Alibaba Cloud
Apsara Stack Enterprise

User Guide - Middleware and Enterprise Applications

Version: 1907, Internal: V3.8.0

Issue: 20191227

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent**

ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips , and other content.	 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	**This format is used for a required value, where only one item can be selected.**	`switch {active|stand}`

Contents

1 API Gateway

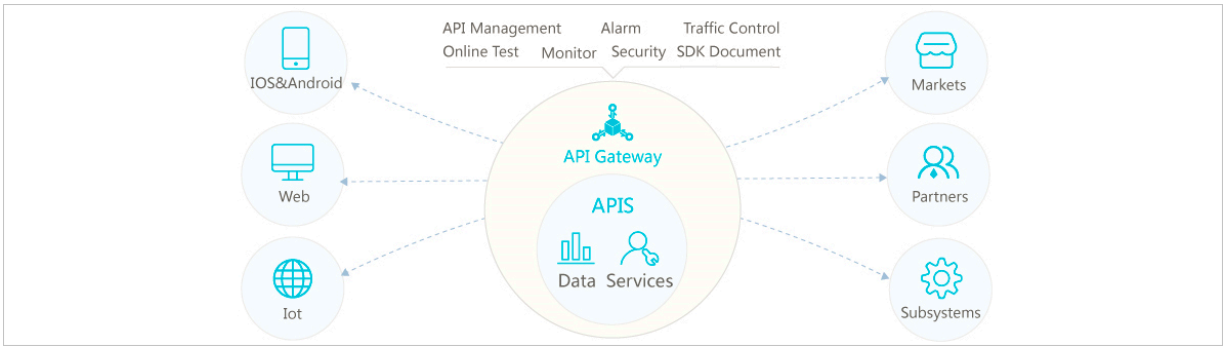
1.1 What is API Gateway?

API Gateway provides a complete suite of API hosting services that helps you share capabilities, services, and data with partners in the form of APIs. It also enables you to release your APIs in the API marketplace for more developers to purchase and use.

- **API Gateway provides multiple security mechanisms to secure APIs and reduce risks that open APIs introduce. These mechanisms include protection against attacks such as replay attacks, request encryption, identity authentication, access control, and throttling.**
- **API Gateway provides API lifecycle management that allows you to create, test , publish, and unpublish APIs. It also generates SDK and API documentation to improve API management and iterative efficiency.**
- **API Gateway provides simple O&M tools such as monitoring, alerting, and analysis as well as API marketplace to reduce O&M costs of APIs.**

API Gateway maximizes the reuse of capabilities. It allows enterprises to share their capabilities with each other and focus on their core business, achieving mutually beneficial results.

Figure 1-1: API Gateway



1.2 Log on to the API Gateway console

This topic demonstrates how to log on to the API Gateway console from Google Chrome.

Prerequisites

- **Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is http://***IP address or domain name of the Apsara Stack console***/manage.**
- **We recommend that you use the Chrome browser.**

Procedure

1. **Open your browser.**
2. **In the address bar, enter the access address of the Apsara Stack console in the format of http://***IP address or domain name of the Apsara Stack console***/manage, and then press Enter.**
3. **Enter the correct username and password.**
   - **The system has a default super administrator with the username super and password super. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.**
   - **You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%)).**
4. **Click LOGIN to go to the Dashboard page.**
5. **In the top navigation bar, click the icon and choose Compute, Storage & Networking > API Gateway.**

1.3 Quick start for consumers

1.3.1 Overview

You can use API Gateway to call the API services enabled by other Alibaba Cloud users or third-party service providers. API Gateway provides a series of management and support services.

Call an API based on the following conditions:

- **API: The API that you call is clearly defined by API parameters.**
- **Application: The application that you use to call the API has a key pair that uniquely identifies you.**
- **Authorization relationship between the API and application: An application can be used to call an API only when the application has been granted the permission to call that API. This permission is granted through authorization.**

1.3.2 Step 1: View API settings

You must create an application and provide the application ID to the API provider before the application can be authorized in the API Gateway console. For more information about applications, see *Create an application*. **Assume that you have created an application and the API provider has authorized your application.**

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Applications tab to go to the Applications tab.**

   Your created applications are displayed on the Applications tab.
3. **Click the application ID to go to the application details page.**

   Basic Information, AppKey, and Callable APIs are displayed.

   On the details page,

   - **The AppKey section shows the AppKey and the AppSecret of an application. Your API request must contain the AppKey and the AppSecret. API Gateway verifies your identity based on this key pair.**
   - **The Callable APIs section shows the APIs that applications have been authorized to call. If the API provider has authorized the applications, the corresponding APIs are displayed. Click the management icon in the Actions**

column corresponding to the API and choose View Details from the shortcut menu to view details of the API.

1.3.3 Step 2: Create an application

Applications are the identities that you use to call APIs. You can own multiple applications. Your applications can be authorized to call different APIs based on your business requirements. Applications instead of user accounts are authorized to call APIs. In the API Gateway console, you can create, modify, or delete applications, view details of applications, manage key pairs, and view callable APIs of applications.

Each application has an AppKey and an AppSecret. You can regard them as an account and a password. When you call an API, you must pass in the AppKey as a parameter. AppSecret is used to calculate the signature string. API Gateway authenticates the key pair to verify your identity. An application must be authorized to call an API. Both authorization and authentication are intended for applications.

You can log on to the API Gateway console to create applications on the Applications tab.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Applications tab to go to the Applications tab.
3. Click Create Application.
4. Specify parameters and click Create.

   The application name must be globally unique. It must be 4 to 26 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.

   After an application is created, the system automatically assigns an AppKey and an AppSecret to it. You must use the AppSecret to calculate the signature string. When calling an API, you must include the signature string in the request. API Gateway verifies your identity based on the signature string.

   On the Applications tab, click the application ID to go to the application details page. The AppKey and AppSecret are displayed on the application details page. You can reset the AppSecret as needed.

1.3.4 Step 3: Obtain authorization

Authorization is the process of authorizing an application to call an API. Your applications must be authorized before they can call APIs.

You must provide your application IDs to the API provider for authorization. After authorization, you can view the APIs that your applications have been authorized to call in the API Gateway console.

The APIs that your applications have been authorized to call are displayed in the Callable APIs section on the application details page.

After the API provider authorizes your applications to call APIs, you do not need to and cannot authorize your applications.

1.3.5 Step 4: Call an API

You can use call examples of multiple programming languages in the API Gateway console to test APIs. You also can configure an HTTP or HTTPS request to test APIs.

Part 1: Request

Request URL

```
http://e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com/demo/
post
```

Request method

```
POST
```

Request body

```
FormParam1=FormParamValue1&FormParam2=FormParamValue2
//HTTP Request Body
```

Request header

```
Host: e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com
Date: Mon, 22 Aug 2016 11:21:04 GMT
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
// The request body type. Set the request body type based on the
contents of the actual request.
Accept: application/json
// The response body type. Some APIs can return data in the correspond
ing format based on the specified response type. We recommend that you
 manually set this request header. If the request header is not set,
some HTTP clients will use the default value */*, causing a signature
error.
X-Ca-Request-Mode: debug
```

```
// Specifies whether to enable the Debug mode. This parameter is not
case-sensitive. If it is not specified, the Debug mode is disabled by
default. This mode is typically enabled in the API debugging phase.
X-Ca-Version: 1
// The API version number. Set the value to 1. Default value: 1.
X-Ca-Signature-Headers: X-Ca-Request-Mode,X-Ca-Version,X-Ca-Stage,X-Ca
-Key,X-Ca-Timestamp
// Custom request headers involved in signature calculation. The
server will read the request headers based on this configuration to
sign the request. This configuration does not include the Content-Type
, accept, Content-MD5, and Date request headers, which are already
included in the basic signature structure. For more information about
the signature, see Sign signatures.
X-Ca-Stage: RELEASE
// The stage of the API. Valid values: TEST, PRE, and RELEASE. This
parameter is case-insensitive. The API provider can select the stage
to which the API is published. The API can be called only after it is
 published to the specified stage. Otherwise, the system will prompt
that the API cannot be found or that the request URL is invalid.
X-Ca-Key: 60022326
// The AppKey. You must obtain the AppKey in the API Gateway console.
Applications can only call APIs after they have been authorized.
X-Ca-Timestamp: 1471864864235
// The request timestamp. This value is a UNIX timestamp representing
the number of milliseconds that have elapsed since January 1, 1970 00:
00:00 UTC. The timestamp is valid for 15 minutes by default.
X-Ca-Nonce:b931bc77-645a-4299-b24b-f3669be577ac
// The unique ID of the request. AppKey, API, and Nonce must be unique
 within the last 15 minutes. To prevent replay attacks, you must
specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.
X-Ca-Signature: FJleSrCYPGCU7dMlLTG+UD3Bc5Elh3TV3CWHtSKh1Ys=
// The request signature.
CustomHeader: CustomHeaderValue
// Custom request headers. CustomHeaderValue is used as an example.
You can set multiple custom request headers in actual requests based
on the definition of the API being called.
```

