

# API Gateway User Guide

## User Guide

**Issue** 01  
**Date** 2022-08-17



**Copyright © Huawei Technologies Co., Ltd. 2022. 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 Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks 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 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.

---

# Contents

---

<b>1 Overview</b>	<b>1</b>
<b>2 Gateway Management</b>	<b>5</b>
2.1 Creating a Dedicated Gateway	5
2.2 Modifying a Dedicated Gateway	8
<b>3 API Group Management</b>	<b>13</b>
3.1 Creating an API Group	13
3.2 Binding a Domain Name	14
3.3 Deleting an API Group	16
3.4 Adding a Gateway Response	17
<b>4 API Management</b>	<b>20</b>
4.1 Creating an API	20
4.2 CORS	33
4.3 Debugging an API	39
4.4 Authorizing Apps to Call an API	41
4.5 Publishing an API	42
4.6 Taking an API Offline	45
4.7 Deleting an API	46
4.8 Importing APIs	46
4.9 Exporting APIs	48
<b>5 Request Throttling</b>	<b>50</b>
5.1 Creating a Request Throttling Policy	50
5.2 Deleting a Request Throttling Policy	53
5.3 Adding an Excluded App or Tenant	54
5.4 Removing an Excluded App or Tenant	56
<b>6 Access Control</b>	<b>58</b>
6.1 Creating an Access Control Policy	58
6.2 Deleting an Access Control Policy	60
<b>7 Environment Management</b>	<b>62</b>
7.1 Creating an Environment and Environment Variable	62
7.2 Deleting an Environment	65
<b>8 Signature Key Management</b>	<b>66</b>

8.1 Creating and Using a Signature Key.....	66
8.2 Deleting a Signature Key.....	69
<b>9 VPC Channel Management.....</b>	<b>70</b>
9.1 Creating a VPC Channel.....	70
9.2 Deleting a VPC Channel.....	73
9.3 Editing Health Check Configurations.....	74
9.4 Editing Cloud Server Configurations of a VPC Channel.....	76
<b>10 Custom Authorizers.....</b>	<b>78</b>
10.1 Creating a Custom Authorizer.....	78
10.2 Deleting a Custom Authorizer.....	81
<b>11 Plug-ins.....</b>	<b>83</b>
11.1 Creating a Plug-in.....	83
11.2 CORS Plug-in.....	85
11.3 HTTP Response Header Management Plug-in.....	86
11.4 Deleting a Plug-in.....	89
<b>12 Monitoring.....</b>	<b>90</b>
12.1 API Gateway Metrics.....	90
12.2 Creating Alarm Rules.....	92
12.3 Viewing Metrics.....	93
<b>13 App Management.....</b>	<b>94</b>
13.1 Creating an App and Obtaining Authorization.....	94
13.2 Deleting an App.....	96
13.3 Resetting the AppSecret of an App.....	96
13.4 Adding an AppCode for Simple Authentication.....	97
13.5 Viewing API Details.....	98
<b>14 Log Analysis.....</b>	<b>100</b>
<b>15 SDKs.....</b>	<b>103</b>
<b>16 Calling Published APIs.....</b>	<b>104</b>
16.1 Calling APIs.....	104
16.2 Response Headers.....	106
16.3 Error Codes.....	108
<b>17 Permissions Management.....</b>	<b>115</b>
17.1 Creating a User and Granting API Gateway Permissions.....	115
17.2 API Gateway Custom Policies.....	117
<b>18 Key Operations Recorded by CTS.....</b>	<b>119</b>
18.1 API Gateway operations that can be recorded by CTS.....	119
18.2 Querying Audit Logs.....	123

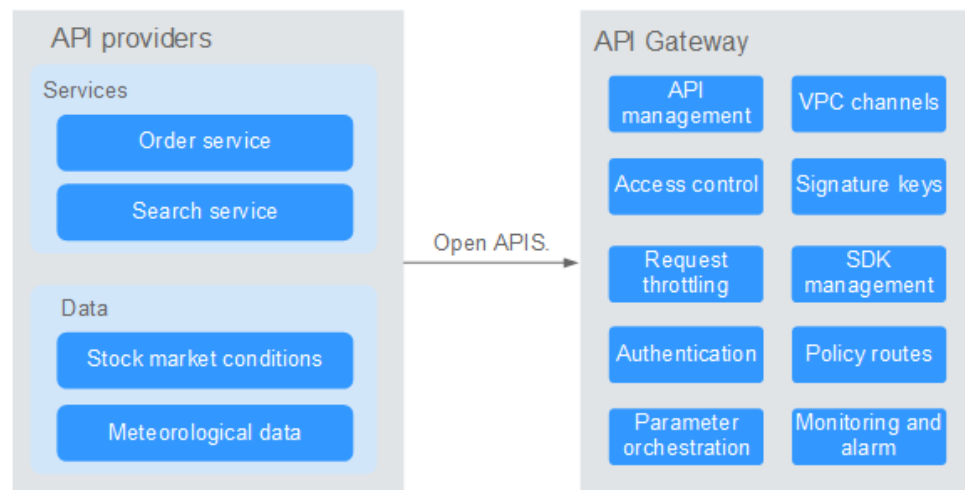
# 1 Overview

API Gateway is a fully managed service that enables you to securely build, manage, and deploy APIs at any scale with high performance and availability. With API Gateway, you can easily integrate your internal service systems and selectively expose your service capabilities through its API opening and calling functions.

- **API Opening**

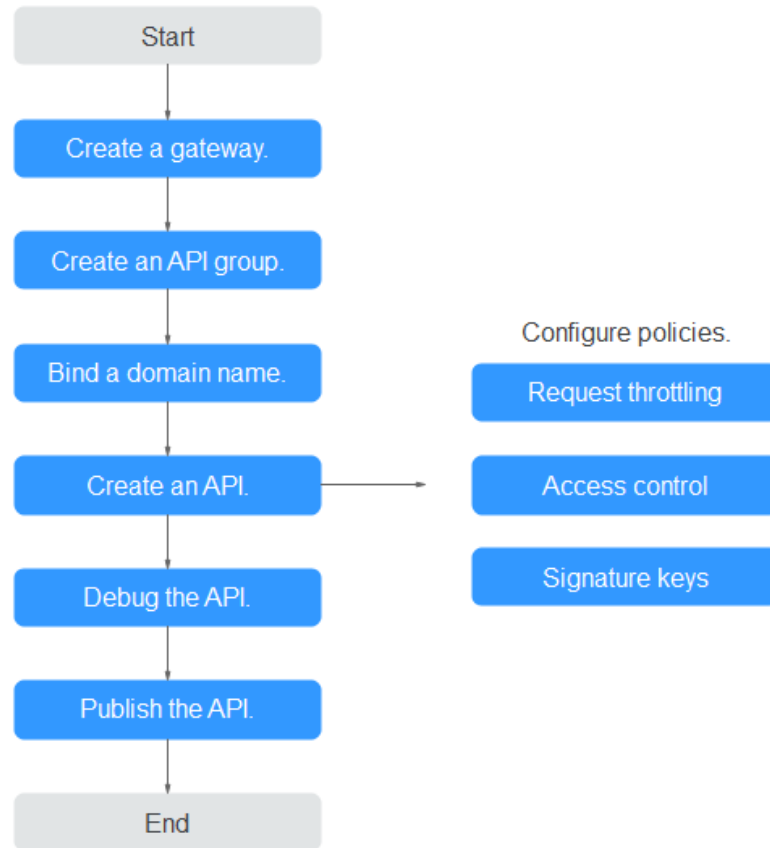
Enterprises and developers selectively expose their services and data through API Gateway.

**Figure 1-1** API opening



The following figure shows the API opening process.

**Figure 1-2** API opening process



- a. Create a gateway.  
**Create a dedicated gateway.**
- b. **Create an API group.**  
Each API belongs to an API group. Create a group before creating an API.
- c. **Bind a domain name.**  
Before exposing an API, bind an independent domain name to the group so that users can access the API.  
You can debug the API using the default subdomain name allocated to the group to which the API belongs. The subdomain name can be called a maximum of 1000 times every day.
- d. **Create an API.**  
Encapsulate existing backend services into standard RESTful APIs and expose them to external systems.  
After creating an API, configure the following settings to control API access:
  - **Request throttling**  
Set the maximum number of times the API can be called within a time period to protect backend services.
  - **Access control**  
Set a blacklist or whitelist to deny or allow API access from specific IP addresses or accounts.



- b. **Create an app.**  
For an API that uses app authentication, create an app to generate an AppKey and AppSecret. Bind the app to the API so that you can call the API through app authentication.
- c. **Obtain an SDK.**  
Use the SDK to generate a signature for the AK/SK and call the API.
- d. **Call the API.**  
Obtain the API using its access address and perform authentication based on its authentication mode.



# 2 Gateway Management

---

## 2.1 Creating a Dedicated Gateway

This section describes how to create a dedicated gateway. You can create APIs and use them to provide services only after a dedicated gateway is created.

### Information on Creating a Dedicated Gateway

There are some limitations on creating a dedicated gateway. If you cannot create a dedicated gateway or a gateway fails to be created, check the following items:


- Gateway quota  
By default, your account can be used to create five dedicated gateways in a project. To create more dedicated gateways, submit a service ticket to increase the quota.
- Permissions  
You must be assigned both the **APIG Administrator** and **VPC Administrator** roles so that you can create dedicated gateways.
- Number of available private IP addresses in the subnet  
The basic, professional, enterprise, and platinum editions of API Gateway require 3, 5, 6, and 7 private IP addresses in a subnet, respectively. Ensure that the subnet you choose has sufficient private IP addresses on the Virtual Private Cloud (VPC) console.

### Network Environment

- VPC  
VPC Dedicated gateways are deployed in VPCs. Cloud resources, such as Elastic Cloud Servers (ECSs), in the same VPC can call APIs using the private IP address of the dedicated gateway deployed in the VPC.  
You are advised to deploy your dedicated gateways in the same VPC as your other services to facilitate network configuration and secure network access.


#### NOTE

VPCs of dedicated gateways cannot be modified.

- **EIP**  
To allow public inbound access to the APIs deployed in a dedicated gateway, create an Elastic IP (EIP) and bind it to the dedicated gateway.  
 **NOTE**  
For APIs whose backend services are deployed on a public network, API Gateway automatically generates an IP address for public outbound access and you do not need to create an EIP.
- **Security group**  
Similar to a firewall, a security group controls access to a gateway through a specific port and transmission of communication data from the gateway to a specific destination address. For security purposes, create inbound rules for the security group to allow access only on specific ports.  
The security group bound to a dedicated gateway must meet the following requirements:
  - **Inbound access:** To allow the APIs in the dedicated gateway to be accessed over public networks or from other security groups, add inbound rules for the security group to allow access on ports 80 (HTTP) and 443 (HTTPS).
  - **Outbound access:** If the backend service of an API is deployed on a public network or in another security group, add outbound rules for the security group to allow access to the backend service address through the API calling port.
  - If the frontend and backend services of an API are bound with the same security group and VPC as the dedicated gateway, no inbound or outbound rules are needed to allow access through the preceding ports.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**.

**Step 4** Click **Create Dedicated Gateway**. The following table describes the parameters.

**Table 2-1** Parameters for creating a dedicated gateway

Parameter	Description
Region	A geographic area where the gateway will be deployed. Deploy the gateway in the same region as your other services to allow all services to communicate with each other through subnets within a VPC. This reduces public bandwidth costs and network latency.

Parameter	Description
AZ	<p>A physical region where resources use independent power supplies and networks. Availability zones (AZs) are physically isolated but interconnected through an internal network.</p> <p>To enhance gateway availability, deploy the gateway in multiple AZs.</p>
Gateway Name	Gateway name.
Edition	<p>The basic, professional, enterprise, and platinum editions are available. The number of concurrent requests allowed varies depending on the gateway edition. For more information, see <a href="#">Specifications</a>.</p>
Scheduled Maintenance	<p>Time period when the gateway can be maintained. The technical support personnel will contact you before maintenance.</p> <p>Select a time period with low service demands.</p>
Public Inbound Access	<p>Determine whether to allow the APIs created in the dedicated gateway to be called by external services using an EIP. To enable this function, assign an EIP to the dedicated gateway.</p> <p>APIs in the dedicated gateway can be called using independent domain names or subdomain names. There is a limitation on the number of times that APIs in an API group can be called per day using the subdomain name. To overcome the limitation, bind independent domain names to the API group and ensure that the independent domain names have already been CNAMEd to the EIP of the dedicated gateway to which the API group belongs.</p> <p>For example, you have an HTTPS API (path: <b>/apidemo</b>) with public access enabled. The API can be called using "<b>https://{domain}/apidemo</b>", where <i>domain</i> indicates an independent domain name bound to the API group to which the API belongs. The independent domain name must have already been CNAMEd to the EIP of the dedicated gateway. The default port is 443.</p>
Public Outbound Access	<p>Determine whether to allow backend services of the APIs created in the dedicated gateway to be deployed on public networks. If you enable this option, set a bandwidth that meets your service requirements. The bandwidth ranges from 1 to 2000 Mbit/s and will be billed by hour based on the pricing of the EIP service.</p>

Parameter	Description
Network	Select a VPC and subnet for the dedicated gateway. Cloud resources (such as ECSs) within the same VPC can call APIs using the private IP address of the dedicated gateway. Deploy the dedicated gateway in the same VPC as your other services to facilitate network configuration and secure network access.
Security Group	Select a security group to control inbound and outbound access. If the backend service of an API is deployed on an external network, configure security group rules to allow access to the backend service address through the API calling port. <b>NOTE</b> If public inbound access is enabled, add inbound rules for the security group to allow access on ports 80 (HTTP) and 443 (HTTPS).
Description	Description of the gateway.

**Step 5** Click **Next**.

**Step 6** If the gateway configurations are correct, and submit the request. The creation progress is displayed on the interface.

----End

## Follow-Up Operations

After the gateway is created, you can create and manage APIs on the console of the gateway. The **Gateway Information** page shows the gateway details, network configurations, API resources, and metrics.

You can modify the gateway name, description, scheduled maintenance time window, security group, and EIP.


## 2.2 Modifying a Dedicated Gateway

You can modify the basic information and configuration parameters of dedicated gateways.

### Modifying Basic Information

To modify the basic information about a dedicated gateway, do as follows:

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**.

**Step 4** Click **Access Console** in the upper right corner of the dedicated gateway you want to modify.

**Step 5** On the **Basic Information** tab page, modify the basic information.


**Table 2-2** Basic information about a dedicated gateway

Parameter	Description
Gateway Name	Name of the gateway.
Description	Description of the gateway.
Scheduled Maintenance	Time period when the gateway can be maintained by technical support personnel. The technical support personnel will contact you if any maintenance activity is going to take place during the window. Select a time period with low service demands.
Security Group	Select a security group to control inbound and outbound access.  If the backend service of an API is deployed on an external network, configure security group rules to allow access to the backend service address through the API calling port. <b>NOTE</b> <ul style="list-style-type: none"><li>If you change the security group, the new security group must meet the requirements for calling APIs included in the dedicated gateway and accessing the backend services of these APIs.</li><li>If public inbound access is enabled, add inbound rules for the security group to allow access on ports 80 (HTTP) and 443 (HTTPS).</li></ul>
EIP	Determine whether to allow the APIs created in the dedicated gateway to be called by external services using an EIP. To enable this function, assign an EIP to the dedicated gateway.  APIs in the dedicated gateway can be called using independent domain names or subdomain names. There is a limitation on the number of times that APIs in an API group can be called per day using the subdomain name.  To overcome the limitation, bind independent domain names to the API group and ensure that the independent domain names have already been CNAMEd to the EIP of the dedicated gateway to which the API group belongs.
Outbound Access	Determine whether to allow API backend services to be deployed on public networks and accessed using the IP address automatically generated by API Gateway. You can enable or disable outbound access at any time.

