 0425	Bad Request	The software package or image of the instance is empty.	The artifact parameter in the packet cannot be empty.
400	SVCSTG.0010 0449	Bad Request	The environment name already exists.	Use another environment name.

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.REPO.0400	Bad Request	Invalid parameter.	Enter a valid value and try again.
400	SVCSTG.REPO.0401	Bad Request	The repository is not implemented.	Select another repository.
400	SVCSTG.REPO.0402	Bad Request	Invalid OAuth code or state.	Perform OAuth 2.0 authorization again to obtain a correct authentication code and random string.
400	SVCSTG.REPO.0403	Bad Request	The resource already exists.	Check whether the resource already exists.
400	SVCSTG.REPO.0450	Bad Request	No repository authorization.	Check whether the authorization name in the request header is correct.
400	SVCSTG.REPO.0451	Bad Request	Invalid repository authorization.	Perform authorization again.
400	SVCSTG.REPO.0452	Bad Request	Invalid clone URL.	Enter a correct clone URL. Example: <code>https://username@github.com/username/demo.git</code> .
404	SVCSTG.00100403	Not Found	Application not found.	Specify an existing application.

Status Code	Error Code	Error Message	Description	Measure
404	SVCSTG.0010 0404	Not Found	Component not found.	Specify an existing component.
404	SVCSTG.0010 0408	Not Found	Request not found.	Specify an existing request.
404	SVCSTG.0010 0411	Not Found	Environment not found.	Specify an existing environment.
404	SVCSTG.0010 0412	Not Found	Instance not found.	Specify an existing instance.
404	SVCSTG.0010 0424	Not Found	ELB instance not found.	Specify an existing ELB instance.
404	SVCSTG.0010 0429	Not Found	Build job not found.	Specify an existing build job.
404	SVCSTG.0010 0433	Not Found	Release task not found.	Specify an existing release task.
404	SVCSTG.0010 0436	Not Found	Instance snapshot not found.	Specify an existing instance snapshot.
404	SVCSTG.0010 0438	Not Found	Template image not found.	Specify an existing template image.
404	SVCSTG.0010 0441	Not Found	The component instance version number has been used in the environment.	Use another version number.
405	SVCSTG.0010 0421	Method Not Allowed	The account quota has been suspended.	Restore the account.

Status Code	Error Code	Error Message	Description	Measure
405	SVCSTG.0010 0427	Method Not Allowed	No permission.	Perform authorization again.
408	SVCSTG.0010 0428	Request Timeout	The system processing timed out.	The system is busy. Try again later.
409	SVCSTG.0010 0401	Conflict	The application is not empty.	Delete all components of the application and then delete the application.
409	SVCSTG.0010 0413	Conflict	The instance name already exists.	Use another instance name.
409	SVCSTG.0010 0419	Conflict	The operation cannot be performed in the current state.	The operation is being processed. Try again later.
409	SVCSTG.0010 0423	Conflict	The component instance cannot be deleted.	The operation is being processed. Try again later.
409	SVCSTG.0010 0426	Conflict	The environment is in use.	Delete the instances deployed in the environment and then delete the environment.
409	SVCSTG.0010 0432	Conflict	The component is already in the release task.	Wait until the current task is complete and try again.
409	SVCSTG.0010 0435	Conflict	The release task cannot be deleted.	The operation is being processed. Try again later.

Status Code	Error Code	Error Message	Description	Measure
409	SVCSTG.0010 0439	Conflict	The ELB instance has been used in the component.	Use another ELB instance.
500	SVCSTG.0010 0500	Internal Server Error	Internal server error.	The system is busy. Try again later.
500	SVCSTG.0010 0501	Internal Server Error	Remote server error.	The system is busy. Try again later.
500	SVCSTG.REPO. 0500	Internal Server Error	Internal server error.	Network error. Try again later.
500	SVCSTG.REPO. 0501	Internal Server Error	Remote repository error.	Check whether the request parameter is correct or the remote Git repository file has been modified.

12.3 CSE Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [Error Codes](#).

Table 12-2 Engine management error codes

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0050 1134	Bad Request	Invalid engine type.	The engine type field is the specType field in the request query. Set it to a valid value (for example, CSE2) and try again. For example, to query the specifications supported by CSE, set specType to CSE2 .
400	SVCSTG.0050 1111	Bad Request	Duplicate engine instance name.	Change the engine name.
400	SVCSTG.0050 1112	Bad Request	Insufficient engine instance quota.	Contact technical support engineers to increase the engine instance quota or delete unnecessary engine instances.
400	SVCSTG.0050 1133	Bad Request	Too many engines at the site.	Contact technical support engineers.