Part 2: Response

Status code

```
400
// The status code of the response. If the value is greater than or
 equal to 200 and less than 300, the request was successful. If the
value is greater than or equal to 400 and less than 500, a client-side
 error occurred and the request failed. If the value is greater than
500, a server-side error occurred and the call failed.
```

Response header

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
// The unique ID of the request. When API Gateway receives a request,
it generates a request ID and returns the request ID to the client in
the X-Ca-Request-Id header. We recommend that you record the request
ID in both the client and the back-end service for troubleshooting and
 tracking.
X-Ca-Error-Message: Invalid Url
// The error message returned by API Gateway. When a request fails,
API Gateway returns the error message to the client in the X-Ca-Error-
Message header.
X-Ca-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
```

```
// The message can be returned only when the debug mode is enabled.
The message is used only for reference at the debugging stage.
```

When you call an API by using HTTP or HTTPS, the request must include the signature information. For more information about how to calculate and pass the encrypted signature, see *Request Signatures*.

1.4 Quick start for providers

1.4.1 Overview

This topic is a quick start guide for you to create and publish an API.

1. **Create an API group.**
2. **Create an API.**
3. **Publish an API.**
4. **Authorize applications to call the API.**

1.4.2 Create a group

You can create an API Group in the API Gateway console.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Groups tab.**
3. **On the Groups tab, click Create Group.**
4. **In the Create Group dialog box that appears, set parameters and click Create.**

   The name of a group must be globally unique. The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.

1.4.3 Create an API

Procedure

1. *Log on to the API Gateway console*.
2. **Click the APIs tab.**
3. **Click Create API.**

4. Specify the basic information of the API and click Next.

Parameter	Description
Groups	The basic management unit of an API. You must create a group before creating an API. When you select a group, you select a region for the API.
API Name	The name of the API to be created.
Authentication Mode	The authentication mode of API requests. Valid values: `Alibaba Cloud Applications` and `None`. • `Alibaba Cloud Applications`: This mode requires the requester to pass the application authentication to call the API. • `None`: This mode allows any user who knows the request definition of the API to initiate a request. API Gateway directly forwards the requests to your back-end service without verifying the identity of the requesters.
Signature Algorithm	The algorithm used to sign API requests. Valid values: • HmacSHA256 • HmacSHA1 and HmacSHA256
Description	The description of the API.

5. **Define API requests. Define how users call your APIs, including the request type, network protocol, URL suffix, HTTP method, and request mode.**

Parameter	Description
Request Type	**The request type. Four options are available. However, only the common request type is supported.** • `Common Request`**: indicates common HTTP or HTTPS requests.** • `Logon Request (Bidirectional Communication)`**: indicates the bidirectional control signal to register devices. It is sent from the client to the server.** • `Logoff Request (Bidirectional Communication)`**: indicates the bidirectional control signal to register devices. It is sent from the client to the server. After devices are deregistered, server-to-client notifications are no longer received.** • `Server-to-Client Notification (Bidirectional Communication)`**: After receiving the registration signal sent from the client, the back-end service records the device ID and send a server-to-client notification to API Gateway. Then, API Gateway sends the notification to the device. As long as the device is online, API Gateway sends the server-to-client notification to the device.**
Network protocol	`HTTP`**,** `HTTPS`**, and** `WEBSOCKET` **are supported in API calls.**
URL Suffix	**The API request path. It corresponds to the service host . The request path can be different from the actual back-end service path. You can specify any valid and semantically-correct path as the request path. You can configure dynamic parameters in the request path, which requires users to specify path parameters in the request. At the same time, the path parameters can be mapped to query and header parameters that are received by the back-end service.**
HTTP Method	**You can select** `PUT`**,** `GET`**,** `POST`**,** `PATCH`**,** `DELETE`**, and** `HEAD`**.**

Parameter	Description
Request Mode	You can select either `Request Parameter Mapping` or `Request Parameter Passthrough`. · Request Parameter Mapping indicates that you must configure request and response data mappings for query , path, and body form parameters. API Gateway only passes the configured parameters through to the back end. Other parameters are filtered out. · Request Parameter Passthrough indicates that you do not need to configure query and body form parameters, but still must configure path parameters in the Request Parameters section. All parameters sent from the client are passed through by API Gateway to the back-end service.

6. Define request parameters.

   Define the request parameters of your APIs. You can specify different request parameters for different parameter paths. Head, Query, Body, and Path can be selected. When you configure a dynamic path parameter, you must provide a description of this dynamic parameter. Supported parameter types include String, Number, and Boolean.

   · Note that the names of all parameters must be unique.
   · You can use the shortcut key on the left to adjust the parameter order.
   · To delete unwanted parameters, you can click the management icon in the Actions column and choose Delete from the shortcut menu.

7. Configure parameter validation rules.

   To configure validation rules, you can click the management icon in the Actions column and choose Configure Advanced Settings from the shortcut menu.
   For example, you can configure validation rules such as the length of a string and enumeration of numbers. API Gateway pre-verifies requests based on the validation rules. Requests with invalid parameters are not sent to your back-end service. This greatly reduces the workload on your back-end service.

8. Configure your back-end service and click Next.

   This section defines mappings between request and response parameters, and specifies the API configurations of your back-end service. Back-end service configurations include the back-end service URL, URLsuffix, timeout period,

parameter mappings, constants, and system parameters. After receiving requests, API Gateway converts the format of the requests to the format required by the back-end service based on the back-end service configuration. Then, API Gateway forwards the requests to your back-end service.

 **Note:**
You can enter the following parameters: dynamic parameters in the URL suffix , header parameters, query parameters, body parameters (non-binary), constants, and system parameters. The names of these parameters must be globally unique. For example, you cannot specify a parameter of the same name in both the header and query path.

a. Configure the basic settings of the back-end service.

Parameter	Description
Backend Service Type	By default, HTTP or HTTPS service is supported. API Gateway only supports HTTP or HTTPS service. The Mock option indicates that you do not access the back-end service, but expect API Gateway to simulate responses based on the specified values. For more information, see the Mock option.
VPC ID	Do Not Authorize Access to VPCs is selected by default. The back-end service can be connected with API Gateway directly. If the back-end service is deployed in a VPC, and API Gateway needs to access the back-end service, select the corresponding VPC ID for the back-end service.
Backend Service URL	The host name of the back-end service can be a domain name or `http(s)://host:port`. If the back-end service is deployed in a VPC, the back-end service URL can be in the http(s)://{ip}.{vpcId}.gateway.vpc:{port} format. Example: http://172.10.1.3.vpc-12ssar3e123.gateway.vpc:8080.
URL Suffix	The actual request path of your API service on your back-end server. If you want to receive dynamic parameters in the back-end path, you must declare parameter mappings by specifying the locations and names of the corresponding request parameters.
HTTP Method	`PUT`, `GET`, `POST`, `PATCH`, `DELETE`, and `HEAD` can be selected.

Parameter	Description
Timeout	The response time for API Gateway to access the back-end service after API Gateway receives an API request. The response time is from the time when API Gateway sends an API request to the back-end service to the time when API Gateway receives responses returned by the back-end service. The response time cannot exceed 30 seconds. If API Gateway does not receive a response from the back-end service within 30 seconds, API Gateway stops accessing the back-end service and returns an error message.

b. Configure back-end service parameters.

API Gateway can set up mappings between request and response parameters, including name mappings and parameter path mappings. API Gateway can map a request parameter at any location such as path, header, query, or body to a response parameter at a different location. In this way, you can package your back-end services into standard APIs. This section describes the mappings between front-end APIs and back-end APIs.

Note:
The request and response parameters must be globally unique.

c. Configure constant parameters.

If you want API Gateway to attach the apigateway tag to each request that API Gateway forwards to your back-end service, you can configure this tag as a constant. Constants are not visible to your users. After API Gateway receives requests, API Gateway automatically adds constants to the specified locations and then forwards the requests to your back-end service.

d. Configure system parameters.

By default, API Gateway does not send its system parameters to you. However, if you need the system parameters, you can configure their locations and names. The following table lists the system parameters.

Parameter	Description
CaClientIp	The IP address of the client sending a request.
CaDomain	The domain name from which a request originates.

Parameter	Description
CaRequestHandleTime	The request time. It must be in GMT.
CaAppId	The ID of the application sending a request.
CaRequestId	The unique ID of the request.
CaApiName	The name of the API.
CaHttpSchema	The protocol that is used to call the API. The protocol can be either HTTP or HTTPS.
CaProxy	The proxy (AliCloudApiGateway).

9. Define responses and click Create.