Parameter	Description
Bandwidth	The bandwidth is billed by hour based on the rate of the EIP service.
Routes	Configure routes at your premises if the subnet of your data center is within the following three segments: 10.0.0.0/8-24, 172.16.0.0/12-24, and 192.168.0.0/16-24.

----End

## Modifying Configuration Parameters

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**.
- Step 4** Click **Access Console** in the upper right corner of the dedicated gateway you want to modify.
- Step 5** Click the **Configuration Parameters** tab, and click **Edit** in the row containing the parameter you want to modify.

**Table 2-3** Configuration parameters

Parameter Name	Description
ratelimit_api_limits	Default request throttling value applied to all APIs. The total number of times an API can be called is determined by this parameter only if no request throttling policy is bound to the API. The <b>Max. API Requests</b> of a request throttling policy cannot exceed the value of this parameter.
request_body_size	The maximum body size allowed for an API request.
backend_timeout	Backend response timeout. Value range: 1 ms to 600,000 ms.

Parameter Name	Description
app_token	<p>Determine whether to enable app_token authentication. If you enable this function, an access_token can be added to the API request for authentication.</p> <ul style="list-style-type: none"><li>• <b>app_token_expire_time</b>: the validity period of an access_token. A new access_token must be obtained before the original access_token expires.</li><li>• <b>refresh_token_expire_time</b>: the validity period of a refresh_token. A refresh_token is used to obtain a new access_token.</li><li>• <b>app_token_uri</b>: the URI used to obtain an access_token.</li><li>• <b>app_token_key</b>: the encryption key of an access token.</li></ul>
app_basic	<p>Determine whether to enable app_basic authentication. After this option is enabled, users can add the header parameter <b>Authorization</b> and set the parameter value to "Basic + base64 (<i>appkey</i> + : + <i>appsecret</i>)", in which <i>appkey</i> and <i>appsecret</i> are the key and secret of an app or the AppKey and AppSecret of a client.</p>
app_secret	<p>Determine whether to enable app_secret authentication. If you enable this function, the <b>X-HW-ID</b> and <b>X-HW-AppKey</b> parameters can be added to the API request to carry the key and secret of an app (the AppKey and AppSecret of a client) for authentication.</p> <p>If you want to enable app_secret authentication, app_api_key authentication must be disabled.</p>
app_route	<p>Determine whether to support IP address-based API access. If you enable this function, APIs that use app authentication in any group except <b>DEFAULT</b> can be called using IP addresses.</p>
backend_client_certificate	<p>Determine whether to enable backend two-way authentication. If you enable this function, you can configure two-way authentication for a backend when creating an API.</p>
ssl_ciphers	<p>Supported HTTPS cipher suites. Select cipher suites that meet your requirements.</p>

Parameter Name	Description
real_ip_from_xff	<p>Determine whether to use the IP addresses in the <b>X-Forwarded-For</b> header for access control and request throttling.</p> <p><b>xff_index</b>: Sequence number of an IP address in the <b>X-Forwarded-For</b> header. The value can be positive, negative, or 0.</p> <ul style="list-style-type: none"><li>• If the value is 0 or positive, the IP address of the corresponding index in the <b>X-Forwarded-For</b> header will be obtained.</li><li>• If the value is negative, the IP address of the indicated reverse sequence in the <b>X-Forwarded-For</b> header will be obtained.</li></ul> <p>For example, assume that the <b>X-Forwarded-For</b> header of a request received by API gateway contains three IP addresses: IP1, IP2, and IP3. If the value of <b>xff_index</b> is 0, IP1 is obtained. If the value is 1, IP2 is obtained. If the value is -1, IP3 is obtained. If the value is -2, IP2 is obtained.</p>

----End



# 3 API Group Management

## 3.1 Creating an API Group

### Scenario


Before creating an API, you must create an API group. An API group contains different APIs used for the same service.

 **NOTE**

Each API can only belong to one API group.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > API Groups**.

**Step 5** Click **Create API Group**, and set the parameters described in [Table 3-1](#).

**Table 3-1** Parameters for creating an API group

Parameter	Description
Name	API group name.
Description	Description of the API group.

**Step 6** Click **OK**.

After the API group is created, it is displayed in the API group list.

 NOTE

- The system automatically allocates a subdomain name to the API group for internal testing. The subdomain name can be accessed 1000 times a day.
- To make your APIs available for users to access, bind independent domain names to the API group to which the APIs belong.

----End

## Follow-Up Operations

After the API group is created, bind independent domain names to it so that API callers can use the domain names to call APIs in the group. For more information, see [Binding a Domain Name](#).

## 3.2 Binding a Domain Name

### Scenario

Before you open an API, you must bind one or more independent domain names to the group to which the API belongs. If no domain names are bound to the group, the API will be called using the default subdomain name of the group and can be called only 1000 times a day.

 NOTE

- In a dedicated gateway, you cannot bind the same independent domain name to different API groups.

Note the following points before you bind a domain name:


- Subdomain name: After an API group is created, the system automatically allocates a unique subdomain name to it for internal testing. The subdomain name can be accessed 1000 times a day, but it cannot be modified.
- Independent domain name: An independent domain name is a custom domain name used for API callers to call open APIs in the group to which the domain name is bound.

### Prerequisites

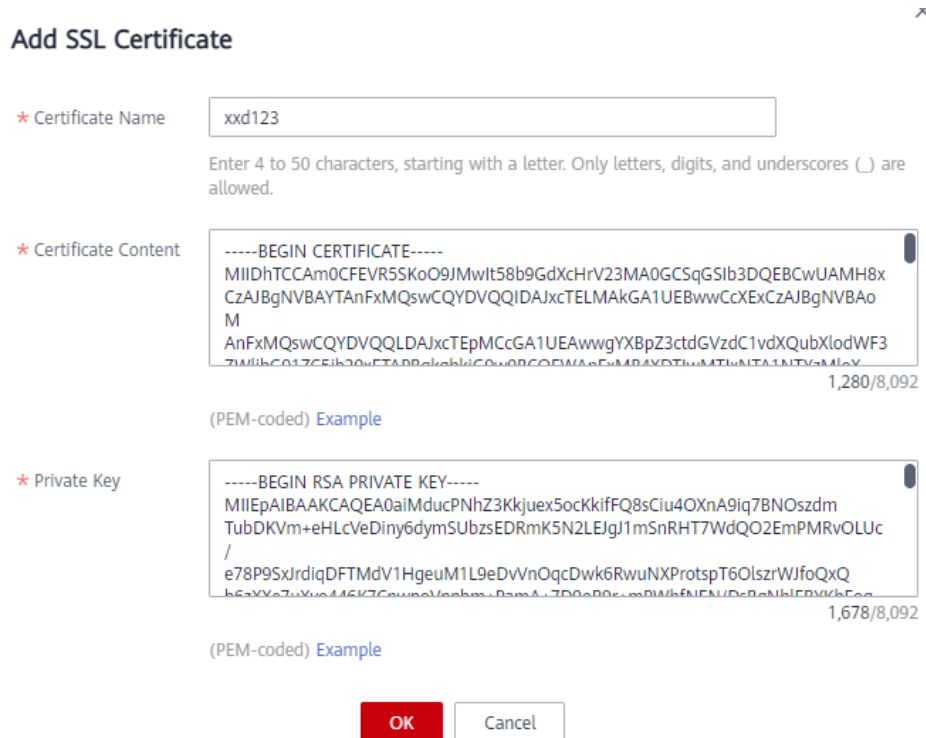
1. There is an independent domain name available.
2. Dedicated gateway: An A record points the independent domain name to the address of the gateway. For details, see section "Adding an A Record Set" in the *Domain Name Service User Guide*.
3. If the API group contains APIs that are called through HTTPS, there needs to be SSL certificates configured for the independent domain name. SSL certificates can only be added manually with a custom name, content, and a key.

### Procedure

**Step 1** Log in to the management console.

- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > API Groups**.
- Step 5** Go to the **Domain Names** tab page using one of the following methods:
- Click the name of the target API group, and click the **Domain Names** tab on the displayed API group details page.
  - In the **Operation** column of the target API group, choose **More > Manage Domain Name**.
- Step 6** Click **Bind Independent Domain Name** and enter a domain name.
- For API groups created under dedicated gateways, specify the minimum TLS version (TLS 1.1 or TLS 1.2) supported by domain names that you bind to the API groups. TLS 1.2 is recommended.
- Step 7** Click **OK**.
- If the domain name is not needed, click **Unbind** to unbind it from the API group.
- Step 8** (Optional) If the API group contains APIs that are accessed through HTTPS, add an SSL certificate.
- Click **Add SSL Certificate**.
  - Enter the name, content, and key of the **obtained SSL certificate**, and click **OK**.

**Figure 3-1** Adding an SSL certificate



**Add SSL Certificate**


\* Certificate Name   
Enter 4 to 50 characters, starting with a letter. Only letters, digits, and underscores ( \_ ) are allowed.

\* Certificate Content   
1,280/8,092  
(PEM-coded) Example

\* Private Key   
1,678/8,092  
(PEM-coded) Example

**OK**

 NOTE

- Currently, you can only add SSL certificates in the PEM format. To add SSL certificates of other formats, convert the certificates into the PEM format first.
- To replace or edit an SSL certificate, click  next to the certificate name. The certificate content and key will not be visible after you click **OK** to add the certificate. If the content has been updated, add the entire content or key again.
- If you do not require an SSL certificate, click **Delete SSL Certificate** in the row containing the certificate to delete it.

----End

## Troubleshooting

- Failure in binding an independent domain name: The independent domain name is not CNAMEd to the subdomain name of the API group, or the independent domain name already exists.
- Failure in adding an SSL certificate: The domain name of the SSL certificate is different from the domain name for which you add the SSL certificate.

## Follow-Up Operations

After binding independent domain names to the API group, create APIs in the group to selectively expose backend capabilities. For details, see [Creating an API](#).

# 3.3 Deleting an API Group

## Scenario

You can delete an API group if you do not require it.

 NOTE

API groups that contain APIs cannot be deleted.

## Prerequisites

You have created an API group.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > API Groups**.

**Step 5** Delete an API group. You can use one of the following methods:

- In the **Operation** column of the target API group, choose **More > Delete**.

- Click the name of the target API group, and click **Delete Group** in the upper right corner of the displayed API group details page.

**Step 6** Enter **DELETE** and click **Yes**.

----End

## 3.4 Adding a Gateway Response

### Scenario

A gateway response is displayed if API Gateway fails to process an API request. API Gateway provides a set of default responses and also allows you to create gateway responses with custom status codes and content, on the **API Groups** page. The response content must be in JSON format.

For example, the content of a default gateway response is as follows:

```
{"error_code": "$context.error.code", "error_msg": "$context.error.message", "request_id": "$context.requestId"}
```

You can add a response with the following content:

```
{"errorcode": "$context.error.code", "errormsg": "$context.error.message", "requestid": "$context.requestId", "apild": "$context.apild"}
```

You can add more fields to or delete existing fields from the JSON body.

#### NOTE


- The default gateway responses provided by API Gateway can be edited.
- You can create gateway responses and configure different responses for APIs in the same API group.
- The type of a gateway response cannot be changed. For details, see [Response Types](#).
- Gateway responses can contain the API gateway context variables (starting with **\$context**). For details, see [API Gateway Context Variables](#).

### Prerequisites

You have created an API group.

### Procedure

**Step 1** Log in to the management console.

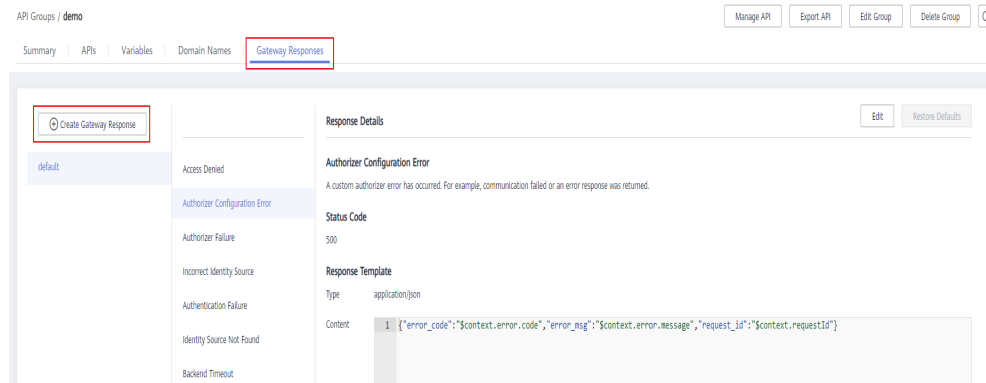
**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > API Groups**.

**Step 5** Locate the API group for which you want to create or modify a gateway response, and click the group name to go to the API group details page.

**Step 6** Click the **Gateway Responses** tab and create a gateway response.



**NOTE**

- To edit a response, click the **Edit** button in the upper right corner and modify the status code and content of the response.
- You can modify only the status code and content of a default or custom gateway response, and you cannot change the response type.
- Error information and other response details can be obtained using variables. For details about the supported variables, see [Table 3-3](#).

----End

## Response Types

[Table 3-2](#) lists the response types supported by API Gateway. You can define status codes of responses to meet your service requirements.

**Table 3-2** Error Response types supported by API Gateway

Response Name	Default Status Code	Description
Access Denied	403	Access denied. For example, the access control policy is triggered or an attack is detected.
Authorizer Configuration Error	500	A custom authorizer error has occurred. For example, communication failed or an error response was returned.
Authorizer Failed	500	The custom authorization failed.
Incorrect Identity Source	401	The identity source of the custom authorizer is missing or invalid.
Authentication Failure	401	IAM or app authentication failed.
Identity Source Not Found	401	No identity source has been specified.
Backend Timeout	504	Communication with the backend service timed out.

Response Name	Default Status Code	Description
Backend Unavailable	502	The backend service is unavailable due to communication error.
Default 4XX	-	Another 4XX error occurred.
Default 5XX	-	Another 5XX error occurred.
No API Found	404	No API is found.
Incorrect Request Parameters	400	The request parameters are incorrect or the HTTP method is not supported.
Request Throttled	429	The request was rejected due to request throttling.
Unauthorized App	401	The app you are using has not been authorized to call the API.

## API Gateway Context Variables

**Table 3-3** Variables that can be used in response message body

Variable	Description
\$context.apid	API ID.
\$context.appld	ID of the app that calls the API.
\$context.requestId	Request ID generated when the API is called.
\$context.stage	Deployment environment in which the API is called.
\$context.sourceIp	Source IP address of the API caller.
\$context.authorizer.frontend.property	Values of the specified attribute-value pairs mapped to the context in the frontend custom authorizer response
\$context.authorizer.backend.property	Values of the specified attribute-value pairs mapped to the context in the backend custom authorizer response
\$context.error.message	Error message.
\$context.error.code	Error code.
\$context.error.type	Error type.

# 4 API Management

---

## 4.1 Creating an API

### Scenario

You can selectively expose your services by configuring their APIs in API Gateway.

To create an API, set the basic information and define the API request, backend service, and responses.

#### NOTE

API Gateway uses a REST-based API architecture, so API opening and calling must comply with related RESTful API specifications.

### Prerequisites

- You have created an API group. If no API group is available, create one during API creation.
- If the backend service of the API is deployed in a VPC, you have created a VPC channel to access the service by following the procedure in [Creating a VPC Channel](#). You can also create a VPC channel during API creation.