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0050 1103	Bad Request	Empty or invalid billing mode.	Input the billing mode parameter of the engine and check whether the user can create such an engine.
400	SVCSTG.0050 1104	Bad Request	Incorrect engine instance flavor format.	Invalid flavor. Enter a valid flavor.
400	SVCSTG.0050 1105	Bad Request	Incorrect AZ format.	Specify a valid AZ in the request parameter. The AZ field is a string array and cannot be empty. The number of AZ name levels ranges from 1 to 3.
400	SVCSTG.0050 1117	Bad Request	Engine cannot be deleted.	Wait until the engine state changes to Available, Unavailable, or Creation failed, and then try again.
400	SVCSTG.0050 1152	Bad Request	Incorrect task ID format.	Use the job ID returned by the APIs for creating and deleting an engine.
400	SVCSTG.0050 1153	Bad Request	Failed to find the task.	Use the job ID returned by the APIs for creating and deleting an engine.

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0050 1149	Bad Request	Abnormal engine node.	Try again later or contact technical support engineers.
400	SVCSTG.0050 1116	Bad Request	Failed to find the engine instance.	Check whether the input engine ID is correct. Use the username and password of the tenant to log in to CSE and check whether the current tenant has an engine with the corresponding ID.
400	SVCSTG.0050 1201	Bad Request	Incorrect format of the peer engine instance name.	The peer engine instance name must contain 3 to 24 characters, including letters, digits, and hyphens (-), and must start with a letter but cannot end with a hyphen (-). The name cannot be default and must be different from the current engine instance name.

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0050 1202	Bad Request	The peer engine must be an HA engine.	The engine must be an HA engine. Enter the engine flavor again.
400	SVCSTG.0050 1203	Bad Request	The peer engine must be in a single AZ.	The AZs must be single AZs. Enter the AZ list again.
500	SVCSTG.0050 0400	Internal Server Error	Incorrect internal request parameter of the service.	Check the error message. If the parameter is invalid, check whether the input argument is correct. For other errors, contact technical support engineers.

Table 12-3 Dynamic configuration error codes

Status Code	Error Code	Error Message	Description	Measure
400	SVCSTG.0040 0649	Bad Request	Too long key or value.	The configuration item contains 1 to 2048 characters, including digits, letters, and special characters (. - _ : []). The value can contain 1 to 131072 characters, including digits, letters, and special characters (' ~ ! @ # \$ % ^ & * () _ \ { } : " < > ? ` - [] , / =) .
400	SVCSTG.0040 0650	Bad Request	Invalid configuration file.	Upload a valid file.
400	SVCSTG.0040 0651	Bad Request	The file is empty or in an incorrect format.	The uploaded file is empty or not in JSON format.
400	SVCSTG.0040 0652	Bad Request	The key or value is empty or is not a string.	The uploaded key or value is null or is not a string.
400	SVCSTG.0040 0653	Bad Request	Invalid label or body.	Enter a valid body or label in the query parameter.
400	SVCSTG.0040 0100	Bad Request	Invalid parameter.	Change parameters as prompted.
404	SVCSTG.0040 0106	Not Found	The resource does not exist.	Enter valid search criteria.

Status Code	Error Code	Error Message	Description	Measure
409	SVCSTG.00400107	Conflict	The resource already exists.	Do not create the same record.
500	SVCSTG.00300605	Failed to connect the configuration center to ETCD.	Internal Server Error	Contact technical support engineers.
400	SVCSTG.00300401	Invalid token	Unauthorized	Enter a correct token.

12.4 ServiceComb Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [Error Codes](#).

Category	Status Code	Error Code	Error Message	Description	Measure
Common error codes	400	400001	Bad Request	Invalid parameter.	Change parameters as prompted.
	404	404001	Not Found	The resource does not exist.	Enter valid search criteria.
	409	409001	Conflict	The resource already exists.	Do not create the same record.
	500	500003	Internal Server Error	Internal server error.	Contact technical support engineers.
Microservices	400	400002	Bad Request	The service is unhealthy.	Try again later or contact technical support engineers.

Cat ego ry	Status Code	Error Code	Error Message	Description	Measure
		400010	Bad Request	The service already exists.	Modify the service ID or microservice description in the body of the request for creating a microservice.
		400011	Bad Request	No available backend instance.	Try again later or contact technical support engineers.
		400012	Bad Request	The service does not exist.	Enter a valid service ID.
		400013	Bad Request	The microservice cannot be deleted because instances have been deployed.	Take the instance offline and then delete the microservice. Alternatively, forcibly delete the microservice by setting the query parameter force to true in the URL.
		400014	Bad Request	The schema ID does not exist.	Enter a valid schema ID.
		400015	Bad Request	The schema cannot be modified.	The schema has been registered and cannot be modified.
		400016	Bad Request	The schema does not exist.	Register the schema first.
		400017	Bad Request	The instance does not exist.	Enter a valid instance ID.

Category	Status Code	Error Code	Error Message	Description	Measure
		400018	Bad Request	The tag does not exist.	This error code is usually generated for a query API, indicating that the tag does not exist. The service performs subsequent processing based on the returned value.
		400019	Bad Request	The rule already exists.	The rule is repeatedly created. Generally, this error can be ignored.
		400020	Bad Request	Blacklist and whitelist error.	Modify the parameter value based on the error message.
		400021	Bad Request	The rule cannot be modified.	You can modify the microservice information only after changing the version number.
		400022	Bad Request	The rule does not exist.	This error code is usually generated for a query API, indicating that the rule does not exist. The service performs subsequent processing based on the returned value.