   You can set Response ContentType, Success Result Example, Error Response Example, and Error Codes. API Gateway does not parse responses, but forwards them to API users.

1.4.4 Publish an API

After you create an API, you must publish the API to the test, pre-release, or release environment before it can be called.

- When you use a second-level or independent domain name to access an API that has been published to an environment, you must specify the environment in the request header.
- If you attempt to publish an API that already has a running version in the test or release environment, the previously running version will be overwritten by the new API. All historical versions and definitions are recorded, allowing you to roll back the API to previous versions as needed.
- You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API.

Step 1: Test the API

To simulate API requests, you can create an application and authorize the application to call your API.

You can compile code based on actual scenarios, or use the SDK samples provided by API Gateway to call the API.

You can publish the API to the test or release environment. If no independent domain name is bound to the group to which the API belongs, you can test or call the API by using the second-level domain name. When you make an API request, specify the environment of the API by setting the X-Ca-Stage header to TEST, PRE , or RELEASE. If you do not specify the header, the API will be published to the release environment.

Step 2: Publish the API

After testing the API, you can publish it.

API Gateway enables you to manage versions of APIs in the test or release environment. You can publish or unpublish an API, and switch the versions of an API. The switch of versions takes effect in real time.

1. *Log on to the API Gateway console*.
2. **Click the API tab.**
3. **Select the API that you want to publish, click the management icon in the Actions column corresponding to the API, and choose Publish from the shortcut menu.**
4. **In the dialog box that appears, select the environment where the API is published, enter a description, and click OK.**

1.4.5 Authorize an application

You must authorize an application before it can call an API. After publishing an API to the release environment, you must authorize the user applications to call the API. You can authorize an application to call an API or revoke the authorization of an application to call an API. API Gateway verifies the authorization relationship.

Note:

- **You can authorize one or more applications to use one or more APIs.**
- **If an API has been published to both the test and release environments and the test environment has been selected, applications are only authorized to call the API in the test environment.**
- **You can locate an application based on its ID provided by a user.**
- **If you want to revoke the authorization of an application to call an API, go to the Authorization tab of the API. Then, select the application, click the management icon in the Actions column corresponding to the application, and choose**

Deauthorize from the shortcut menu, or click Deauthorize in the upper-right corner.

An application indicates the identity of a requester. Before you or your users test or call APIs, you or your users must create an application that is used as the identity of a requester. Then, you need to authorize the application to call an API.

Note:

Authorizations are environment-specific. If you want to use an application to call an API in both the test and release environments, you must authorize the application in both environments separately. Otherwise, errors may occur due to the inconsistency between the authorized environment and requested environment.

Procedure

1. *Log on to the API Gateway console*.
2. Click the APIs tab.
3. Select the API that an application is authorized to call, click the management icon in the Actions column corresponding to the API, and choose Authorize from the shortcut menu. The Authorize Applications dialog box appears.
4. Select the environment and the application to authorize.

   The system automatically displays the applications that belong to your account. If you want to authorize an application that belongs to a different account, search for the application by application ID.
5. Select the application to authorize, and click the > icon to add the selected application to the right-side list.
6. Click OK to complete the authorization process.

1.5 Call an API

1.5.1 Manage applications

1.5.1.1 Create an application

Applications are the identities that you use to call APIs. You can own multiple applications. Your applications can be authorized to call different APIs based on your business requirements. Applications instead of user accounts are authorized

to call APIs. In the API Gateway console, you can create, modify, or delete applications, view details of applications, manage key pairs, and view callable APIs of applications.

Each application has an AppKey and an AppSecret. You can regard them as an account and a password. When you call an API, you must pass in the AppKey as a parameter. AppSecret is used to calculate the signature string. API Gateway authenticates the key pair to verify your identity. An application must be authorized to call an API. Both authorization and authentication are intended for applications.

You can log on to the API Gateway console to create applications on the Applications tab.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Applications tab to go to the Applications tab.
3. Click Create Application.
4. Specify parameters and click Create.

   The application name must be globally unique. It must be 4 to 26 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.

   After you create the application, the system automatically assigns an AppKey and an AppSecret to it. You must use the AppSecret to calculate a signature string. When calling an API, you must include the signature string in the request. API Gateway verifies your identity based on the signature string.

   On the Applications tab page, click the application ID to go to the application details page. The AppKey and AppSecret are displayed on the application details page. You can reset the AppSecret as needed.

1.5.1.2 View application details

You can view the details of existing applications.

Procedure

1. *Log on to the API Gateway console*.
2. Click Applications tab to go to the Applications tab.

3. **Click the application that you want to view.**

   You can view the basic information, AppKey, and callable APIs of the application.

1.5.1.3 Modify an application

You can modify existing applications.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Applications tab to go to the Applications tab.**
3. **Select the application you want to modify, click the management icon in the Actions column corresponding to the application, and choose Change from the shortcut menu.**
4. **Modify the application information and click OK.**

1.5.1.4 Delete an application

You can delete existing applications.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Applications tab to go to the Applications tab.**
3. **Select the application you want to delete, click the management icon in the Actions column corresponding to the application, and choose Delete from the shortcut menu.**
4. **In the dialog box that appears, click OK.**

1.5.2 View existing APIs

You can view existing APIs in the API Gateway console.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the APIs tab.**

1.5.3 Authorize an application

Authorization is the process of authorizing an application to call an API. Your applications must be authorized before they can call APIs.

You must provide your application IDs to the API provider for authorization. After authorization, you can view the APIs that your applications have been authorized to call in the API Gateway console.

The APIs that your applications have been authorized to call are displayed in the Callable APIs section on the application details page.

After the API provider authorizes your applications to call APIs, you do not need to and cannot authorize your applications.

1.5.4 Encrypt a signature

When you call an API, you must construct a signature string and add the calculated signature string to the request header. API Gateway uses symmetric encryption to verify the identity of the request sender.

- **Add the calculated signature string to the request header.**
- **Organize the request parameters into a string-to-sign based on** *Request signatures*. **Then, use the algorithm provided in the SDK sample to calculate the signature. The result is the calculated signature string.**
- **Both HTTP and HTTPS requests must be signed.**

For more information about how to construct a string-to-sign, see *Request signatures*. **Replace the AppKey and AppSecret in the SDK sample with your own AppKey and AppSecret. Then, construct a string-to-sign based on Request signatures. After creating the string-to-sign, you can use it to initiate a request.**

1.5.5 Request signatures

Endpoint

- **Each API belongs to an API group, and each API group has a unique endpoint. An endpoint is an independent domain name that is bound to an API group by the API provider. API Gateway uses endpoints to locate API groups.**
- **An endpoint must be in the** `www.[ Independent domain name].com/[Path]?[ HTTP method]` **format.**

- **API Gateway locates a unique API group by endpoint, and locates a unique API in the group through the combination of Path and HTTP method.**
- **After you purchase an API, you can obtain the API documentation from the Purchased APIs list in the API Gateway console. If you have not purchased an API, you need to require the API provider to authorize your applications to call the API. After authorization, you can obtain the API documentation from the Callable APIs list on the application details page.**

System-level header parameters

- **(Required) X-Ca-Key: AppKey.**
- **(Required) X-Ca-Signature: the signature string.**
- **(Optional) X-Ca-Timestamp: the timestamp passed in by the API caller. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970 00:00:00 UTC. The timestamp is valid for 15 minutes by default.**
- **(Optional) X-Ca-Nonce: the UUID generated by the API caller. To prevent replay attacks, you must specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.**
- **(Optional) Content-MD5: When the request body is not a form, you can calculate the MD5 value of the request body. Then, you can send the value to API Gateway for MD5 validation.**
- **(Optional) X-Ca-Stage: the stage of the API. Valid values: TEST, PRE, and RELEASE. Default value: RELEASE. If the API that you intend to call has not been published to the production environment, you must specify the value of this parameter. Otherwise, a URL error will be reported.**

Signature validation

Construct the signature calculation strings