### Setting Basic Information

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Click **Create API**, and set the parameters listed in [Table 4-1](#).



**Table 4-1** Basic information

Parameter	Description
Name	API name. It is recommended that you enter a name based on naming rules to facilitate search.
API Group	The group to which the API belongs. If no API group is available, click <b>Create API Group</b> to create one.
Gateway Response	Displayed if API Gateway fails to process an API request. API Gateway provides a set of default responses and also allows you to <a href="#">create gateway responses</a> with custom status codes and content, on the <b>API Groups</b> page. The response content must be in JSON format.
Visibility	Determine whether the API is available to the public. Options: <ul style="list-style-type: none"> <li>• <b>Public</b></li> </ul>
Security Authentication	<p>The following authentication modes are available:</p> <ul style="list-style-type: none"> <li>• <b>App</b>: Requests for the API will be authenticated by API Gateway.</li> <li>• <b>IAM</b>: Requests for the API will be authenticated by Identity and Access Management (IAM).</li> <li>• <b>Custom</b>: Requests for the API will be authenticated by using your own authentication system or service (for example, an OAuth-based authentication system).</li> <li>• <b>None</b>: No authentication will be required.</li> </ul> <p>The API calling method varies depending on the authentication mode. For details, see <i>Developer Guide</i>.</p> <p>App authentication is recommended.</p> <p><b>NOTICE</b></p> <ul style="list-style-type: none"> <li>• If you set the authentication mode of an API to <b>IAM</b>, any API Gateway user can access the API, which can result in excessive charges if the API is bombarded with malicious requests.</li> <li>• If you set the authentication mode of an API to <b>None</b>, any user can access the API over public networks, which can result in excessive charges if the API is bombarded with malicious requests.</li> <li>• If you set the authentication mode of an API to <b>Custom</b>, you can create a function in FunctionGraph to interconnect with your own authentication system or service. This authentication mode is not supported in regions where FunctionGraph is unavailable.</li> </ul>

Parameter	Description
Simple Authentication	<p>This parameter is available only if you set <b>Security Authentication</b> to <b>App</b>.</p> <p>If you select app authentication, you can configure whether to enable simple authentication. In simple authentication, the <b>X-Api-AppCode</b> parameter is added to the HTTP request header for quick response. API Gateway verifies only the AppCode and the request content does not need to be signed.</p> <p>Simple authentication only supports HTTPS requests and does not support HTTP requests. For details, see <a href="#">Adding an AppCode for Simple Authentication</a>.</p> <p><b>NOTE</b> After you enable simple authentication for an existing API, you need to publish the API again. For details, see <a href="#">Publishing an API</a>.</p>
Custom Authorizer	<p>This parameter is mandatory if <b>Security Authentication</b> is set to <b>Custom</b>.</p> <p>Select a custom authorizer if you set <b>Security Authentication</b> to <b>Custom</b>. If no custom authorizer is available, click <b>Create Custom Authorizer</b> to create one.</p>
Tag Name	Classification attribute used to quickly identify the API from other APIs.
Description	Description of the API.

**Step 6** Click **Next**.

----End

## Defining API Request

**Step 1** On the **Define API Request** page, set the parameters listed in [Table 4-2](#).

**Figure 4-1** Define API Request

The screenshot shows the 'Define API Request' configuration interface. It includes the following elements:

- Domain Name:** fcf0213b01d54adf857fe0571c20dbd5.apigw-ae-ad-1-g42cloud.com
- Protocol:** Three radio buttons for HTTP, HTTPS, and HTTP&HTTPS. The HTTPS option is selected.
- Path:** A text input field containing the example path /getUserInfo/{userid}. A note below states: 'Enclose parameters in braces, for example, /a/{b}. You can also use a plus sign (+) to match parameters starting with specific characters, for example, /a/{b+}'.
- Matching:** Two radio buttons for Exact match and Prefix match. The Exact match option is selected. A note below states: 'API requests will be forwarded to the specified path.'
- Method:** A dropdown menu with 'GET' selected.
- CORS:** A toggle switch that is currently turned off. A note below states: 'Enable cross-origin resource sharing (CORS) if you want to allow restricted resources on a web page to be requested from other domains. Learn more about CORS.'

**Table 4-2** Parameters for defining API requests

Parameter	Description
Domain Name	The subdomain automatically allocated to the API group.
Protocol	<p>The protocol used for calling the API. Options:</p> <ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> <li>• HTTP&amp;HTTPS</li> </ul> <p>HTTPS is recommended for transmitting important or sensitive data.</p>
Path	<p>The path for requesting the API.</p> <p>Enter a path in the format of <code>"/users/{userId}/projects"</code>.</p> <ul style="list-style-type: none"> <li>• The variable in braces (<code>{}</code>) is a request parameter. Ensure that it is an entire segment between a pair of slashes (<code>/</code>). A segment that is not marked by a pair of slashes, for example, <code>/abc{userId}</code>, is not supported. If you set the matching mode to <b>Exact match</b>, you can add a plus sign (+) to the end of the request parameter, for example, <code>/users/{p+}</code>. The variable <i>p</i> matches the segments between one or multiple pairs of slashes (<code>/</code>).</li> <li>• Ensure that you define the parameters contained in the request path as input parameters.</li> <li>• The content is case-sensitive.</li> </ul>
Matching	<p>Options:</p> <ul style="list-style-type: none"> <li>• <b>Exact match</b>: The API can be called only using the specified request path.</li> <li>• <b>Prefix match</b>: The API can be called using paths starting with the matching characters. For example, if you set the request path to <code>/test/AA</code> and the matching mode to <b>Prefix match</b>, the API can be called using <code>/test/AA/CC</code> but cannot be called using <code>/test/AACC</code>.</li> </ul> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• Exact match takes precedence over prefix match. Prefix match with a short prefix has a lower priority. For example, for request path <code>/a/b/c</code> (exact match), <code>/a</code> (prefix match), and <code>/a/b</code> (prefix match), the matching order is <code>/a/b/c &gt; /a/b &gt; /a</code>.</li> <li>• If you set the matching mode to <b>Prefix match</b>, the characters of the API request path excluding the prefix are transparently transmitted to the backend service. For example, if you define the frontend and backend request paths of an API as <code>/test/</code> and <code>/test2/</code>, respectively, and the API is called using <code>/test/AA/CC</code>, the characters <code>AA/CC</code> will be transparently transmitted to the backend service. The request URL received by the backend service is <code>/test2/AA/CC/</code>.</li> </ul>

Parameter	Description
Method	<p>The API calling method. The options are <b>GET</b>, <b>POST</b>, <b>DELETE</b>, <b>PUT</b>, <b>PATCH</b>, <b>HEAD</b>, <b>OPTIONS</b>, and <b>ANY</b>.</p> <ul style="list-style-type: none"> <li>• <b>ANY</b> indicates that the API can be called using any request method.</li> <li>• If you set <b>Method</b> to <b>POST</b>, <b>PUT</b>, <b>PATCH</b>, or <b>ANY</b>, set the request body.</li> </ul>
CORS	<p>Determine whether to enable cross-origin resource sharing (CORS).</p> <p>CORS allows browsers to send XMLHttpRequest to servers in other domains, overcoming the limitation that Asynchronous JavaScript and XML (AJAX) can be used only within the same domain.</p> <p>There are two types of CORS requests:</p> <ul style="list-style-type: none"> <li>• Simple requests: requests that have the <b>Origin</b> field in the header.</li> <li>• Not-so-simple requests: HTTP requests sent before the actual request.</li> </ul> <p>If you enable CORS, you need to create another API that uses the <b>OPTIONS</b> method. For details, see <a href="#">CORS</a>.</p>

**Step 2** (Optional) Set input parameters.

Input parameters are transmitted together with the request when the API is called.

1. Click **Add Input Parameter**.
2. Set the parameters listed in [Table 4-3](#).

**Table 4-3** Input parameter definition

Parameter	Description
Name	<p>Name of the input parameter. If you set the parameter location to <b>PATH</b>, ensure that the parameter name is the same as that defined in the request path.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>- The parameter name is not case-sensitive. It cannot start with <b>x-apig-</b> or <b>x-sdk-</b>.</li> <li>- The parameter name cannot be <b>x-stage</b>.</li> <li>- If you set the parameter location to <b>HEADER</b>, ensure that the parameter name is not <b>Authorization</b> or <b>X-Auth-Token</b> and does not contain underscores (_).</li> </ul>
Location	<p>Position of the parameter in requests. The options are <b>PATH</b>, <b>HEADER</b>, and <b>QUERY</b>.</p> <p><b>NOTE</b></p> <p>If you set the parameter location to <b>PATH</b>, you must include the parameter in the request path.</p>

Parameter	Description
Type	Type of the parameter value. Options: <b>STRING</b> and <b>NUMBER</b> . <b>NOTE</b> Set the type of Boolean parameters to <b>STRING</b> .
Mandatory	Determine whether the input parameter is required in each request sent to call the API. If you select <b>Yes</b> , API requests that do not contain the input parameter will be rejected.
Passthrough	Determine whether to transparently transmit the input parameter to the backend service.
Default Value	The value that will be used if no value is specified for the input parameter when the API is called. If the input parameter is not specified in a request, API Gateway will automatically send the default value to the backend service.
Enumerated Value	Enumerated value of the input parameter. Use commas (,) to separate multiple enumerated values. The value of this input parameter can only be one of the enumerated values.
Minimum Length	The minimum length of the parameter value. Only numbers are allowed.
Maximum Length	The maximum length of the parameter value. Only numbers are allowed.
Example	Example value for the parameter.
Description	Description of the parameter.

3. Click **OK**.

**Step 3** Click **Next**.

----End

## Defining Backend Service

API Gateway allows you to define multiple backend policies for different scenarios. Requests that meet specified conditions will be forwarded to the corresponding backend. For example, you can have certain requests to an API forwarded to a specific backend by specifying the source IP address in the policy conditions of the backend.

You can define a maximum of five backend policies for an API in addition to the default backend.

**Step 1** Define the default backend.

API requests that do not meet the conditions of any backend will be forwarded to the default backend.

On the **Define Backend Request** page, select a backend type.

**Table 4-4** and **Table 4-5** describe the backend service parameters.

**Table 4-4** Parameters for defining an HTTP/HTTPS backend service

Parameter	Description
Protocol	<p>HTTP or HTTPS. This protocol must be the one used by the backend service.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• WebSocket is supported for HTTP and HTTPS.</li> <li>• HTTPS is recommended for transmitting important or sensitive data.</li> </ul>
Method	<p>The API calling method. The options are <b>GET, POST, DELETE, PUT, PATCH, HEAD, OPTIONS, and ANY</b>.</p> <p><b>ANY</b> indicates that the API can be called using any request method.</p>
VPC Channel	<p>Determine whether the backend service will be accessed using a VPC channel.</p> <ul style="list-style-type: none"> <li>• <b>If yes, select a VPC channel.</b></li> </ul> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>- To ensure a successful health check and service availability, configure the security groups of cloud servers in each VPC channel to allow access from 100.125.0.0/16.</li> </ul> <ul style="list-style-type: none"> <li>• <b>If no, configure the backend service address.</b> Enter a backend address in the format of "host IP address or domain name":"port number". The default port (80 for HTTP and 443 for HTTPS) will be used if you do not specify a port. Available ports: 1 to 65535. If you want to use a variable, enclose the variable name in number signs (#), for example, <b>#ipaddress#</b>. You can use multiple variables, for example, <b>#ipaddress##test#</b>.</li> </ul>
Host Header (if applicable)	<p>This parameter is available only if you set <b>VPC Channel to Configure</b>.</p> <p>Define a host header for requests to be sent to cloud servers associated with the VPC channel. By default, the original host header in each request will be used.</p>
Path	<p>The request path (URI) of the backend service. Ensure that any parameters in the path are enclosed in braces ({}). For example, <b>/getUserInfo/{userId}</b>.</p> <p>If the path contains an environment variable, enclose the environment variable in number signs (#), for example, <b>/#path#</b>. You can use multiple environment variables, for example, <b>/#path##request#</b>.</p>

Parameter	Description
Timeout (ms)	<p>Backend request timeout.</p> <p>If a backend timeout error occurs during API debugging, increase the timeout to locate the reason.</p> <p><b>NOTE</b> For dedicated gateways, you can modify the maximum timeout by referring to <a href="#">Configuration Parameters</a>. The value range is 1 ms to 600,000 ms.</p>
Two-way Authentication	<p>Determine whether to allow API Gateway to authenticate the API backend service through HTTPS. For details about how to configure the certificate for two-way authentication, see <a href="#">Configuration Parameters</a>.</p>
Backend Authentication	<p>Determine whether your backend service needs to authenticate API requests.</p> <p>If you enable this option, select a custom authorizer for backend authentication. <a href="#">Custom authorizers</a> are functions that are created in FunctionGraph to implement an authentication logic or to invoke an authentication service.</p> <p><b>NOTE</b> Backend authentication relies on FunctionGraph and is only available in certain regions.</p>

**Table 4-5** Parameters for defining a Mock backend service

Parameter	Description
Response	<p>You can use Mock for API development, debugging, and verification. It enables API Gateway to return a response without sending the request to the backend. This is useful if you need to test APIs when the backend is unavailable.</p>
Backend Authentication	<p>For details, see the description about backend authentication in <a href="#">Table 4-4</a>.</p>
Header Parameters	<p>API response headers.</p> <p>Click <b>Add Header</b>, and enter the parameter name, value, and description.</p>

 **NOTE**

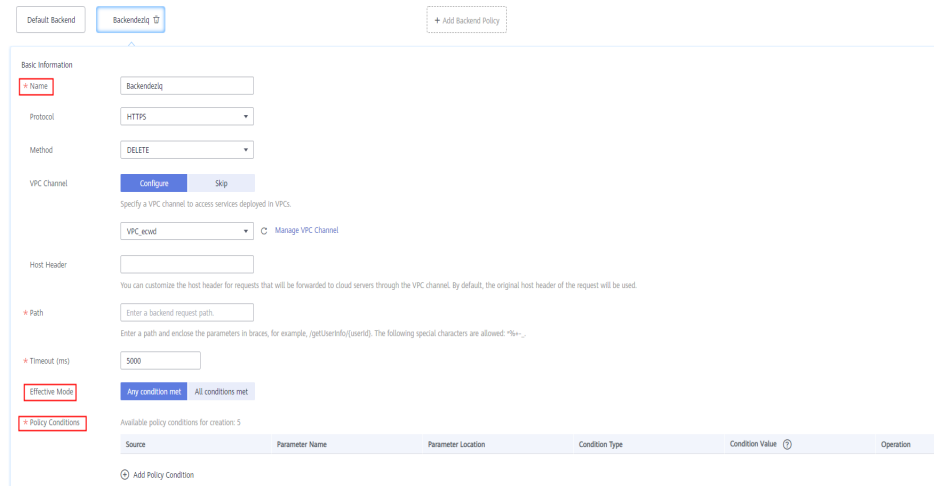
- If you have defined an environment variable in the backend request path, the API cannot be debugged on the API debugging page.
- For variables defined in the backend request path of an API, corresponding environment variables and their values must be configured. Otherwise, the API cannot be published because there will be no values that can be assigned to the variables.
- Environment variable names are case-sensitive.

**Step 2** (Optional) Add a backend policy.

You can add backend policies to forward requests to different backend services.

1. Click **Add Backend Policy**.
2. Set parameters by referring to [Table 4-6](#) and [Table 4-4](#).

**Figure 4-2** Adding a backend policy



**Table 4-6** Backend policy parameters

Parameter	Description
Name	The backend policy name.
Effective Mode	<ul style="list-style-type: none"> <li>– <b>Any condition met:</b> The backend policy takes effect if any of the policy conditions has been met.</li> <li>– <b>All conditions met:</b> The backend policy takes effect only when all the policy conditions have been met.</li> </ul>
Policy Conditions	Conditions that must be met for the backend policy to take effect. Set conditions by referring to <a href="#">Table 4-7</a> .