Category	Status Code	Error Code	Error Message	Description	Measure
		400023	Bad Request	The microservice cannot be deleted because it is the dependent service of other microservices.	You can forcibly delete microservices by setting the query parameter force to true in the URL.
		400024	Bad Request	Insufficient permissions.	Use a proper account to perform operations.
		400025	Bad Request	The port already exists.	Check whether the port is occupied by another instance.
		400026	Bad Request	The microservice instance does not exist.	Enter a correct version number or range.
		400100	Bad Request	Insufficient quota.	The quotas of resources such as microservices, instances, or schemas are insufficient. Delete some resources and create again.

Category	Status Code	Error Code	Error Message	Description	Measure
	401	401204	Unauthorized	Unauthorized.	This parameter is mandatory if security authentication is enabled for the exclusive microservice engine. Otherwise, this parameter is not required. The token of the exclusive microservice engine with security authentication enabled is in the following format: Authorization:Bearer {Token} For details about how to obtain the token, see Obtaining the User Token of an Exclusive Microservice Engine .
		401201	Unauthorized	Unauthorized.	Enter a valid authorization.
	403	403001	Forbidden	Insufficient permissions.	Use a proper account to perform operations.
	500	500011	Internal Server Error	The registry service is unavailable.	Contact technical support engineers.
500101		Internal Server Error	No quota.	Try again later or contact technical support engineers.	

Category	Status Code	Error Code	Error Message	Description	Measure
		500605	N/A	Failed to connect to etcd of the configuration center.	Try again later or contact technical support engineers.
Authentication	401	401202	Unauthorized	Incorrect account name or password.	Enter the correct account name and password.

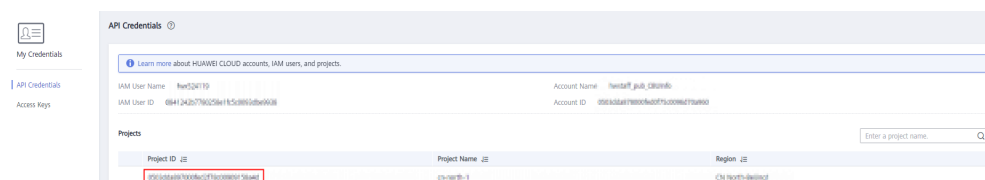
12.5 Obtaining a Project ID

Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain an ID, perform the following operations:

- Step 1** Log in to ServiceStage.
- Step 2** Move the cursor to the username in the upper right corner and select **My Credentials** from the drop-down list.
- Step 3** On the **Projects** area, obtain the required project ID in the project list.

Figure 12-1 Viewing project IDs



----End

Obtaining a Project ID by Calling an API

A project ID can also be obtained by calling a specific API. For details, see [Querying Project Information Based on the Specified Criteria](#).

The API used to obtain a project ID is **GET https://{Endpoint}/v3/projects/**, where *{Endpoint}* indicates the IAM endpoint. You can obtain the IAM endpoint from [Regions and Endpoints](#). For details on API calling authentication, see [Authentication](#).

The following is an example response. The value of **id** in the **projects** section is the project ID.

```
{  
  "projects": [  

```

```
{
  "domain_id": "65382450e8f64ac0870cd180d14e684b",
  "is_domain": false,
  "parent_id": "65382450e8f64ac0870cd180d14e684b",
  "name": "region_id",
  "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"
}
}
```

12.6 Obtaining an Account ID

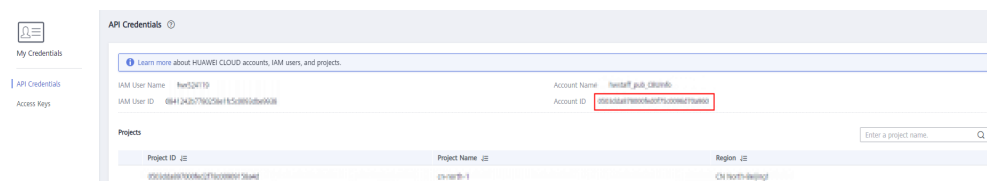
An accountA tenant ID is required for some URLs when an API is called. To obtain an ID, perform the following operations:

Step 1 Log in to ServiceStage.

Step 2 Move the cursor to the username in the upper right corner and select **My Credentials** from the drop-down list.

View the account ID.

Figure 12-2 Viewing the account ID



----End

13 Change History

Released On	Description
2023-07-10	This issue is the fifth official release, which incorporates the following change: Added Application Management V3 APIs .
2022-02-15	This issue is the fourth official release, which incorporates the following change: Corrected some errors.
2021-10-30	This issue is the third official release, which incorporates the following change: Added Obtaining the User Token of an Exclusive Microservice Engine .
2021-03-30	This issue is the second official release, which incorporates the following changes: <ul style="list-style-type: none">• Optimized the description in Application Management V2 APIs.• Corrected some errors.
2020-02-29	This issue is the first official release.