```
String stringToSign=
HTTPMethod + "\n" +
Accept + "\n" +                // We recommend that you specify the
Accept header in the request. If the request header is not set, some
HTTP clients will use the default value */*. This will cause signature
 validation to fail.
Content-MD5 + "\n"
Content-Type + "\n" +
Date + "\n" +
Headers +
```

```
Url
```

An HTTP method must be uppercase, for example, POST.

If Accept, Content-MD5, Content-Type, and Date are empty, add a line break `\n` after each of them . If Headers is empty, `\n` is not required.

Content-MD5

Content-MD5 indicates the MD5 value of the request body. The MD5 value is calculated only when the request body is not a form. The calculation formula is as follows:

```
String content-MD5 = Base64.encodeBase64(MD5(bodyStream.getbytes("UTF-
8")));
```

bodyStream indicates a byte array.

Headers

Headers indicates the string constructed by the keys and values of the header parameters that are used for Headers signature calculation. We recommend that you use the parameters starting with X-Ca and custom header parameters for signature calculation.

Notice:
The following parameters are not used for Headers signature calculation: X-Ca-Signature, X-Ca-Signature-Headers, Accept, Content-MD5, Content-Type, and Date.

Headers construction method:

Sort the header keys used for Headers signature calculation in alphabetical order. Construct the string based on the following rule: If the value of a header parameter is empty, use `HeaderKey + ":" + "\n"` for signature calculation. The key and colon (:) must be retained.

```
String headers =
HeaderKey1 + ":" + HeaderValue1 + "\n"\+
HeaderKey2 + ":" + HeaderValue2 + "\n"\+
...
HeaderKeyN + ":" + HeaderValueN + "\n"
```

The keys of the header parameters used for Headers signature calculation must be separated with commas (,), and placed in the request headers. The key is X-Ca-Signature-Headers.

Url

Url indicates the `Form` parameter in `Path + Query + Body`. For `Query + Form`, sort `Key` in lexicographical order and construct the string based on the following rules: If `Query` or `Form` is empty, `Url = Path` and a question mark (`?`) is not required. If the `value` of a parameter is empty, only `key` is used for signature calculation and an equal sign (=) is not required.

```
String url =
Path +
"?" +
Key1 + "=" + Value1 +
"&" + Key2 + "=" + Value2 +
...
"&" + KeyN + "=" + ValueN
```

Notice:

The `Query` parameter or the `Form` parameter may have multiple `values`. If both parameters have multiple values, only the first value of each parameter is used for signature calculation.

Signature calculation

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");
byte[] keyBytes = secret.getBytes("UTF-8");
hmacSha256.init(new SecretKeySpec(keyBytes, 0, keyBytes.length, "
HmacSHA256"));
String sign = new String(Base64.encodeBase64(Sha256.doFinal(stringToSi
gn.getBytes("UTF-8")),"UTF-8"));
```

`secret` indicates an AppSecret.

Signature passing

Add the calculated signature to the request header. The key is X-Ca-Signature.

Signature troubleshooting

If signature validation fails, API Gateway places the returned string-to-sign in the HTTP response header and sends the response to the client. The key is X-Ca-Error-Message. You only need to compare the string-to-sign calculated by the client with the string-to-sign returned by the server.

If both are the same, check the AppSecret used for signature calculation.

The HTTP header does not support line breaks. Line breaks in the string-to-sign are filtered out. Ignore the line breaks when you make a comparison.

Signature demo

For more information about the Java demo of signature calculation, see *https://github.com/aliyun/api-gateway-demo-sign-java*.

1.5.6 Call examples

You can use call examples of multiple programming languages in the API Gateway console to test APIs. You also can configure an HTTP or HTTPS request to test APIs.

Part 1: Request

Request URL

```
http://e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com/demo/
post
```

Request method

```
POST
```

Request body

```
FormParam1=FormParamValue1&FormParam2=FormParamValue2
//HTTP request body
```

Request header

```
Host: e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com
Date: Mon, 22 Aug 2016 11:21:04 GMT
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
// The request body type. Set the request body type based on the
actual request you want to make.
Accept: application/json
// The response body type. Some APIs can return data in the correspond
ing format based on the specified response type. We recommend that you
 manually set this request header. If the request header is not set,
some HTTP clients will use the default value */*, causing a signature
error.
X-Ca-Request-Mode: debug
// Specifies whether to enable the Debug mode. This parameter is not
case-sensitive. If it is not specified, the Debug mode is disabled by
default. This mode is typically enabled in the API debugging phase.
X-Ca-Version: 1
// The API version number. Set the value to 1. Default value: 1.
X-Ca-Signature-Headers: X-Ca-Request-Mode,X-Ca-Version,X-Ca-Stage,X-Ca
-Key,X-Ca-Timestamp
// The custom request headers involved in signature calculation. The
 server will read the request headers based on this configuration to
sign the request. This configuration does not include the Content-Type
, Accept, Content-MD5, and Date request headers, which are already
included in the basic signature structure. For more information about
the signature, see Request signatures.
X-Ca-Stage: RELEASE
```

```
// The stage of the API. Valid values: TEST, PRE, and RELEASE. This
parameter is not case-sensitive. The API provider can select the stage
 to which the API is published. The API can be called only after it is
 published to the specified stage. Otherwise, the system will prompt
that the API cannot be found or that the request URL is invalid.
X-Ca-Key: 60022326
// The AppKey of the request. You must obtain the AppKey in the API
Gateway console. Applications can only call APIs after they have been
authorized.
X-Ca-Timestamp: 1471864864235
// The request timestamp. This value is a UNIX timestamp representing
the number of milliseconds that have elapsed since January 1, 1970 00:
00:00 UTC. The timestamp is valid for 15 minutes by default.
X-Ca-Nonce:b931bc77-645a-4299-b24b-f3669be577ac
// The unique ID of the request. AppKey, API, and Nonce must be unique
 within the last 15 minutes. To prevent replay attacks, you must
specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.
X-Ca-Signature: FJleSrCYPGCU7dMlLTG+UD3Bc5Elh3TV3CWHtSKh1Ys=
// The request signature.
CustomHeader: CustomHeaderValue
// The custom request headers. CustomHeaderValue is used as an example
. You can set multiple custom request headers in actual requests based
 on the definition of the API being called.
```

Part 2: Response

Status code

```
400
// The status code of the response. If the value is greater than or
 equal to 200 and less than 300, the request was successful. If the
value is greater than or equal to 400 and less than 500, a client-side
 error occurred and the request failed. If the value is greater than
500, a server-side error occurred and the call failed.
```