**Table 4-7** Policy conditions


Parameter	Description
Source	<ul style="list-style-type: none"> <li>– Source IP address</li> <li>– Input parameter</li> </ul> <p><b>NOTICE</b> Input parameters (for example, headers) set as policy conditions must have already been defined in the API request settings.</p>
Parameter Name	– When setting <b>Source</b> to <b>Input parameter</b> , select an input parameter.



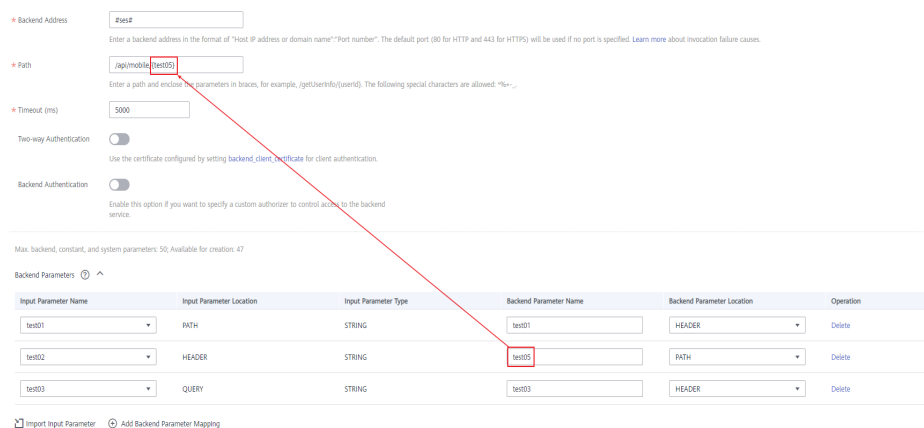
Parameter	Description
Parameter Location	The parameter location is displayed only if you set <b>Source</b> to <b>Input parameter</b> .
Condition Type	This parameter is required only if you set <b>Source</b> to <b>Input parameter</b> . <ul style="list-style-type: none"> <li>- <b>Equal:</b> The request parameter must be equal to the specified value.</li> <li>- <b>Enumerated:</b> The request parameter must be equal to any of the enumerated values.</li> <li>- <b>Matching:</b> The request parameter must be equal to any value of the regular expression.</li> </ul>
Condition Value	Set a condition value according to the condition type. <ul style="list-style-type: none"> <li>- <b>Equal:</b> Enter a value.</li> <li>- <b>Enumerated:</b> Enter multiple values and separate them using commas.</li> <li>- <b>Matching:</b> Enter a range, for example, <b>[0-5]</b>.</li> </ul> If you have set <b>Source</b> to <b>Source IP address</b> , enter one or more IP addresses and separate them using commas.

**Step 3** (Optional) Set backend parameters.

Input parameters of the API are mapped to corresponding backend parameters in backend requests.

1. Click  next to **Backend Parameters**, and define backend parameters. You can use one of the following methods:
  - Click **Import Input Parameter**. All the defined input parameters are automatically displayed.
  - Click **Add Backend Parameter Mapping**, and add required backend parameters.
2. Modify the mappings based on the parameters and their locations in backend requests. **Figure 4-3** highlights the backend parameters.

**Figure 4-3** Configuring backend parameters



- a. If you set the parameter location to **PATH**, ensure that the parameter name is the same as that defined in the backend request path.
- b. The name and location of an input parameter can be different from those of the mapped backend request parameter.

 **NOTE**

- The parameter name is not case-sensitive. It cannot start with **x-apig-** or **x-sdk-**.
  - The parameter name cannot be **x-stage**.
  - If you set the parameter location to **HEADER**, ensure that the parameter name does not contain underscores (**\_**).
- c. In the preceding figure, parameters **test01** and **test03** are located in the path and query positions of API requests, and their values will be received in the header of backend requests. **test02** is located in the header of API requests, and its value will be received through **test05** in the path of backend requests.

For example, **test01** is **abc**, **test02** is **def**, and **test03** is **xyz**.

API request:


```
curl -ik -H 'test02:def' -X GET https://www.example01.com/v1.0/abc?test03=xyz
```

Backend request:

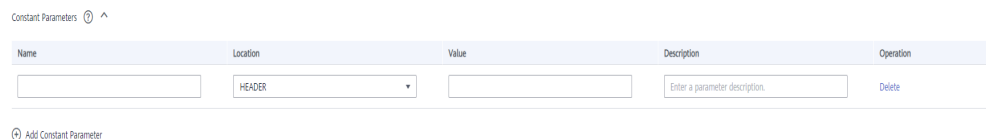
```
curl -ik -H 'test01:abc' -H 'test03:xyz' -X GET https://www.example02.com/v1.0/def
```

**Step 4** (Optional) Set constant parameters.

You can define constant parameters for the backend service to receive constants that are invisible to API callers. API Gateway adds constant parameters to specified positions in the request sent to the backend service.

1. Click  next to **Constant Parameters**.
2. Click **Add Constant Parameter**, and set the parameters listed in [Table 4-8](#).

**Figure 4-4** Adding constant parameters



**Table 4-8** Setting constant parameters


Parameter	Description
Name	<p>Constant parameter name. If you set the parameter location to <b>PATH</b>, ensure that the parameter name is the same as that defined in the backend request path.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>- The parameter name is not case-sensitive. It cannot start with <b>x-apig-</b> or <b>x-sdk-</b>.</li> <li>- The parameter name cannot be <b>x-stage</b>.</li> <li>- If you set the parameter location to <b>HEADER</b>, ensure that the parameter name does not contain underscores (_).</li> </ul>
Location	<p>Position of the parameter in requests.</p> <p>The options are <b>PATH</b>, <b>QUERY</b>, and <b>HEADER</b>.</p>
Value	Value of the parameter.
Description	Description of the constant parameter.

 **NOTE**

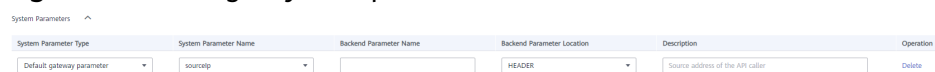
- API Gateway sends requests containing constant parameters to backend services after percent-encoding of special parameter values. Ensure that the backend services support percent-encoding. For example, parameter value **[apig]** becomes **%5Bapig%5D** after percent-encoding.
- For values of path parameters, the following characters will be percent-encoded: ASCII codes 0–31, blank symbols, ASCII codes 127–255, and special characters ?></%#"[\]^`{}
- For values of query strings, the following characters will be percent-encoded: ASCII codes 0–31, blank symbols, ASCII codes 127–255, and special characters >=<+&%#"[\]^`{}

**Step 5** (Optional) Set system parameters.

System parameters refer to runtime parameters regarding gateway running and frontend and backend authentications. The parameters are transferred to the API backend service for access control and custom authentication.

1. Click  next to **System Parameters**.
2. Click **Add System Parameter**, and set the parameters listed in [Table 4-9](#).

**Figure 4-5** Adding a system parameter



System Parameter Type	System Parameter Name	Backend Parameter Name	Backend Parameter Location	Description	Operation
Default gateway parameter	sourcecp		HEADER	Source address of the API caller	Delete

**Table 4-9** System parameters

Parameter	Description
System Parameter Type	<ul style="list-style-type: none"> <li>- <b>Default gateway parameter:</b> Default parameters supported by API Gateway.</li> <li>- <b>Frontend authentication parameter:</b> Parameters to be displayed in the frontend custom authentication result. This option is available only if you select <b>Custom</b> for <b>Security Authentication</b> on the <b>Set Basic Information</b> page.</li> <li>- <b>Backend authentication parameter:</b> Parameters to be displayed in the backend custom authentication result. This option is available only if you enable <b>Backend Authentication</b> on the <b>Define Backend Request</b> page.</li> </ul>
System Parameter Name	<ul style="list-style-type: none"> <li>- If <b>System Parameter Type</b> is <b>Default gateway parameter</b>, select any of the following parameters. <ul style="list-style-type: none"> <li>▪ <b>sourceIp:</b> source IP address of the API caller</li> <li>▪ <b>stage:</b> environment in which the API is called</li> <li>▪ <b>apId:</b> ID of the API</li> <li>▪ <b>appId:</b> ID of the app that calls the API</li> <li>▪ <b>requestId:</b> request ID generated when the API is called</li> <li>▪ <b>serverAddr:</b> IP address of the gateway server</li> <li>▪ <b>serverName:</b> name of the gateway server</li> <li>▪ <b>handleTime:</b> processing time of the API request</li> <li>▪ <b>providerAppId:</b> app ID of the API provider</li> </ul> </li> <li>- Ensure that the frontend and backend authentication parameters are consistent with the return result parameters defined for the corresponding custom authorizer function. For details about how to create a custom authorizer function and obtain returned result parameters, see <i>API Gateway Developer Guide</i>.</li> </ul>
Backend Parameter Name	<p>Name of the backend parameter to which the system parameter will be mapped.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>- The parameter name is not case-sensitive. It cannot start with <b>x-apig-</b> or <b>x-sdk-</b>.</li> <li>- The parameter name cannot be <b>x-stage</b>.</li> <li>- If you set the parameter location to <b>HEADER</b>, ensure that the parameter name does not contain underscores (_).</li> </ul>

Parameter	Description
Backend Parameter Location	Position of the backend parameter in requests.
Description	Description of the system parameter.

**Step 6** Click **Next**.

----End

## Defining Responses

**Step 1** On the **Define Response** page, set the parameters listed in [Table 4-10](#).

**Table 4-10** Defining responses

Parameter	Description
Example Success Response	An example of a response returned when the API is called successfully.
Example Failure Response	An example of a response returned when the API fails to be called.

**Step 2** Click **Finish**.

After the API is created, click its name in the API list to view details.

----End

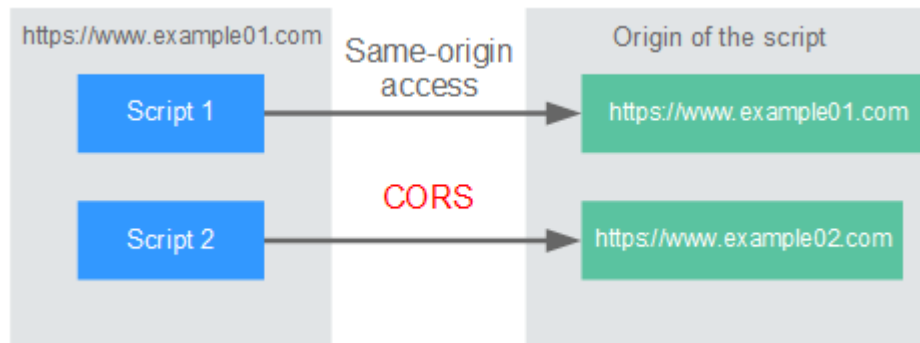
## Follow-Up Operations

After creating an API, verify it by following the procedure in [Debugging an API](#).

## 4.2 CORS

### What Is CORS?

For security reasons, browsers restrict cross-origin requests initiated from within scripts. This means that a web application can only request resources from its origin. The CORS mechanism allows browsers to send XMLHttpRequest to servers in other domains and request access to the resources there.

**Figure 4-6** Process flow of the CORS mechanism

There are two types of CORS requests:

- **Simple requests**

Simple requests must meet the following conditions:

- a. The request method is HEAD, GET, or POST.
- b. The request header contains only the following fields:
  - Accept
  - Accept-Language
  - Content-Language
  - Last-Event-ID
  - Content-Type (**application/x-www-form-urlencoded, multipart/form-data, or text/plain**)

In the header of a simple request, browsers automatically add the **Origin** field to specify the origin (including the protocol, domain, and port) of the request. After receiving such a request, the target server determines whether the request is safe and can be accepted based on the origin. If the server sends a response containing the **Access-Control-Allow-Origin** field, the server accepts the request.

- **Not-so-simple requests**

Requests that do not meet the conditions for simple requests are not-so-simple requests.

Before sending a not-so-simple request, browsers send an HTTP preflight request to the target server to confirm whether the origin the web page is loaded from is in the allowed origin list, and to confirm which HTTP request methods and header fields can be used. If the preflight request is successful, browsers send simple requests to the server.

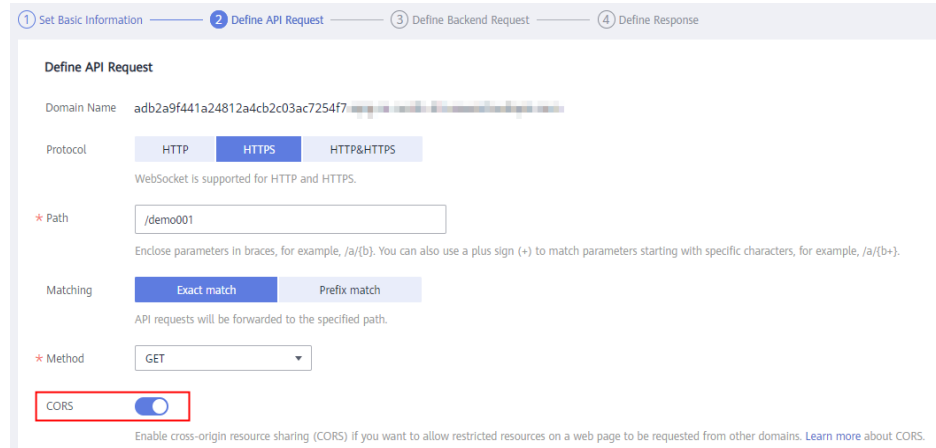
## Configuring CORS

CORS is disabled by default. To enable CORS for an API, perform the operations described in this section. To customize request headers, request methods, and origins allowed for cross-domain access, create a [CORS plug-in](#).

- **Simple CORS requests**

When creating an API, enable CORS on the API request configuration page. For more information, see [Simple Request](#).

**Figure 4-7 CORS**



- **Not-so-simple CORS requests**

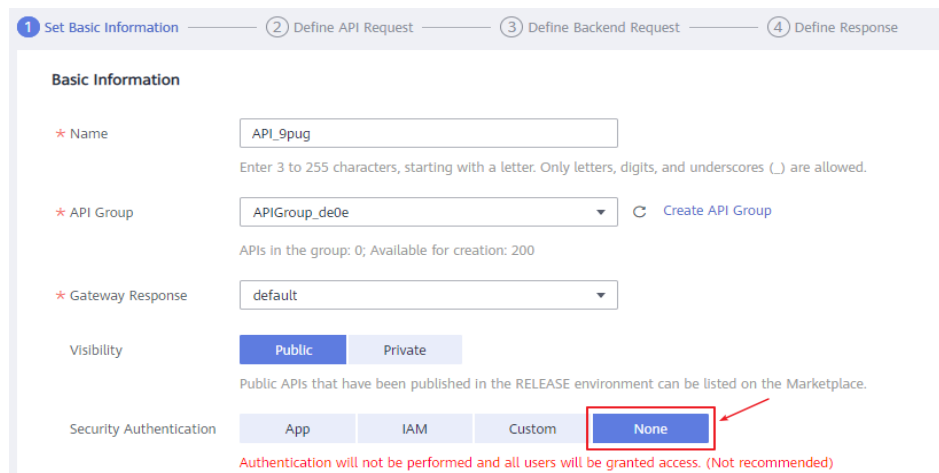
**NOTICE**

If your API will receive not-so-simple requests, **create another API that will be accessed using the OPTIONS method** in the same group as the target API to receive preflight requests.

Follow this procedure to define the preflight request API. For more information, see [Not-So-Simple Request](#).

- On the **Set Basic Information** page, select **None** to skip over security authentication.

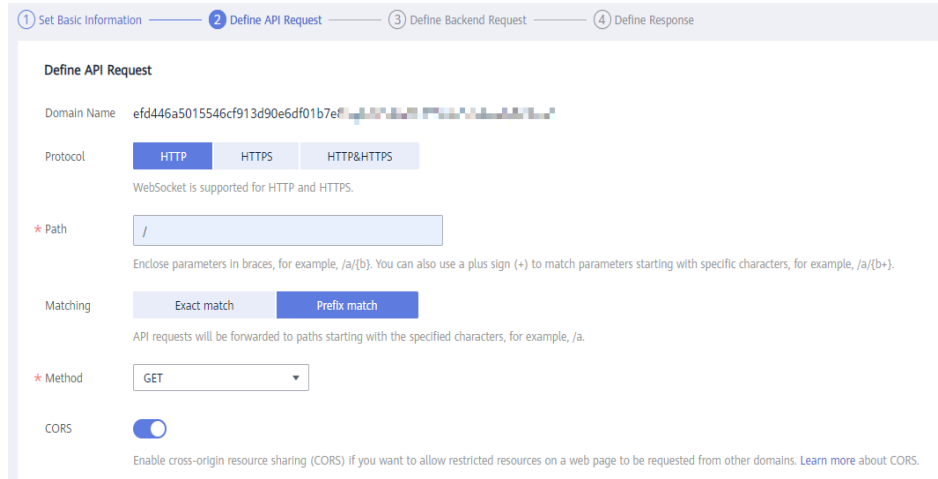
**Figure 4-8 None authentication**



- On the **Define API Request** page, perform the following settings:
  - **Protocol:** The same protocol used by the API with CORS enabled.

- **Path:** Enter a slash (/).
- **Method:** Select **OPTIONS**.
- **CORS:** Enabled.

**Figure 4-9** Defining the API request



- c. Select the **Mock** backend type.

## Simple Request

When creating an API that will receive simple requests, **enable CORS** for the API.

**Scenario 1:** If CORS is enabled and the response from the backend does not contain a CORS header, API Gateway handles requests from any domain, and returns the **Access-Control-Allow-Origin** header. For example:

### Request sent by a browser and containing the Origin header field:

```
GET /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

**Origin:** This field is required to specify the origin (**http://www.cors.com** in this example) of the request. API Gateway and the backend service determine based on the origin whether the request is safe and can be accepted.

### Response sent by the backend service:

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: api-gateway

{"status": "200"}
```

### Response sent by API Gateway:

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
```



```
Content-Type: application/json
Content-Length: 16
Server: api-gateway
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: *

{"status": "200"}
```

**Access-Control-Allow-Origin:** This field is required. The asterisk (\*) means that API Gateway handles requests sent from any domain.

**Scenario 2:** If CORS is enabled and the response from the backend contains a CORS header, the header will overwrite that added by API Gateway. The following messages are used as examples:

#### Request sent by a browser and containing the Origin header field:

```
GET /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

**Origin:** This field is required to specify the origin (**http://www.cors.com** in this example) of the request. API Gateway and the backend service determine based on the origin whether the request is safe and can be accepted.

#### Response sent by the backend service:

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: api-gateway
Access-Control-Allow-Origin: http://www.cors.com

{"status": "200"}
```

**Access-Control-Allow-Origin:** Indicates that the backend service accepts requests sent from **http://www.cors.com**.

#### Response sent by API Gateway:

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: api-gateway
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: http://www.cors.com

{"status": "200"}
```

The CORS header in the backend response overwrites that in API Gateway's response.

## Not-So-Simple Request

When creating an API that will receive not-so-simple requests, enable CORS for the API by following the instructions in [Configuring CORS](#), and create another API that will be accessed using the OPTIONS method.

The request parameters of an API accessed using the OPTIONS method must be set as follows:

- **API Group:** The same group to which the API with CORS enabled belongs.
- **Security Authentication:** Select **None**. No authentication is required for requests received by the new API no matter which security authentication mode has been selected.
- **Protocol:** The same protocol used by the API with CORS enabled.
- **Path:** Enter a slash (/) or select the path that has been set for or matches the API with CORS enabled.
- **Method:** Select **OPTIONS**.
- **CORS:** Enabled.

The following are example requests and responses sent to or from a mock backend.

#### Request sent from a browser to an API that is accessed using the OPTIONS method:

```
OPTIONS /HTTP/1.1
User-Agent: curl/7.29.0
Host: localhost
Accept: */*
Origin: http://www.cors.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-Sdk-Date
```

- **Origin:** This field is required to specify the origin from which the request has been sent.
- **Access-Control-Request-Method:** This field is required to specify the HTTP methods to be used by the subsequent simple requests.
- **Access-Control-Request-Headers:** This field is optional and used to specify the additional header fields in the subsequent simple requests.

**Response sent by the backend:** none

#### Response sent by API Gateway:

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 02:38:48 GMT
Content-Type: application/json
Content-Length: 1036
Server: api-gateway
X-Request-Id: c9b8926888c356d6a9581c5c10bb4d11
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Stage,X-Sdk-Date,X-Sdk-Nonce,X-Proxy-Signed-Headers,X-Sdk-Content-Sha256,X-Forwarded-For,Authorization,Content-Type,Accept,Accept-Ranges,Cache-Control,Range
Access-Control-Expose-Headers: X-Request-Id,X-Apig-Latency,X-Apig-Upstream-Latency,X-Apig-RateLimit-Api,X-Apig-RateLimit-User,X-Apig-RateLimit-App,X-Apig-RateLimit-Ip,X-Apig-RateLimit-Api-Allenv
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,HEAD,OPTIONS,PATCH
Access-Control-Max-Age: 172800
```

- **Access-Control-Allow-Origin:** This field is required. The asterisk (\*) means that API Gateway handles requests sent from any domain.
- **Access-Control-Allow-Headers:** This field is required if it is contained in the request. It indicates all header fields that can be used during cross-origin access.
- **Access-Control-Expose-Headers:** This is the response header fields that can be viewed during cross-region access.
- **Access-Control-Allow-Methods:** This field is required to specify which HTTP request methods the API Gateway supports.

- **Access-Control-Max-Age:** This field is optional and used to specify the length of time (in seconds) during which the preflight result remains valid. No more preflight requests will be sent within the specified period.

**Request sent by a browser and containing the Origin header field:**

```
PUT /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

**Response sent by the backend:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: api-gateway

{"status":"200"}
```

**Response sent by API Gateway:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: api-gateway
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: *

{"status":"200"}
```

## 4.3 Debugging an API

### Scenario

After creating an API, debug it on the API Gateway console by setting HTTP headers and body parameters to verify whether the API is running normally.

**NOTE**

- APIs with backend request paths containing variables cannot be debugged.
- If an API has been bound with a request throttling policy, the policy will not work during debugging of the API.

### Prerequisites

- You have created an API group and API.
- You have set up the backend service of the API.

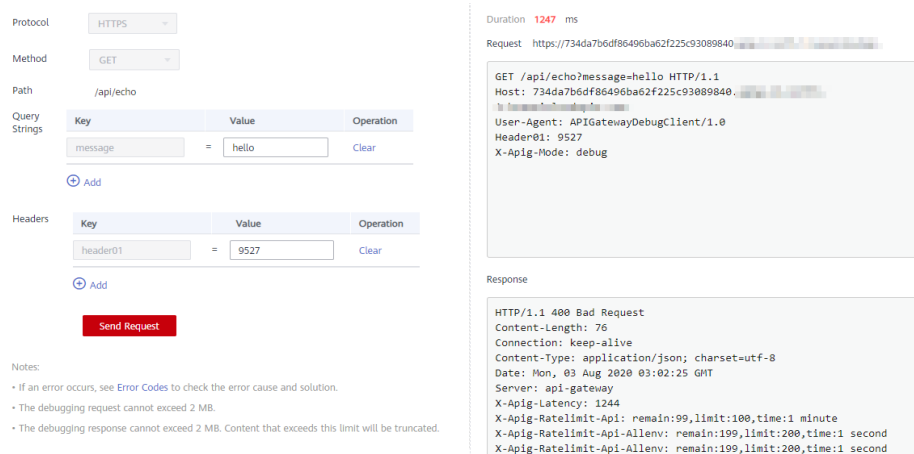
### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > APIs**.
- Step 5** Debug an API. You can use one of the following methods:
- In the **Operation** column of the API you want to debug, choose **More > Debug**.
  - Click the name of the target API, and click **Debug** in the upper right corner of the displayed API details page.

**Figure 4-10** Debugging an API



On the left side, set the API request parameters listed in [Table 4-11](#). On the right side, view the API request and response information after you click **Send Request**.

**Table 4-11** Parameters for debugging an API

Parameter	Description
Protocol	This parameter can be modified only if you set <b>Protocol</b> to <b>HTTP&amp;HTTPS</b> for the API.
Method	This parameter can be modified only if you set <b>Method</b> to <b>ANY</b> for the API.
Suffix	You can define a path only if you have set <b>Matching</b> to <b>Prefix match</b> for the API.
Path	Request path of the API.
Path Parameters	This parameter can be modified only if you have defined path parameters (such as <b>{test}</b> ) for the API.
Headers	HTTP headers and values.
Query Strings	Query string parameters and values.
Body	This parameter can be modified only if you set <b>Method</b> to <b>PATCH, POST, or PUT</b> for the API.

 NOTE

The fields displayed on the debugging page vary according to the request type.

**Step 6** After setting request parameters, click **Send Request**.

The box on the lower right displays the response of the API request.

- If the debugging is successful, the HTTP status code **200** and response details are displayed.
- If the request fails to be sent, an HTTP status code **4xx** or **5xx** is displayed. For details, see [Error Codes](#).

**Step 7** You can send more requests with different parameters and values to verify the API.

 NOTE

To modify the API configurations, click **Edit** in the upper right corner, and modify the parameters on the **Edit API** page.

----End

## Follow-Up Operations

After the API is successfully debugged, [publish](#) the API in a specific environment so that the API can be called by users. To ensure security of the API, create request throttling policies (see [Creating a Request Throttling Policy](#)), access control policies ([Creating an Access Control Policy](#)), and signature keys ([Creating and Using a Signature Key](#)) for the API.

## 4.4 Authorizing Apps to Call an API

### Scenario

APIs using app authentication can only be called by apps that have been authorized to call them.

 NOTE

- You can only authorize apps to call published APIs.
- You can authorize apps only to call APIs that use app authentication.

### Prerequisites

- You have created an API group and API.
- (Optional) You have created an environment.
- You have created an app.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Authorize apps to call an API. You can use one of the following methods:

- In the **Operation** column of the target API, click **Authorize App**, and then click **Select App**.
- Select the target API, click **Authorize App** over the API list, and then click **Select App**.
- Authorize apps through the API details page.
  - a. Click the name of the target API.
  - b. Click the **Authorization** tab.
  - c. Click **Select App**.

 **NOTE**

To authorize an app to access multiple APIs, select the APIs, and click **Authorize App**. Click **Select App**, select the app you wish to authorize, and click **OK**. You can grant access to a maximum of 1000 APIs at a time.

**Step 6** Select an environment, search for and select desired apps, and click **OK**.

Select App

Environment  App name

<input type="checkbox"/> App Name	App ID	Description
<input type="checkbox"/> App_lr0c33	6800a756aca746b7b80bd0464e3466bc	--

**Step 7** After the authorization is complete, view the authorized apps on the **Authorization** tab page or the **Authorize App** page.

 **NOTE**

If an app does not need to call the API, click **Cancel Authorization** in the row containing the app to unbind it.

----End

## Follow-Up Operations

After you authorize an app to call an API, the API can be called using SDKs of different programming languages.

## 4.5 Publishing an API

### Scenario

APIs can be called only after they have been published in an environment. You can publish APIs in different environments. API Gateway allows you to view the

publication history (such as the version, description, time, and environment) of each API, and supports rollback of APIs to different historical versions.

 **NOTE**


- If you modify a published API, you must publish it again for the modifications to take effect in the environment in which the API has been published.
- A maximum of 10 publication records of an API are retained in an environment.

## Prerequisites

- You have created an API group and API.
- You have created an environment.

## Publishing an API

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Publish an API. You can use one of the following methods:

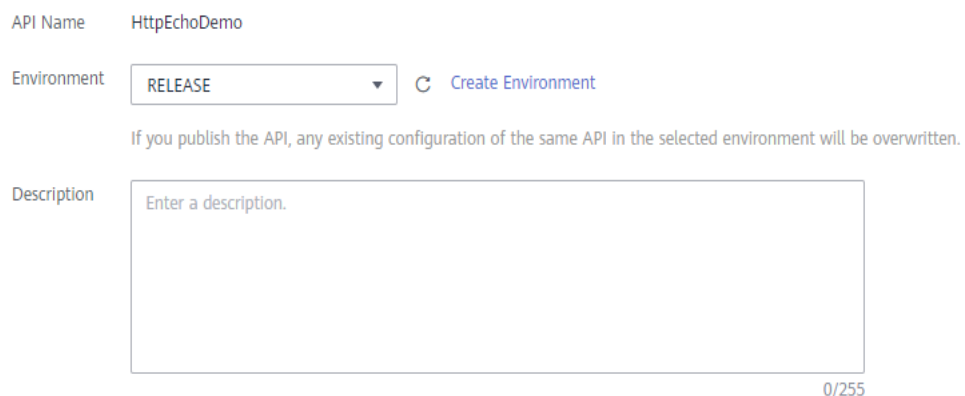
- Click **Publish** in the row containing the API you want to publish.
- Click the name of the target API, and click **Publish** in the upper right corner of the displayed API details page.

 **NOTE**

To publish multiple APIs, select the APIs, and click **Publish**. You can publish a maximum of 1000 APIs at a time.

**Step 6** Select the environment where the API will be published, and enter a description.

**Figure 4-11** Publishing an API



The screenshot shows a form for publishing an API. It includes the following fields and elements:

- API Name:** HttpEchoDemo
- Environment:** A dropdown menu with 'RELEASE' selected, and a 'Create Environment' button.
- Description:** A text input field with the placeholder 'Enter a description.' and a character count '0/255' at the bottom right.
- Warning:** A note below the environment field states: 'If you publish the API, any existing configuration of the same API in the selected environment will be overwritten.'

 **NOTE**


- If the API has already been published in the environment, publishing it again will overwrite its definition in that environment.
- If there is no environment that meets your requirements, create a new one.

**Step 7** Click **Publish**.

----End

## Viewing Publication History

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

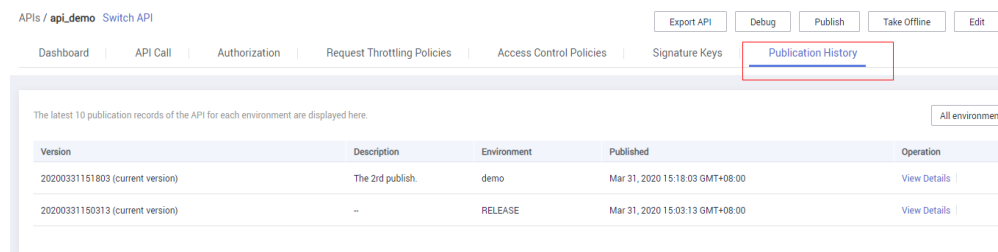
**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Click the name of the target API.

**Step 6** Click the **Publication History** tab.

The publication history of the API is displayed.