Response header

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
// The unique ID of the request. When API Gateway receives a request,
it generates a request ID and returns the request ID to the client in
the X-Ca-Request-Id header. We recommend that you record the request
ID in both the client and the back-end service for troubleshooting and
 tracking.
X-Ca-Error-Message: Invalid Url
// The error message returned by API Gateway. When a request fails,
API Gateway returns the error message to the client in the X-Ca-Error-
Message header.
X-Ca-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
// The message can be returned only when the debug mode is enabled.
The message is used only for reference at the debugging stage.
```

When you call an API by using HTTP or HTTPS, the request must include the signature information. For more information about how to calculate and pass the encrypted signature, see *Request signatures*.

1.6 APIs

1.6.1 Limits

Item	Description
Number of API groups	A maximum of 50 API groups can be created for each account.
Number of APIs	A maximum of 200 APIs can be created for each API group. A maximum of 10,000 APIs can be created for each account.
The maximum number of independent domain names that can be bound to an API group	A maximum of five independent domain names can be bound to each API Group.
TPS handled by an API group	Each API group can handle a maximum of 500 TPS. You can raise this limit to suit your business needs.

1.6.2 Manage groups

1.6.2.1 Create a group

You can create an API Group in the API Gateway console.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Groups tab.**
3. **On the Groups tab, click Create Group.**
4. **In the Create Group dialog box that appears, set parameters and click Create.**

   The name of a group must be globally unique. The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.

1.6.2.2 Manage an environment

To understand environment management, you need to be familiar with two concepts: environment and environment variables.

- **An environment is a configuration of an API group. You can configure several environments for a group. APIs that have not been published are considered**

as defined APIs. An API can only provide external services after it has been published to an environment.

- **Environment variables are environment-specific variables that you can create and manage. For example, you create an environment variable in the production environment. The variable name is `Path`. The variable value is `/stage/release`.**

On the API Settings tab, set the `Path` parameter to `#Path#`, which indicates a variable named `Path`.

When you publish the API to a production environment, the value of the `#Path#` variable on the API Settings tab is `/stage/release`.

When you publish the API to another environment that does not have the `#Path#` variable, the variable will be null and the API cannot be called.

Environment variables enable back-end services to be in different runtime environments. You can access various back-end services by configuring the same API definition but different back-end service endpoints and paths across different runtime environments. When you use environment variables, take note of the following points:

- **Variable names are case-sensitive.**
- **If you configure a variable in the API definition, you must configure the name and value of the variable for the environment to which the API is published. Otherwise, the variable will be null and the API cannot be called.**

Create an environment variable

1. *Log on to the API Gateway console*.
2. **Click the Groups tab.**
3. **Select the group, click the management icon in the Actions column corresponding to the group, and choose View Details from the shortcut menu.**
4. **Click the Environment Variables tab.**
5. **Click Create Variable. Enter `Variable Name` and `Variable Value`, and click OK.**

> **Notice:**
> **The variable name for the production, pre-release, and test environments must be the same. However, the variable values for the three environments can be**

different. After an API is published to the corresponding environment, the variable value will be automatically replaced.

Delete an environment variable

1. *Log on to the API Gateway console*.
2. **Click the Groups tab.**
3. **Select the group, click the management icon in the Actions column corresponding to the group, and choose View Details from the shortcut menu.**
4. **Click the Environment Variables tab.**
5. **Select the runtime environment and the variable to be deleted, click the management icon in the Actions column corresponding to the variable, and choose Delete Variable from the shortcut menu.**
6. **In the message that appears, click OK.**

1.6.2.3 Delete a group

You can delete an existing group.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Groups tab.**
3. **Select the group to be deleted, click the management icon in the Actions column corresponding to the group, and choose Delete from the shortcut menu.**
4. **In the dialog box that appears, click OK.**

Note:
Before deleting a group, you must delete APIs that belong to the group.

1.6.3 Create an API

1.6.3.1 Overview

Creating an API is the process of defining the API in the API Gateway console. When creating an API, you must define the basic information, back-end service information, API request information, and response information of the API.

- **API Gateway enables you to configure verification rules for input parameters . API Gateway can be configured to pre-verify and forward API requests that contain valid parameters.**

- API Gateway enables you to configure mappings between front-end and back-end parameters. API Gateway can map a front-end parameter at one location to a back-end parameter at a different location. For example, you can configure API Gateway to map a Query parameter in an API request to a Header parameter in a back-end service request. In this way, you can encapsulate your back-end services into standard API operations.
- API Gateway enables you to configure constant and system parameters. These parameters are not visible to your users. API Gateway can add these parameters to requests based on your business requirements before sending the requests to your back-end services. If you want API Gateway to attach the keyword apigateway to each request that API Gateway forwards to your back-end services, you can configure aligateway as a constant parameter and specify where it is received.

1.6.3.2 Create an API

Procedure

1. *Log on to the API Gateway console*.
2. Click the APIs tab.
3. Click Create API.
4. Specify the basic information of the API and click Next.

Parameter	Description
Groups	The basic management unit of an API. You must create a group before creating an API. When you select a group, you select a region for the API.
API Name	The name of the API to be created.
Authentication Mode	The authentication mode of API requests. Valid values: `Alibaba Cloud Applications` and `None`. • `Alibaba Cloud Applications`: This mode requires the requester to pass the application authentication to call the API. • `None`: This mode allows any user who knows the request definition of the API to initiate a request. API Gateway directly forwards the requests to your back-end service without verifying the identity of the requesters.

Parameter	Description
Signature Algorithm	The algorithm used to sign API requests. Valid values: · HmacSHA256 · HmacSHA1 and HmacSHA256
Description	The description of the API.

5. Define API requests. Define how users call your APIs, including the request type, network protocol, URL suffix, HTTP method, and request mode.

Parameter	Description
Request Type	The request type. Four options are available. However, only the common request type is supported. · `Common Request`: indicates common HTTP or HTTPS requests. · `Logon Request (Bidirectional Communication)`: indicates the bidirectional control signal to register devices. It is sent from the client to the server. · `Logoff Request (Bidirectional Communication)`: indicates the bidirectional control signal to register devices. It is sent from the client to the server. After devices are deregistered, server-to-client notifications are no longer received. · `Server-to-Client Notification (Bidirectional Communication)`: After receiving the registration signal sent from the client, the back-end service records the device ID and send a server-to-client notification to API Gateway. Then, API Gateway sends the notification to the device. As long as the device is online, API Gateway sends the server-to-client notification to the device.