**Figure 4-12** Viewing publication history



**Step 7** Click **View Details** in the **Operation** column of a version.

The **View Details** dialog box displays the basic information, frontend and backend request information, input and constant parameters, parameter mappings, and example responses of the API.

**Step 8** To roll back the API to a historical version, click **Switch Version** in the row containing the target version, and click **Yes**.

If "current version" is displayed next to the target version, the rollback was successful.

When the API is called, configuration of the current version is used instead of the previously saved configuration.

For example, an API was published in the RELEASE environment on August 1, 2018. On August 20, 2018, the API was published in the same environment after



modification. If the version published on August 1 is set as the current version, configuration of this version will be used when the API is called.

----End

## 4.6 Taking an API Offline

### Scenario

You can remove APIs that you do not need from the environments where the APIs have been published.

#### NOTICE


This operation will cause the APIs to be inaccessible in the environments. Ensure that you have notified users before this operation.

### Prerequisites

- You have created an API group and API.
- You have published the API.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Take the API offline. You can use one of the following methods:

- In the **Operation** column of the target API, choose **More > Take Offline**.
- Click the name of the target API, and click **Take Offline** in the upper right corner of the API details page.

#### NOTE

To take multiple APIs offline, select the APIs, and click **Take Offline**. You can take a maximum of 1000 APIs offline at a time.

**Step 6** Select the environment from which you want to take the API offline, and click **Yes**.

----End

### Follow-Up Operations

After taking an API offline, delete it based on the instructions provided in [Deleting an API](#).

## 4.7 Deleting an API

### Scenario


You can delete published APIs you no longer require.

#### NOTICE

- Deleted APIs cannot be accessed by apps or users who were using the APIs, so make sure you notify users before the deletion.
- Published APIs must be first taken offline and then deleted.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Delete the API. You can use one of the following methods:

- In the **Operation** column of the API you want to delete, choose **More > Delete**.
- Click the name of the target API, and click **Delete** in the upper right corner of the displayed API details page.

#### NOTE

To delete multiple APIs, select the APIs, and click **Delete**. You can delete a maximum of 1000 APIs at a time.

**Step 6** Enter **DELETE** and click **Yes**.

----End

## 4.8 Importing APIs

### Scenario

API Gateway allows you to import Swagger 2.0 APIs to existing or new API groups. Swagger is an open-source tool built based on OpenAPI specifications to design, build, record, and use REST APIs.


You can import APIs individually or in batches depending on the number of APIs contained in a Swagger file.

## Prerequisites

- The API Swagger file to be imported is available and already has extended API definitions supplemented. For more information, see "Extended Definition" in the *Developer Guide*.
- You have sufficient quotas of API groups and APIs.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > APIs**.

**Step 5** Click **Import API**.

**Step 6** Set the parameters listed in [Table 4-12](#).

**Figure 4-13** Importing APIs



**Table 4-12** Parameters for importing APIs

Parameter	Description
Import	Options: <ul style="list-style-type: none"> <li>• <b>New group</b>: Import APIs to a new API group. If you select this option, the system automatically creates an API group and imports the APIs into this group.</li> <li>• <b>Existing group</b>: Import APIs to an existing API group. If you select this option, the system adds the APIs to the selected API group while retaining the existing APIs in the API group.</li> </ul>
API group	Select an API group if you set <b>Import</b> to <b>Existing group</b> .
Basic Definition Overwrite	Determine whether to overwrite an existing API if the name of the API is the same as that of an imported API. This parameter is available only if you set <b>Import</b> to <b>Existing group</b> .

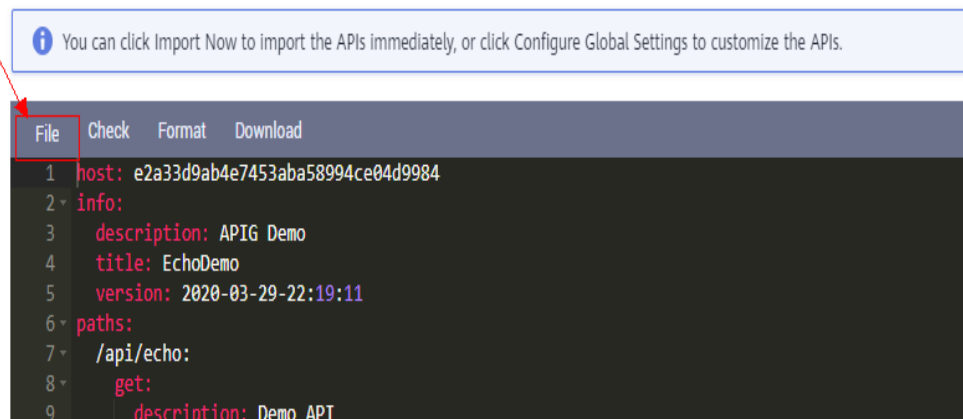
Parameter	Description
Extended Definition Overwrite	If this option is selected, the extended definition items (access control and request throttling policies) of an imported API will overwrite the existing policies with the same name.

**Step 7** In the **Parameter Import** area, click **File** and select a file to import.

YAML and JSON files are supported. You can preview the API content to be imported on the **Import API** page.

**Figure 4-14** Parameter Import

Parameter Import



**Step 8** (Optional) Configure global settings for the APIs to be imported.

You can configure the global settings for the APIs, such as frontend and backend requests, or modify other parameters of the APIs.

**Step 9** Click **Import Now** to import the APIs.

**NOTE**

Imported APIs must be manually published so that they become available for users to access.

----End

## Follow-Up Operations

**Publish** the imported API in an environment so that it can be called by users.

## 4.9 Exporting APIs

### Scenario


You can export APIs one by one or in batches as JSON or YAML files.

## Prerequisites

You have created an API group and API.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** Click **Export API**.

**Step 5** Set the parameters listed in [Table 4-13](#).

**Table 4-13** Parameters for exporting APIs

Parameter	Description
API Group	Select the API group from which APIs will be exported.
Environment	Select the environment where the APIs to be exported have been published.
APIs	By default, all APIs in the API group that have been published in the selected environment are exported. To export only specific APIs, click <b>Select API</b> , and specify the APIs you want to export.
API Definition	<ul style="list-style-type: none"> <li>• <b>Basic:</b> The basic definition of an API is composed of the request and response definitions. It does not include the backend definition. The request definition includes both standard and extended Swagger fields.</li> <li>• <b>Full:</b> The full definition of an API is composed of the request, backend, and response definitions.</li> <li>• <b>Extended:</b> The extended definition of an API is composed of the request, backend, and response definitions as well as the request throttling policy, access control policy, and other configurations of the API.</li> </ul>
Format	Export APIs in <b>JSON</b> or <b>YAML</b> format.
Version	Set the version of the APIs to be exported. If you do not specify a version, the version will be set as the current date and time.

**Step 6** Click **Export**.

The export result is displayed on the right.

----End

# 5 Request Throttling

---

## 5.1 Creating a Request Throttling Policy

### Scenario

Request throttling controls the number of times an API can be called within a time period to protect backend services.

To provide stable, uninterrupted services, you can create request throttling policies to control the number of calls made to your APIs.

Request throttling policies take effect for an API only if they have been bound to the API.

#### NOTE


- An API can be bound with only one request throttling policy for a given environment, but each request throttling policy can be bound to multiple APIs.
- For a dedicated gateway, the limit is the value of **ratelimit\_api\_limits** you have configured on the **Configuration Parameters** page.

### Prerequisites

You have [published the API](#) to which you want to bind a request throttling policy.

### Creating a Request Throttling Policy

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Request Throttling**.

**Step 5** Click **Create Request Throttling Policy**, and set the parameters listed in [Table 5-1](#).

### Create Request Throttling Policy

\* Name   
Enter 3 to 64 characters, starting with a letter. Only letters, digits, and underscores ( \_ ) are allowed.

Type API-based API-shared

\* Period

\* Max. API Requests

Max. User Requests  ( $\leq$  Max. API Requests)

Max. App Requests  ( $\leq$  Max. User Requests)

Max. IP Address Requests  ( $\leq$  Max. API Requests)

Description   
0/255

OK Cancel

**Table 5-1** Parameters for creating a request throttling policy

Parameter	Description
Name	Request throttling policy name.
Type	API-based or API-shared request throttling. <ul style="list-style-type: none"> <li><b>API-based:</b> Request throttling is based on every API to which the policy is bound.</li> <li><b>API-shared:</b> Request throttling is based on all APIs as a whole to which the policy is bound.</li> </ul>
Period	For how long you want to limit the number of API calls. This parameter can be used together with the following parameters: <ul style="list-style-type: none"> <li><b>Max. API Requests:</b> Limit the maximum number of times an API can be called within a specific period.</li> <li><b>Max. User Requests:</b> Limit the maximum number of times an API can be called by a user within a specific period.</li> <li><b>Max. App Requests:</b> Limit the maximum number of times an API can be called by an app within a specific period.</li> <li><b>Max. IP Address Requests:</b> Limit the maximum number of times an API can be called by an IP address within a specific period.</li> </ul>

Parameter	Description
Max. API Requests	The maximum number of times each bound API can be called within the specified period. This parameter must be used together with <b>Period</b> .
Max. User Requests	The maximum number of times each bound API can be called by a user within the specified period. <b>This limit only applies to APIs that are accessed through IAM authentication.</b> <ul style="list-style-type: none"> <li>The value of this parameter cannot exceed that of <b>Max. API Requests</b>.</li> <li>This parameter must be used together with <b>Period</b>.</li> <li>If there are many users under your account that access an API, the request throttling limits of the API will apply to all these users.</li> </ul>
Max. App Requests	The maximum number of times each bound API can be called by an app within the specified period. <b>This limit only applies to APIs that are accessed through app authentication.</b> <ul style="list-style-type: none"> <li>The value of this parameter cannot exceed that of <b>Max. User Requests</b>.</li> <li>This parameter must be used together with <b>Period</b>.</li> </ul>
Max. IP Address Requests	The maximum number of times each bound API can be called by an IP address within the specified period. <ul style="list-style-type: none"> <li>The value of this parameter cannot exceed that of <b>Max. API Requests</b>.</li> <li>This parameter must be used together with <b>Period</b>.</li> </ul>
Description	Description of the request throttling policy.

**Step 6** Click **OK**.

After the policy is created, it is displayed on the **Request Throttling** page. You can bind this policy to APIs to throttle API requests.

----End

## Binding a Request Throttling Policy to an API

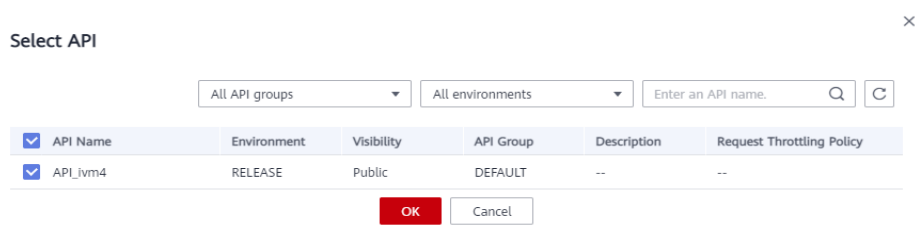
**Step 1** Go to the page for binding a request throttling policy to an API. You can use one of the following methods:

- In the **Operation** column of the request throttling policy to be bound, click **Bind to API**, and then click **Select API**.
- Click the name of the target request throttling policy, and click **Select API** on the **APIs** tab page.

**Step 2** Specify an API group, environment, and API name keyword to search for the desired API.

**Step 3** Select the API and click **OK**.



**Figure 5-1** Binding a request throttling policy to an API**NOTE**

If a request throttling policy is no longer needed for an API, you can unbind it. To unbind a request throttling policy from multiple APIs, select the APIs, and click **Unbind**. You can unbind a request throttling policy from a maximum of 1000 APIs at a time.

----End

## Follow-Up Operations

To control the maximum number of API calls received from a specific app or tenant, specify the app or tenant to exclude by referring to [Adding an Excluded App or Tenant](#). If an app is excluded in a request throttling policy, any threshold configured for that app takes precedence over the request throttling policy. The API and user request limits of this policy are still valid. If a tenant is excluded in a request throttling policy, any threshold configured for that tenant will be applied. The API and app request limits of this policy are still valid.

## 5.2 Deleting a Request Throttling Policy


### Scenario

You can delete request throttling policies you no longer require.

### Prerequisites

You have created a request throttling policy.

### Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing** > **Request Throttling**.
- Step 5** Delete a request throttling policy. You can use one of the following methods:
  - In the **Operation** column of the request throttling policy you want to delete, click **Delete**.

- Click the name of the target request throttling policy, and click **Delete** in the upper right corner of the displayed request throttling policy details page.

 **NOTE**

- If a request throttling policy has been bound to an API, unbind the policy and then delete it. To unbind a request throttling policy, go to the policy details page, click **Unbind** in the row that contains the API from which you want to unbind the policy, and click **Yes**.
- To delete multiple request throttling policies, select the policies, and click **Delete**. You can delete a maximum of 1000 request throttling policies at a time.

**Step 6** Click **Yes**.

----End

## 5.3 Adding an Excluded App or Tenant

### Scenario

If you want to control the number of API calls received from a specific app or tenant, add an excluded app or tenant to a request throttling policy.

### Prerequisites

You have created an app or obtained an app ID of another account or an account ID.

### Adding an Excluded App

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Request Throttling**.

**Step 5** Click the name of the target request throttling policy.

**Step 6** On the displayed request throttling policy details page, click the **Excluded Apps** tab.

**Step 7** Click **Select Excluded App**.

**Step 8** Select an app to exclude. You can use one of the following methods:

**Figure 5-2** Adding an excluded app

### Select Excluded App

App  Existing  Cross-tenant

Threshold  per 1 minute

≤ Max. API Requests


- To select an existing app, click **Existing**, select an app, and enter a threshold.
- To select an app of other tenants, click **Cross-tenant**, and enter the app ID and a threshold.

 **NOTE**


The threshold must be a positive integer and cannot exceed the value of **Max. API Requests**.

----End

## Adding an Excluded Tenant

- Step 1** Log in to the management console.
- Step 2** Hover the mouse pointer over the username, click the username, and choose **My Credentials** from the drop-down list.
- Step 3** On the **API Credentials** page, view the account ID and project ID.
- Step 4** Click  in the upper left corner and choose **API Gateway**.
- Step 5** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 6** In the navigation pane, choose **API Publishing > Request Throttling**.
- Step 7** Click the name of the target request throttling policy.
- Step 8** Click the **Excluded Tenants** tab.
- Step 9** Click **Select Excluded Tenant**.
- Step 10** In the **Select Excluded Tenant** dialog box, set the parameters listed in [Table 5-2](#).

**Figure 5-3** Adding an excluded tenant

\* Account ID 

\* Threshold  per 1 minute

≤ Max. API Requests

**Table 5-2** Excluded tenant configuration

Parameter	Description
Account ID	Account ID or project ID obtained in <a href="#">Step 3</a> . <ul style="list-style-type: none"> <li>Enter a project ID if you will bind or have bound this policy to an API that uses app authentication.</li> <li>Enter an account ID if you will bind or have bound this policy to an API that uses IAM authentication.</li> </ul>
Threshold	The maximum number of times an API can be called by the tenant within a specified period. The value of this parameter cannot exceed that of <b>Max. API Requests</b> .

**Step 11** Click **OK**.

 **NOTE**

Excluded tenant thresholds take precedence over the value of **Max. User Requests**.  
For example, suppose a request throttling policy is configured, with **Max. API Requests** being **10**, **Max. User Requests** being **3**, **Period** being 1 minute, and two excluded tenants (max. 2 API requests for tenant A and max. 4 API requests for tenant B). If the request throttling policy is bound to an API, tenants A and B can access the API 2 and 4 times within 1 minute, respectively.