Network protocol	`HTTP`, `HTTPS`, and `WEBSOCKET` are supported in API calls.
URL Suffix	The API request path. It corresponds to the service host . The request path can be different from the actual back-end service path. You can specify any valid and semantically-correct path as the request path. You can configure dynamic parameters in the request path, which requires users to specify path parameters in the request. At the same time, the path parameters can be mapped to query and header parameters that are received by the back-end service.
HTTP Method	You can select `PUT`, `GET`, `POST`, `PATCH`, `DELETE`, and `HEAD`.

Parameter	Description
Request Mode	You can select either `Request Parameter Mapping` or `Request Parameter Passthrough`. · Request Parameter Mapping indicates that you must configure request and response data mappings for query , path, and body form parameters. API Gateway only passes the configured parameters through to the back end. Other parameters are filtered out. · Request Parameter Passthrough indicates that you do not need to configure query and body form parameters, but still must configure path parameters in the Request Parameters section. All parameters sent from the client are passed through by API Gateway to the back-end service.

6. Define request parameters.

   Define the request parameters of your APIs. You can specify different request parameters for different parameter paths. Head, Query, Body, and Path can be selected. When you configure a dynamic path parameter, you must provide a description of this dynamic parameter. Supported parameter types include String, Number, and Boolean.

   · Note that the names of all parameters must be unique.
   · You can use the shortcut key on the left to adjust the parameter order.
   · To delete unwanted parameters, you can click the management icon in the Actions column and choose Delete from the shortcut menu.

7. Configure parameter validation rules.

   To configure validation rules, you can click the management icon in the Actions column and choose Configure Advanced Settings from the shortcut menu.
   For example, you can configure validation rules such as the length of a string and enumeration of numbers. API Gateway pre-verifies requests based on the validation rules. Requests with invalid parameters are not sent to your back-end service. This greatly reduces the workload on your back-end service.

8. Configure your back-end service and click Next.

   This section defines mappings between request and response parameters, and specifies the API configurations of your back-end service. Back-end service configurations include the back-end service URL, URLsuffix, timeout period,

parameter mappings, constants, and system parameters. After receiving requests, API Gateway converts the format of the requests to the format required by the back-end service based on the back-end service configuration. Then, API Gateway forwards the requests to your back-end service.

 **Note:**
You can enter the following parameters: dynamic parameters in the URL suffix , header parameters, query parameters, body parameters (non-binary), constants, and system parameters. The names of these parameters must be globally unique. For example, you cannot specify a parameter of the same name in both the header and query path.

a. Configure the basic settings of the back-end service.

Parameter	Description
Backend Service Type	By default, HTTP or HTTPS service is supported. API Gateway only supports HTTP or HTTPS service. The Mock option indicates that you do not access the back-end service, but expect API Gateway to simulate responses based on the specified values. For more information, see the Mock option.
VPC ID	Do Not Authorize Access to VPCs is selected by default. The back-end service can be connected with API Gateway directly. If the back-end service is deployed in a VPC, and API Gateway needs to access the back-end service, select the corresponding VPC ID for the back-end service.
Backend Service URL	The host name of the back-end service can be a domain name or `http(s)://host:port`. If the back-end service is deployed in a VPC, the back-end service URL can be in the http(s)://{ip}.{vpcId}.gateway.vpc:{port} format. Example: http://172.10.1.3.vpc-12ssar3e123.gateway.vpc:8080.
URL Suffix	The actual request path of your API service on your back-end server. If you want to receive dynamic parameters in the back-end path, you must declare parameter mappings by specifying the locations and names of the corresponding request parameters.
HTTP Method	`PUT`, `GET`, `POST`, `PATCH`, `DELETE`, and `HEAD` can be selected.

Parameter	Description
Timeout	The response time for API Gateway to access the back-end service after API Gateway receives an API request. The response time is from the time when API Gateway sends an API request to the back-end service to the time when API Gateway receives responses returned by the back-end service. The response time cannot exceed 30 seconds. If API Gateway does not receive a response from the back-end service within 30 seconds, API Gateway stops accessing the back-end service and returns an error message.

b. Configure back-end service parameters.

API Gateway can set up mappings between request and response parameters, including name mappings and parameter path mappings. API Gateway can map a request parameter at any location such as path, header, query, or body to a response parameter at a different location. In this way, you can package your back-end services into standard APIs. This section describes the mappings between front-end APIs and back-end APIs.

Note:
The request and response parameters must be globally unique.

c. Configure constant parameters.

If you want API Gateway to attach the apigateway tag to each request that API Gateway forwards to your back-end service, you can configure this tag as a constant. Constants are not visible to your users. After API Gateway receives requests, API Gateway automatically adds constants to the specified locations and then forwards the requests to your back-end service.

d. Configure system parameters.

By default, API Gateway does not send its system parameters to you. However, if you need the system parameters, you can configure their locations and names. The following table lists the system parameters.

Parameter	Description
CaClientIp	The IP address of the client sending a request.
CaDomain	The domain name from which a request originates.

Parameter	Description
CaRequestHandleTime	The request time. It must be in GMT.
CaAppId	The ID of the application sending a request.
CaRequestId	The unique ID of the request.
CaApiName	The name of the API.
CaHttpSchema	The protocol that is used to call the API. The protocol can be either HTTP or HTTPS.
CaProxy	The proxy (AliCloudApiGateway).

9. Define responses and click Create.

   You can set Response ContentType, Success Result Example, Error Response Example, and Error Codes. API Gateway does not parse responses, but forwards them to API users.

1.6.3.3 Security authentication

The security authentication methods that are supported by API Gateway include Alibaba Cloud applications and none.

- Alibaba cloud applications: An application must be authorized by the API provider to call an API. An API caller must provide an AppKey and encrypted signature. Otherwise, the API request validation will fail. For more information about the signature method, see *Encrypt a signature*.
- None: The API can be called without authorization after it is published. The AppKey and encrypted signature are not required when you make an API request .

1.6.3.4 Network protocol

HTTPS domain names that are configured in the API Gateway console are invalid because the console does not support HTTPS. To use an HTTPS domain name, you can call the API operations of API Gateway.

Find the API on the API tab, click the management icon in the Actions column corresponding to the API, and choose Change from the shortcut menu. In the Request Basic Settings section of the API Settings page, change the network protocol.

You can send API requests by using any of the following network protocols:

- **HTTP: You can send API requests over HTTP.**
- **HTTPS: You can send API requests over HTTPS. HTTPS cannot be configured in the API Gateway console and will be supported in later versions.**
- **WEBSOCKET: You can send API requests over WebSocket.**

1.6.3.5 Request body configuration

You can configure the request body when the HTTP method is POST, PUT, or PATCH. You can use two methods to configure the request body. The following two methods are mutually exclusive.

- **Form-based request body: Add a request parameter and select Body as the parameter path. The configured request body can only be used to transmit form data. Form data occupies space in the URL request. The URL request cannot exceed 258 KB in size.**
- **Non-form-based request body: If the body content to be transmitted is in JSON or XML format, select "The body content description should be a non-form data. For example, JSON string, binary files, and others". The size of the request body cannot exceed 8 MB.**

Note:

API Gateway cannot upload files through HTTP multipart requests. To upload a file, you can convert the file into a Base64 string and add the string to the request body.

1.6.3.6 VPC ID

If the back-end service is deployed in a classic network, select Do Not Authorize Access to VPCs.

If the back-end service is deployed in a VPC, select a VPC ID. After selecting the VPC ID, you must enter the back-end service URL in the format of http://ip:port. ip indicates the IP address of the back-end service.

1.6.3.7 Configure an API in mock mode

Typically, business partners work together to develop a project. However, the interdependence among business partners restricts them in the project development process. Misunderstandings also can affect the development progress or even delay the project. Therefore, the mock mode is used to simulate the

predetermined API responses in the project development process. This can help reduce misunderstanding and improve development efficiency.

API Gateway provides a simplified configuration process of an API in mock mode.

Configure a mock API

Click the APIs tab. Select an API and click Change in the Actions column to go to the Change API page.

On the Change API page, choose Define Backend Service to set the API to mock mode.

1. **Set the back-end service type to Mock.**
2. **Enter a mock response.**

   Enter your responses as the mock response body. Responses can be in JSON, XML , and text formats. Example:

   ```
    {
   "result": {
       "title": " Mock test for API Gateway",
   }
   }
   ```

   Save the mock settings. Publish the API to the test or release environment for testing.
3. **Enter an HTTP status code for the mock response. Enter 200 to indicate a successful API request.**

Take an API out of mock mode

To take an API out of mock mode, you can change the back-end service type from Mock to HTTP or HTTPS service. This change takes effect only after the API is published.

1.6.3.8 Return the Content-Type header

The value of the Content-Type header is only used to generate API documentation. It does not affect responses returned by the back-end service. The Content-Type header is returned by the back-end service.

1.6.4 API management

1.6.4.1 View and modify an API

You can view and modify an API as needed.

> **Note:**
> **If you modify an API that has been published, modifications made to an API that has been published will not affect its operations. You must re-publish the modified API before synchronizing the changes to the release environment.**

Procedure

1. *Log on to the API Gateway console*.
2. **Click the APIs tab.**
3. **Select the API that you want to view and click the Management icon in the Actions column corresponding to the API.**
   - **Choose View Details from the shortcut menu. On the API Settings page, you can view the information of the API.**
   - **Choose Change from the shortcut menu. Modify the API as needed and click Change in the lower right corner.**

     The procedure to create an API is similar to that of modifying an API. For more information about how to create an API, see *Create API*. If you want to cancel the modifications before they are submitted, click the Back icon in the upper-left corner of the Change API page.

1.6.4.2 Publish an API

After you create an API, you must publish the API to the test, pre-release, or release environment before it can be called.

- **When you use a second-level or independent domain name to access an API that has been published to an environment, you must specify the environment in the request header.**

- If you attempt to publish an API that already has a running version in the test or release environment, the previously running version will be overwritten by the new API. All historical versions and definitions are recorded, allowing you to roll back the API to previous versions as needed.
- You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API.

Step 1: Test the API

To simulate API requests, you can create an application and authorize the application to call your API.

You can compile code based on actual scenarios, or use the SDK samples provided by API Gateway to call the API.

You can publish the API to the test or release environment. If no independent domain name is bound to the group to which the API belongs, you can test or call the API by using the second-level domain name. When you make an API request, specify the environment of the API by setting the X-Ca-Stage header to TEST, PRE, or RELEASE. If you do not specify the header, the API will be published to the release environment.

Step 2: Publish the API

After testing the API, you can publish it.

API Gateway enables you to manage versions of APIs in the test or release environment. You can publish or unpublish an API, and switch the versions of an API. The switch of versions takes effect in real time.

1. *Log on to the API Gateway console*.
2. Click the API tab.
3. Select the API that you want to publish, click the management icon in the Actions column corresponding to the API, and choose Publish from the shortcut menu.
4. In the dialog box that appears, select the environment where the API is published, enter a description, and click OK.

1.6.4.3 Authorize an application

You must authorize an application before it can call an API. After publishing an API to the release environment, you must authorize the user applications to call the API. You can authorize an application to call an API or revoke the authorization of an application to call an API. API Gateway verifies the authorization relationship.

Note:

- **You can authorize one or more applications to use one or more APIs.**
- **If an API has been published to both the test and release environments and the test environment has been selected, applications are only authorized to call the API in the test environment.**
- **You can locate an application based on its ID provided by a user.**
- **If you want to revoke the authorization of an application to call an API, go to the Authorization tab of the API. Then, select the application, click the management icon in the Actions column corresponding to the application, and choose Deauthorize from the shortcut menu, or click Deauthorize in the upper-right corner.**

An application indicates the identity of a requester. Before you or your users test or call APIs, you or your users must create an application that is used as the identity of a requester. Then, you need to authorize the application to call an API.

Note:
Authorizations are environment-specific. If you want to use an application to call an API in both the test and release environments, you must authorize the application in both environments separately. Otherwise, errors may occur due to the inconsistency between the authorized environment and requested environment.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the APIs tab.**