----End

## 5.4 Removing an Excluded App or Tenant


### Scenario

You can remove excluded apps or tenants from a request throttling policy. This section takes an excluded app as an example.


### Prerequisites

- You have created a request throttling policy.
- You have already added an excluded app or tenant to the request throttling policy.

## Removing an Excluded App

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > Request Throttling**.
- Step 5** Click the name of the target request throttling policy.
- Step 6** Click the **Excluded Apps** tab on the displayed request throttling policy details page.
- Step 7** In the **Operation** column of the app you want to remove, click **Remove**.
- Step 8** Click **Yes**.
- End

## Removing an Excluded Tenant

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > Request Throttling**.
- Step 5** Click the name of the target request throttling policy.
- Step 6** Click the **Excluded Tenants** tab.
- Step 7** In the **Operation** column of the tenant you want to remove, click **Remove**.
- Step 8** Click **Yes**.
- End

# 6 Access Control

---

## 6.1 Creating an Access Control Policy

### Scenario

Access control policies are a type of security measures provided by API Gateway. You can use them to allow or deny API access from specific IP addresses or accounts.


Access control policies take effect for an API only if they have been bound to the API.

 **NOTE**

Each API can be bound with only one access control policy for a given environment, but each access control policy can be bound to multiple APIs.

### Creating an Access Control Policy

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Access Control**.

**Step 5** Click **Create Access Control Policy**.

**Step 6** In the **Create Access Control Policy** dialog box, set the parameters listed in [Table 6-1](#).

### Create Access Control Policy

\* Name

Enter 3 to 64 characters, starting with a letter. Only letters, digits, and underscores ( \_ ) are allowed.

Restriction Type  IP address  Account name

Specify IP addresses from which API requests are allowed or denied. Do not specify private IP addresses that belong to a VPC.

Effect  Allow  Deny

IP Address	Operation
+ Add IP Address	

**Table 6-1** Parameters for creating an access control policy

Parameter	Description
Name	Access control policy name.
Restriction Type	Type of the source from which API calls are to be controlled. <ul style="list-style-type: none"> <li>● <b>IP address:</b> Specify IP addresses and IP address ranges that are allowed or not allowed to access an API.</li> <li>● <b>Account name:</b> Specify names of the accounts that are allowed or not allowed to access an API.</li> </ul>
Effect	Options: <b>Allow</b> and <b>Deny</b> . Use this parameter along with <b>Restriction Type</b> to control the access of certain IP addresses or accounts to an API.
IP Address	IP addresses and IP address ranges that are allowed or not allowed to access an API You need to set this parameter only if you have set <b>Restriction Type</b> to <b>IP address</b> . <b>NOTE</b> You can set a maximum of 100 IP addresses respectively to allow or deny access.

Parameter	Description
Account Names	<p>Names of the accounts that are allowed or not allowed to access an API. <b>This parameter only applies to APIs that are accessed through IAM authentication.</b></p> <p>You need to set this parameter only if you have set <b>Restriction Type</b> to <b>Account name</b>. You can enter multiple account names and separate them with commas, for example, <b>aaa,bbb</b>.</p> <p><b>NOTE</b> API Gateway performs access control on accounts, not IAM users created using accounts.</p>

**Step 7** Click **OK**. You can bind the policy to APIs to control API access.

----End

## Binding an Access Control Policy to an API

**Step 1** Go to the page for binding an access control policy to an API. You can use one of the following methods:

- In the **Operation** column of the access control policy to be bound, click **Bind to API**, and then click **Select API**.
- Click the name of the target access control policy, and click **Select API**.

**Step 2** Specify an API group, environment, and API name keyword to search for the desired API.

**Step 3** Select the API and click **OK**.

### NOTE

If an access control policy is no longer needed for an API, you can unbind it from that API. To unbind an access control policy from multiple APIs, select the APIs, and click **Unbind**. You can unbind a request throttling policy from a maximum of 1000 APIs at a time.

----End

## 6.2 Deleting an Access Control Policy

### Scenario


You can delete access control policies you no longer require.

### Prerequisites

You have created an access control policy.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.



**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Access Control**.

**Step 5** Delete an access control policy using one of the following methods:

- In the **Operation** column of the access control policy you want to delete, click **Delete**.
- Click the name of the target access control policy, and click **Delete** in the upper right corner of the displayed access control policy details page.

 **NOTE**

- If an access control policy has been bound to APIs, unbind it and then delete it.
- To delete multiple access control policies, select the policies, and click **Delete**. You can delete a maximum of 1000 access control policies at a time.

**Step 6** Click **Yes**.

----End

# 7 Environment Management

## 7.1 Creating an Environment and Environment Variable

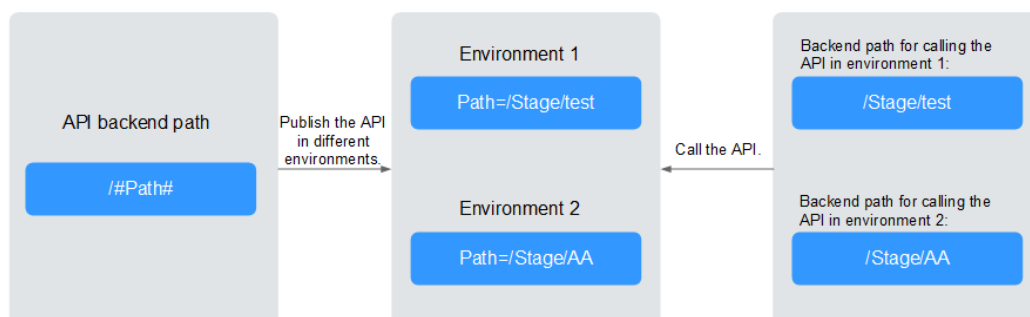
### Scenario

An API can be called in different environments, such as production, testing, and development environments. RELEASE is the default environment provided by API Gateway. You can define environment variables to allow an API to be called in different environments.

Environment variables are manageable and specific to environments. You can create variables in different environments to call different backend services using the same API.

For variables you define during API creation, you must create corresponding variables and values. For example, variable **Path** is defined for an API, and two variables with the same name are created and assigned values **/Stage/test** and **/Stage/AA** in environments 1 and 2, respectively. If the API is published and called in environment 1, the path **/Stage/test** is used. If the API is published and called in environment 2, the path **/Stage/AA** is used.

**Figure 7-1** Use of environment variables



### NOTE


You can create a maximum of 50 variables for an API group in each environment.

## Prerequisites

You have [created an API group](#).

## Creating an Environment

**Step 1** Log in to the management console.

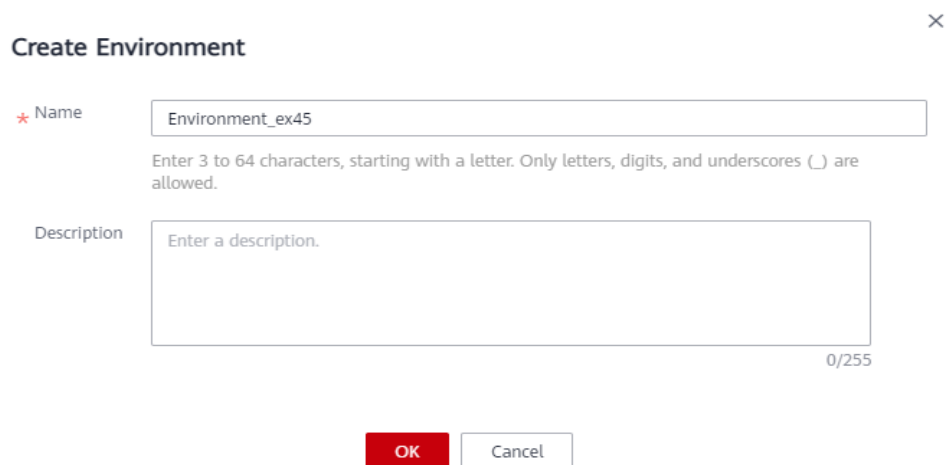
**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Environments**.

**Step 5** Click **Create Environment**, and set the parameters listed in [Table 7-1](#).

**Figure 7-2** Creating an environment



**Table 7-1** Environment information

Parameter	Description
Name	Environment name.
Description	Description of the environment.

**Step 6** Click **OK**.

After the environment is created, it is displayed in the environment list.

----End

## Accessing an Environment


You can call an API in the RELEASE environment by using a RESTful API. To access the API in other environments, add the **X-Stage** header to the request to specify an environment name. For example, add **X-Stage:DEVELOP** to the request header to access an API in the **DEVELOP** environment.

 NOTE

API Gateway does not support API debugging using environment variables.

## Creating an Environment Variable

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > API Groups**.

**Step 5** Create a variable. You can use one of the following methods:

- Click the name of the target API group, and click the **Variables** tab on the displayed API group details page.
- In the **Operation** column of the target API group, choose **More > Manage Variable**.

**Step 6** Select an environment from the **Environment** drop-down list, and click **Create Variable**.

**Step 7** Set the parameters listed in [Table 7-2](#).

**Figure 7-3** Creating an environment variable

★ Name

Enter 3 to 32 characters, starting with a letter. Only letters, digits, hyphens (-), and underscores (\_) are allowed.

Ensure that the variable you create here is consistent with the case-sensitive part that you enclosed within number signs (for example, #varname#) when you created an API. The "#varname#" will be replaced by the value you configure here.

★ Value

0/255

Enter 1 to 255 characters. Only letters, digits, and special characters (-./:) are allowed.

**Table 7-2** Parameters for creating an environment variable

Parameter	Description
Name	Name of the variable you want to create. Ensure that the name is the same as the name of the variable defined for the API.
Value	The path to be used in the selected environment.

**Step 8** Click **OK**.

 **NOTE**

If a variable is not needed, click **Delete** in the row containing the variable to delete it. Environment variable names and values will be displayed in plain text in API requests. Do not include sensitive information in the variable names and values.

----End

## Follow-Up Operations

After creating an environment and variable, **publish APIs** in the environment so that they can be called by API callers.

## 7.2 Deleting an Environment

### Scenario


You can delete environments you no longer require.

### Prerequisites

You have created an environment.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Environments**.

**Step 5** In the **Operation** column of the environment you want to delete, click **Delete**.

 **NOTE**

You can delete an environment only if no APIs have been published in the environment.

**Step 6** Click **Yes**.

----End

# 8 Signature Key Management

---

## 8.1 Creating and Using a Signature Key

### Scenario

Signature keys are used by backend services to verify the identity of API Gateway.

A signature key consists of a key and secret, and can be used only after being bound to an API. When an API bound with a signature key is called, API Gateway adds signature details to the API request. The backend service of the API signs the request in the same way, and verifies the identity of API Gateway by checking whether the signature is consistent with that in the **Authorization** header sent by API Gateway.

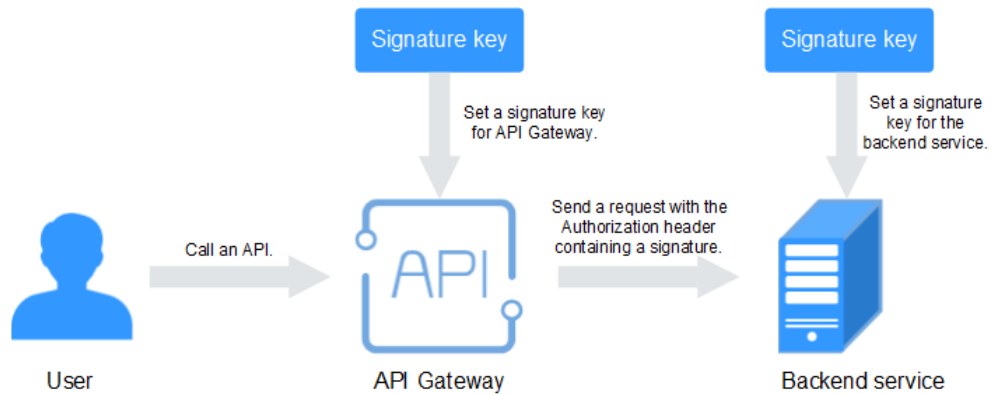
#### NOTE

Each API can only be bound with one signature key in a given environment, but each signature key can be bound to multiple APIs.

### Procedure


1. Create a signature key on the API Gateway console.
2. Bind the signature key to an API.
3. API Gateway sends signed requests containing a signature in the **Authorization** header to the backend service. The backend service can use different programming languages (such as Java, Go, Python, JavaScript, C#, PHP, C++, C, and Android) to sign each request, and check whether the two signatures are consistent.

**Figure 8-1** Signature key process flow



## Creating a Signature Key

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing** > **Signature Keys**.

**Step 5** Click **Create Signature Key**.

**Step 6** In the **Create Signature Key** dialog box, set the parameters listed in [Table 8-1](#).

### Create Signature Key

\* Name   
 Enter 3 to 64 characters, starting with a letter. Only letters, digits, and underscores ( \_ ) are allowed.

\* Type

Key   
 If you do not specify a key, the system will automatically generate a key.

Secret   
 If you do not specify a secret, the system will automatically generate a secret.

Confirm Secret

**Table 8-1** Parameters for creating a signature key

Parameter	Description
Name	Signature key name.
Type	Type of the signature key. Select <b>HMAC</b> or <b>Basic</b> . This parameter is available only for dedicated gateways.
Key	Combined with <b>Secret</b> to form a signature key pair. <ul style="list-style-type: none"><li>• If you set <b>Type</b> to <b>HMAC</b>, enter the key of the key pair used for hash-based message authentication code (HMAC) authentication.</li><li>• If you set <b>Type</b> to <b>Basic</b>, enter the username used for basic authentication.</li></ul>
Secret	Combined with <b>Key</b> to form a signature key pair. <ul style="list-style-type: none"><li>• If you set <b>Type</b> to <b>HMAC</b>, enter the secret of the key pair used for HMAC authentication.</li><li>• If you set <b>Type</b> to <b>Basic</b>, enter the password used for basic authentication.</li></ul>
Confirm Secret	Enter the secret again.

**Step 7** Click **OK**.

----End

## Binding a Signature Key to an API

**Step 1** In the navigation pane, choose **API Publishing > Signature Keys**.

**Step 2** Bind a signature key to an API. You can use one of the following methods:

- In the **Operation** column of the signature key to be bound to an API, click **Bind to API**.
- Click the name of the target signature key.

**Step 3** Click **Select API**.

**Step 4** Specify an API group, environment, and API name keyword to search for the desired API.

**Step 5** Select the API and click **OK**.

### NOTE

If a signature key is no longer needed for an API, unbind it from the API.

----End

## Verifying the Signing Result

Sign each backend request by following the instructions in section "Creating Signatures for Backend Requests" of the *API Gateway Developer Guide*, and check



whether the backend signature is consistent with the signature in the **Authorization** header of the API request.

## 8.2 Deleting a Signature Key

### Scenario


You can delete signature keys you no longer require.

### Prerequisites

You have created a signature key.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Signature Keys**.

**Step 5** Delete a signature key. You can use one of the following methods:

- In the **Operation** column of the signature key you want to delete, click **Delete**.
- Click the name of the target signature key, and click **Delete** in the upper right corner of the displayed signature key details page.

 **NOTE**

If the signature key has been bound to any APIs, unbind it and then delete it.