3. **Select the API that an application is authorized to call, click the management icon in the Actions column corresponding to the API, and choose Authorize from the shortcut menu. The Authorize Applications dialog box appears.**

4. Select the environment and the application to authorize.

   The system automatically displays the applications that belong to your account. If you want to authorize an application that belongs to a different account, search for the application by application ID.

5. Select the application to authorize, and click the > icon to add the selected application to the right-side list.
6. Click OK to complete the authorization process.

1.6.4.4 Revoke authorization

You can revoke the authorization of an application to call an API.

> **Note:**
> Before deleting a published API, you must unpublish it. Otherwise, you cannot delete this API.

Procedure

1. *Log on to the API Gateway console*.
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Authorization tab to view all the applications that are authorized to call the API.
5. Select an authorized application, click the management icon in the Actions column corresponding to the application, and choose Deauthorize from the shortcut menu.
6. In the dialog box that appears, click OK.

1.6.4.5 Unpublish an API

You can unpublish APIs that have been published.

You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API. For more information, see *Delete API definitions*.

Procedure

1. *Log on to the API Gateway console*.
2. Click the API tab.
3. Select the API that you want to unpublish, click the management icon in the Actions column corresponding to the API, and choose Take Offline from the shortcut menu.
4. In the dialog box that appears, select the environment that you want to unpublish the API from, and click OK.

1.6.4.6 View the version history of an API

You can view the version history of an API, including the version number, description, environment, release time, and specific definition of each version.

Procedure

1. *Log on to the API Gateway console*.
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Version History tab.
5. Select the version that you want to view the history of, click the management icon in the Actions column corresponding to the version, and choose View from the shortcut menu. You can see the details of this API version.

1.6.4.7 Change the version of an API

When viewing the version history of an API, you can select a different version and switch to this version. The selected version then replaces the previous version and takes effect in the specified environment.

Procedure

1. *Log on to the API Gateway console*.
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Version History tab.
5. Select the version, click the management icon in the Actions column corresponding to the version, and choose Switch to This Version from the shortcut menu.

6. **In the dialog box that appears, enter a description and click OK.**

1.6.5 Plug-in management

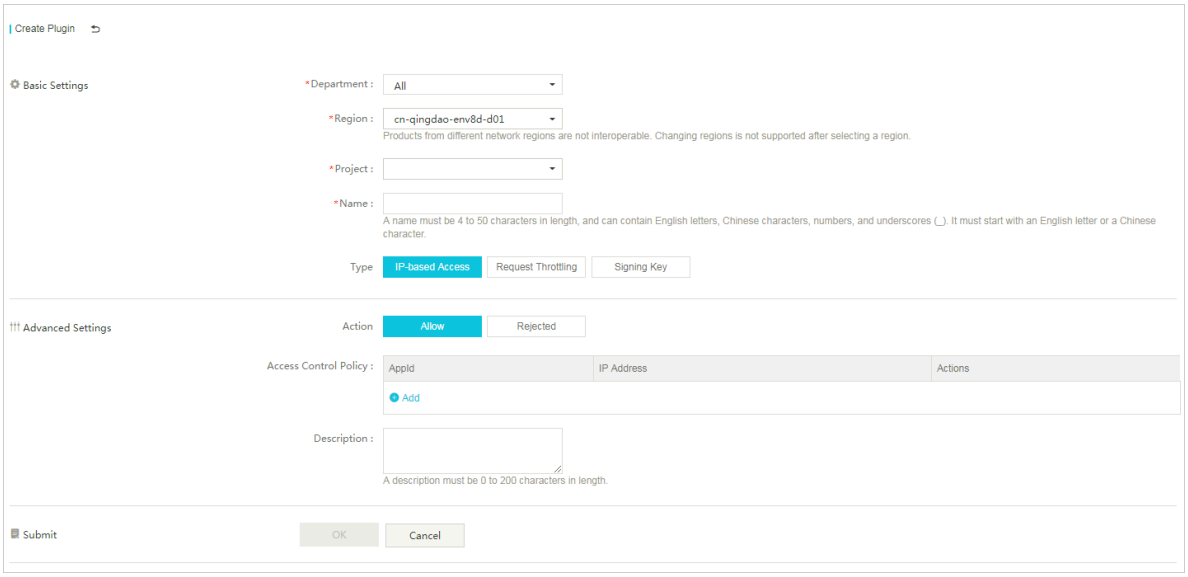
1.6.5.1 Create a plug-in

1.6.5.1.1 Create an IP-based access control plug-in

IP-based access control can help API providers configure an IP whitelist or blacklist for API access. The following steps describe how to create an IP-based access control plug-in.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Plug-ins tab.**
3. **Click Create Plugin. On the Create Plugin page, set Type to IP-based Access Control.**



Parameter	Description
Action	· Allow: Only requests that comply with the IP-based access control policy are allowed. All other requests are rejected. · Reject: Requests that comply with the IP-based access control policy are rejected. All other requests are allowed.

Parameter	Description
AppId	The ID of the application making requests. This parameter is optional. If this value is null, this access control policy is applied to all applications. If this value is not null, this access control policy is only applied to the specified application. When Action is set to Reject, requests from the specified application are ignored.
IP Address	Required. The IP address of the API request. Note: The IP address used to access API Gateway is an egress IP address. The format can be an IP address or an IP address segment. Example: 10.1.1.1 or 11.0.0.0/8.

4. After configuring the parameters, click OK.

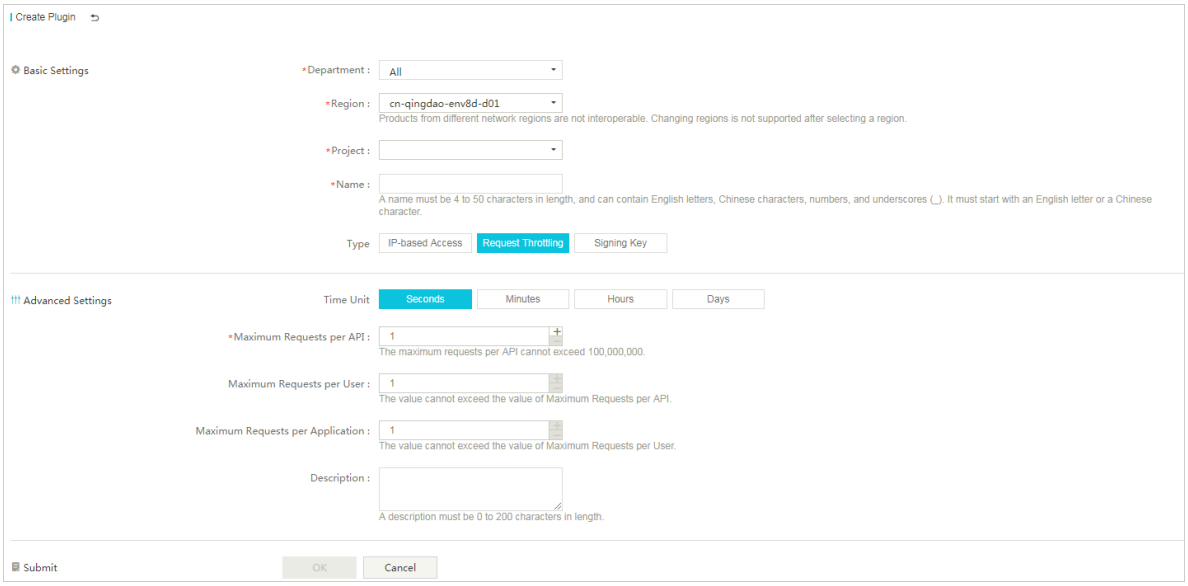
1.6.5.1.2 Create a throttling plug-in

You can use a throttling plug-in to limit the number of API requests. The throttling plug-in helps prevent the back-end service from being overwhelmed by large numbers of API requests. The following steps describe how to create a throttling plug-in.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Plug-ins tab.

3. Click Create Plugin. On the Create Plugin page, set Type to Request Throttling.



Parameter	Description
Time Unit	The time granularity for throttling operations. Valid values: seconds, minutes, hours, and days.
Maximum Requests per API	The maximum number of times each API can be called per unit time. This parameter is set based on the back-end service capability.
Maximum Requests per User	The maximum number of requests that each user can make per unit time. The value of this parameter cannot exceed the value of Maximum Requests per API.
Maximum Requests per Application	The maximum number of requests that each application can make per unit time. The value of this parameter cannot exceed the value of Maximum Requests per User.

4. After configuring the parameters, click OK.

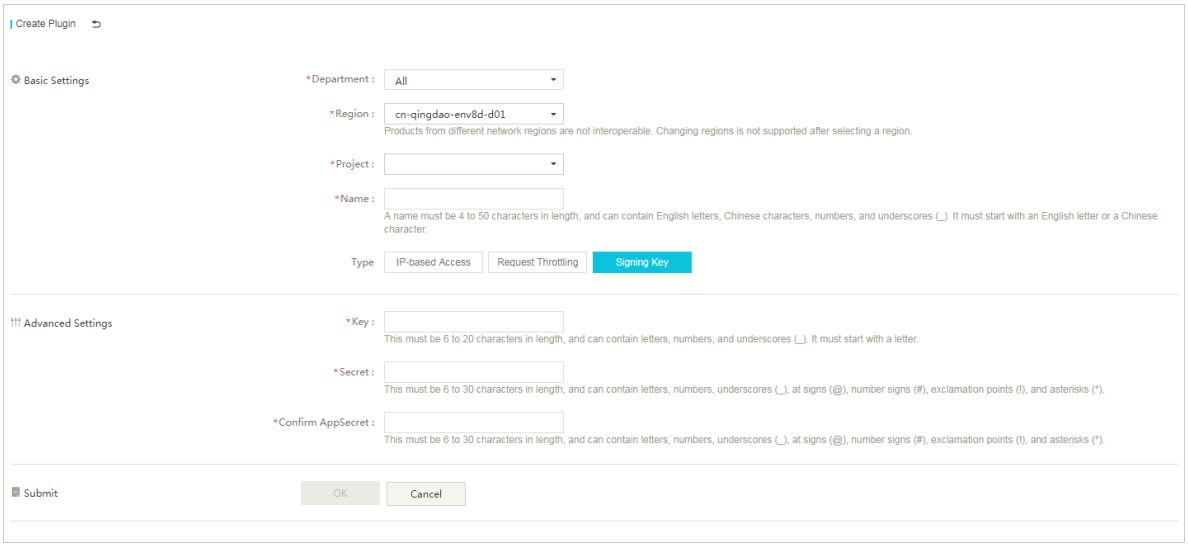
1.6.5.1.3 Create a signature key plug-in

A signature key plug-in is used for the signature verification between API Gateway and back-end services. You can create a signature key that consists of a key and a secret. The key and secret are equivalent to the account and password issued to API Gateway. When API Gateway sends a request to your back-end service, API Gateway uses the signature key to calculate a signature string and passes the signature string to your back-end service. Your back-end service obtains the signature string for

symmetric calculation to verify the identity of API Gateway. The following steps describe how to create a signature key plug-in.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Plug-ins tab.
3. Click Create Plugin. On the Create Plugin page, set Type to Signing Key.



Configure the plug-in parameters as needed.

4. After configuring the parameters, click OK.

1.6.5.2 Bind a plug-in to an API

After you create a plug-in, you must bind the plug-in to an API for the plug-in to take effect.

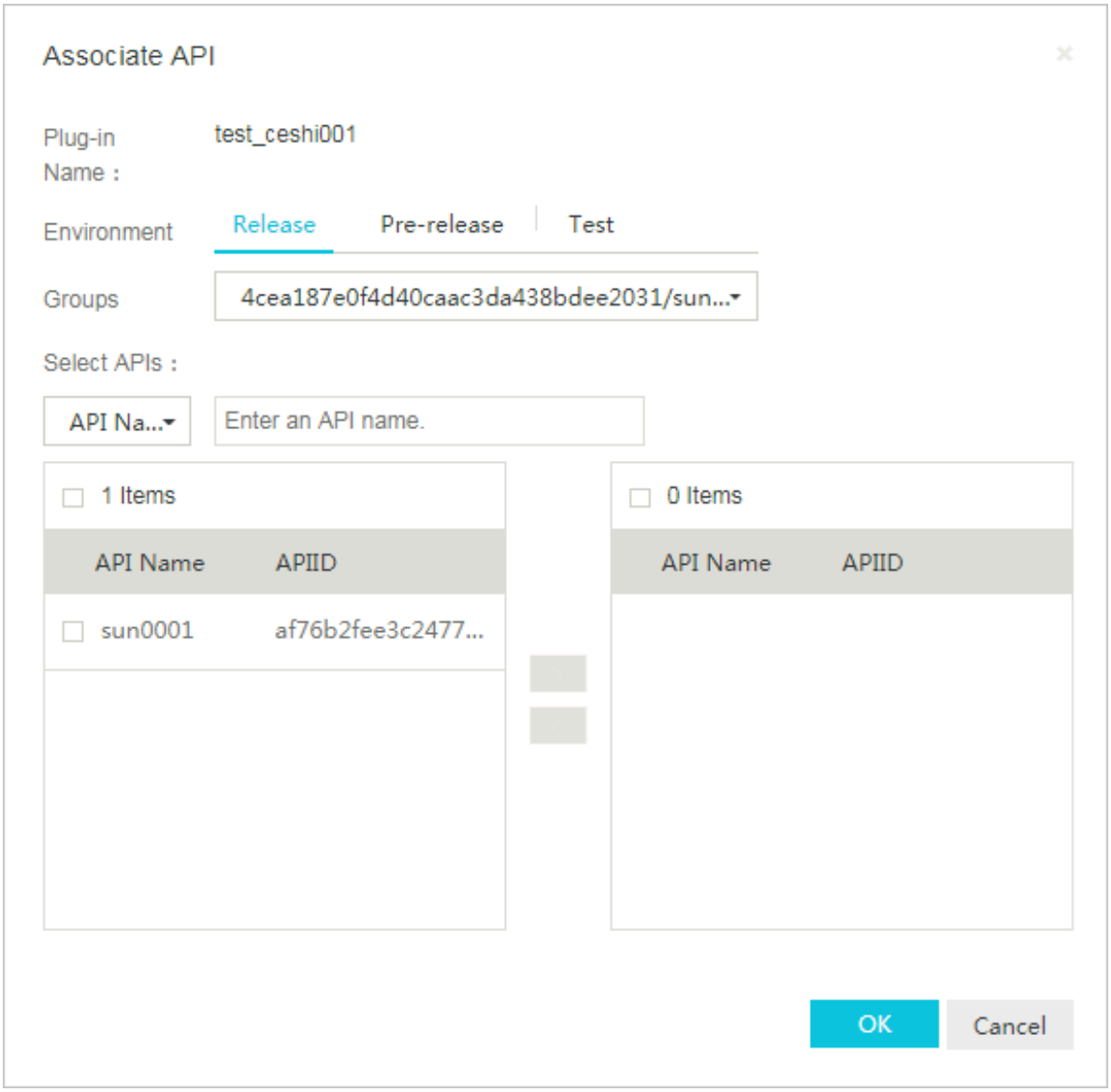
Context

You can bind a plug-in to multiple APIs. The plug-in will take effect on each API individually. Only one plug-in of each type can be bound to an API. If you bind a plug-in to an API that already has a bound plug-in of the same type, the new plug-in will replace the previous one and take effect.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Plug-ins tab.
3. Select the plug-in to be bound, click the management icon in the Actions column corresponding to the plug-in, and choose Associate API from the shortcut menu.
4. Select the environment and group of the API.

5. Select the API that you want to add, click the > icon to add the selected API to the right-side list, and click OK.



1.6.5.3 Delete a plug-in

You can delete existing plug-ins.

Procedure

1. *Log on to the API Gateway console*.
2. Click the Plug-ins tab.
3. Select the plug-in that you want to delete, click the management icon in the Actions column corresponding to the plug-in, and choose Delete Plug-in from the shortcut menu.
4. In the dialog box that appears, click OK.

1.6.5.4 Unbind a plug-in

You can unbind plug-ins from the APIs that they are bound to.

Procedure

1. *Log on to the API Gateway console*.
2. **Click the Plug-ins tab.**
3. **Select the plug-in that you want to unbind, click the management icon in the Actions column corresponding to the plug-in, and choose View Details from the shortcut menu.**
4. **On the Plug-in Management page that appears, click the management icon in the Actions column corresponding to the plug-in, and choose Unbind from the shortcut menu.**
5. **In the dialog box that appears, click OK.**