**Step 6** Click **Yes**.

----**End**

# 9 VPC Channel Management

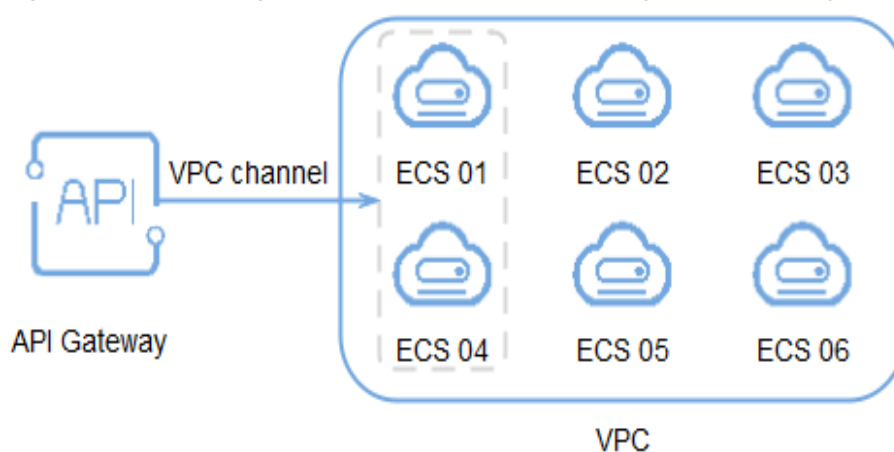
## 9.1 Creating a VPC Channel

### Scenario

VPC channels allow services deployed in VPCs to be accessed through their subnets, lowering latency and balancing loads of backend services.

After creating a VPC channel, you can configure it for an API with an HTTP/HTTPS backend service. For example, six ECSs have been deployed in a VPC, and a VPC channel has been created to reach ECS 01 and ECS 04. API Gateway can access these two ECSs through the VPC channel.


**Figure 9-1** Accessing ECSs in a VPC channel through API Gateway



### Prerequisites

- You have created a cloud server.
- You have the **VPC Administrator** permission.

## Creating a Fast Channel

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > VPC Channels**.
- Step 5** Click **Create VPC Channel**, and set the parameters listed in [Table 9-1](#).

**Figure 9-2** Creating a VPC channel

**Basic Information**

\* Name

\* Port

Member Type Instance IP address

Routing Algorithm WRR WLC SH URI hashing

Forwards requests to each cloud server sequentially according to cloud server weights.

---

**Health Check Configuration**

API Gateway regularly checks the health status of cloud servers associated with the VPC channel. Learn how to configure health check.

Protocol ? TCP HTTP HTTPS

---

**Advanced Settings** ^

Check Port ?

Healthy Threshold ? -  + times/

Unhealthy Threshold ? -  + times/

Timeout (s) ? -  +

Interval (s) ? -  +

**Table 9-1** Parameters for creating a VPC channel

Parameter	Description
Name	VPC channel name.

Parameter	Description
Port	The host port of the VPC channel, that is, the port of the backend service. Range: 1–65535.
Member Type	Select a method that you want to use to specify servers for the VPC channel. The member type is a one-time configuration and cannot be changed after you create the VPC channel. <ul style="list-style-type: none"> <li>● <b>Instance</b>: Select cloud servers.</li> <li>● <b>IP address</b>: Specify cloud server IP addresses.</li> </ul>
Routing Algorithm	The algorithm to be used to forward requests to cloud servers you select. The following routing algorithms are available: <ul style="list-style-type: none"> <li>● <b>WRR</b>: weighted round robin</li> <li>● <b>WLC</b>: weighted least connection</li> <li>● <b>SH</b>: source hashing</li> <li>● <b>URI hashing</b></li> </ul>
Protocol	The protocol used to perform health checks on cloud servers associated with the VPC channel. Options: <ul style="list-style-type: none"> <li>● TCP</li> <li>● HTTP</li> <li>● HTTPS</li> </ul> Default value: <b>TCP</b> .
Path	The destination path for health checks. Set this parameter only when <b>Protocol</b> is not set to <b>TCP</b> .
Check Port	The destination port for health checks. By default, the port of the VPC channel will be used.
Healthy Threshold	The number of consecutive successful checks required for a cloud server to be considered healthy. Range: 2–10. Default value: <b>2</b> .
Unhealthy Threshold	The number of consecutive failed checks required for a cloud server to be considered unhealthy. Range: 2–10. Default value: <b>5</b> .
Timeout (s)	The timeout used to determine whether a health check has failed. Unit: s. Range: 2–30. Default value: <b>5</b> .
Interval (s)	The interval between consecutive checks. Unit: s. Range: 5–300. Default value: <b>10</b> .

Parameter	Description
Response Codes	The HTTP codes used to check for a successful response from a target. Set this parameter only when <b>Protocol</b> is not set to <b>TCP</b> .

**Step 6** Click **Next**.

**Step 7** Click **Select Cloud Server**.

**Step 8** Select the cloud servers you want to add, and click **OK**.

 **NOTE**

To ensure a successful health check and service availability, configure the security groups of the cloud servers to allow access from 100.125.0.0/16.

**Step 9** Click **Finish**.

----End

## Follow-Up Operations

[Create an API](#) for backend services deployed in a VPC to balance loads.

## 9.2 Deleting a VPC Channel

### Scenario

You can delete VPC channels you no longer require.

 **NOTE**


VPC channels that are currently in use by published APIs cannot be deleted.

### Prerequisites

You have created a VPC channel.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > VPC Channels**.

**Step 5** Delete a VPC channel. You can use one of the following methods:

- In the **Operation** column of the VPC channel you want to delete, click **Delete**.

- Click the name of the target VPC channel, and click **Delete** in the upper right corner of the displayed VPC channel details page.

**Step 6** Click **Yes**.

----End

## 9.3 Editing Health Check Configurations

### Scenario


You can modify the health check configurations of a VPC channel to meet service requirements.

### Prerequisites

You have created a VPC channel.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > VPC Channels**.

**Step 5** Click the name of the target VPC channel,

**Step 6** Click the **Health Check** tab.

**Step 7** Click **Edit Health Check**.

**Step 8** In the **Edit Health Check Configuration** dialog box, modify the parameters listed in [Table 9-2](#).

## Edit Health Check Configuration

Name VPC\_ecwd

Protocol ? TCP HTTP HTTPS

Check Port ? 80

Healthy Threshold ? - 2 +

Unhealthy Threshold ? - 5 +

Timeout (s) ? - 5 +

Interval (s) ? - 10 +

OK Cancel

**Table 9-2** Health check configurations

Parameter	Description
Protocol	The protocol used to perform health checks on cloud servers associated with the VPC channel. Options: <ul style="list-style-type: none"> <li>• TCP</li> <li>• HTTP</li> <li>• HTTPS</li> </ul> Default value: <b>TCP</b> .
Path	The destination path for health checks. Set this parameter only when <b>Protocol</b> is not set to <b>TCP</b> .
Check Port	The destination port for health checks. By default, the port of the VPC channel will be used.
Healthy Threshold	The number of consecutive successful checks required for a cloud server to be considered healthy. Range: 2-10. Default value: <b>2</b> .

Parameter	Description
Unhealthy Threshold	The number of consecutive failed checks required for a cloud server to be considered unhealthy. Range: 2–10. Default value: 5.
Timeout (s)	The timeout used to determine whether a health check has failed. Unit: s. Range: 2–30. Default value: 5.
Interval (s)	The interval between consecutive checks. Unit: s. Range: 5–300. Default value: 10.
Response Codes	The HTTP codes used to check for a successful response from a target. Set this parameter only when <b>Protocol</b> is not set to <b>TCP</b> .

**Step 9** Click **OK**.

----End

## 9.4 Editing Cloud Server Configurations of a VPC Channel

### Scenario

You can add or remove cloud servers and edit cloud server weights for VPC channels to meet service requirements.

### Prerequisites

You have created a VPC channel.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > VPC Channels**.

**Step 5** Click the name of the target VPC channel.

**Step 6** Click the **Cloud Servers** tab.

**Step 7** Add or remove cloud servers and edit cloud server weights.



- Adding cloud servers



- a. Click **Select Cloud Server**.
- b. Select the cloud servers you want to add, set cloud server weights, and click **OK**.

 **NOTE**

To ensure a successful health check and service availability, configure the security groups of the backend cloud servers to allow access from 100.125.0.0/16.

- Removing cloud servers
  - a. In the **Operation** column of the cloud servers you want to remove, click **Remove**.
  - b. Click **Yes**.
- Editing the weight of a cloud server
  - a. In the **Weight** column of the target cloud server, click .
  - b. Change the weight and click .
- Editing the weights of multiple cloud servers
  - a. Select the cloud servers to be edited, and click **Edit Weight**.
  - b. Change the weights of the selected cloud servers, and click **OK**.

----End

# 10 Custom Authorizers

---

## 10.1 Creating a Custom Authorizer

---

**NOTICE**

This feature is currently unavailable because FunctionGraph has not been launched.

---

### Scenario

API Gateway supports custom authentication of both frontend and backend requests.

- Frontend custom authentication: If you already have an authentication system, you can configure it in a function and then create a custom authorizer by using the function to authenticate API requests.
- Backend custom authentication: You can create a custom authorizer to authenticate requests for different backend services, eliminating the need to customize APIs for different authentication systems and simplifying API development. You only need to create a function-based custom authorizer in API Gateway to connect to the backend authentication system.

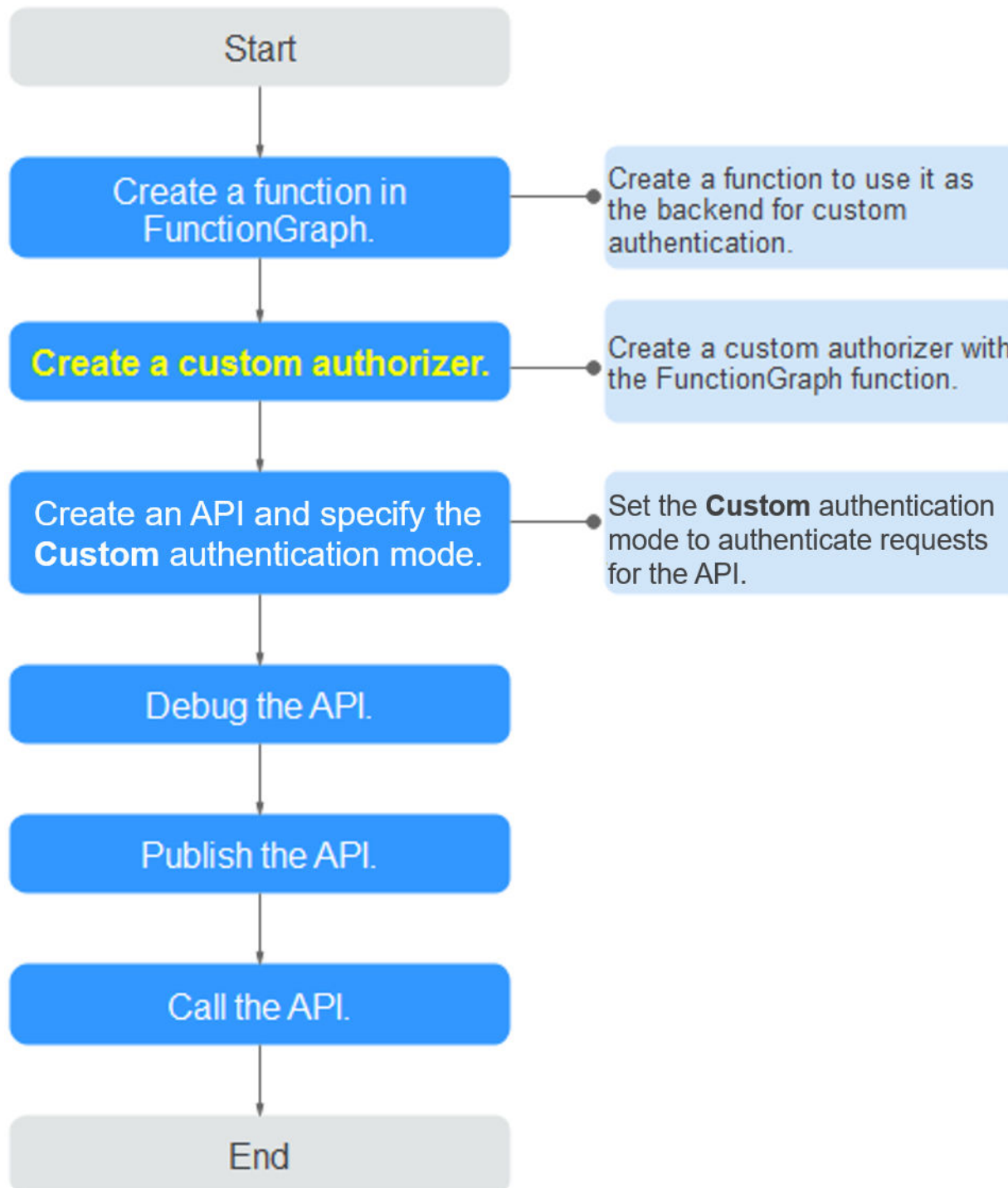
** NOTE**

Custom authentication is implemented using FunctionGraph and not supported if FunctionGraph is unavailable in the selected region.

For details about custom authentication, see *Developer Guide*.

The following figure shows the process of calling APIs through custom authentication.

**Figure 10-1** Calling APIs through custom authentication




### Prerequisites

- You have created a function in FunctionGraph.
- You have the **FunctionGraph Administrator** permission.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** Choose **API Publishing > Custom Authorizers**, and click **Create Custom Authorizer**.


**Step 5** Set the parameters listed in [Table 10-1](#).

### Create Custom Authorizer


\* Name

\* Type  Frontend  Backend


\* Function URN  [Select](#)

Identity Sources 


Parameter Location	Parameter Name	Operation
+ Add Identity Source		

\* Max. Cache Age (s) 

Send Request Body

User Data 

0/2,048

 The user data will be stored in plaintext format. Be careful with information that you include here.

**Table 10-1** Parameters for creating a custom authorizer

Parameter	Description
Name	Authorizer name.
Type	<ul style="list-style-type: none"> <li>● <b>Frontend</b>: Authenticates access to APIs.</li> <li>● <b>Backend</b>: Authenticates access to backend services.</li> </ul>
Function URN	Select a FunctionGraph function.

Parameter	Description
Identity Sources	Request parameters for authentication. You can add headers and query strings. Header names are case-insensitive. This parameter is mandatory only if you set <b>Type</b> to <b>Frontend</b> , and <b>Max. Cache Age (s)</b> is greater than <b>0</b> . When the cache is used, this parameter is used as a search criterion to query authentication results.
Max. Cache Age (s)	The time for caching authentication results. Value <b>0</b> means that authentication results will not be cached. The maximum value is <b>3600</b> .
Send Request Body	Determine whether to send the body of each API request to the authentication function. If you enable this option, the request body will be sent to the authentication function in the same way as the headers and query strings.
User Data	Customized request parameters to be used together with <b>Identity Sources</b> when API Gateway invokes a function.

**Step 6** Click **OK**.

----End

## 10.2 Deleting a Custom Authorizer

### Scenario

You can delete custom authorizers you no longer require.

#### NOTE

- Custom authentication is implemented using FunctionGraph and not supported if FunctionGraph is unavailable in the selected region.
- Custom authorizers that have been configured for APIs cannot be deleted.

### Prerequisites

You have [created a custom authorizer](#).

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** Choose **API Publishing > Custom Authorizers**, and click **Delete** in the row containing the custom authorizer you want to delete.

**Step 5** Click **Yes**.

----End

# 11 Plug-ins

---

## 11.1 Creating a Plug-in


API Gateway provides flexible extension capabilities for APIs through plug-ins.

### Guidelines for Using Plug-ins

- An API can be bound with only one plug-in of the same type.
- Plug-ins are independent of APIs. A plug-in takes effect for an API only after they are bound to each other. When binding a plug-in to an API, you must specify an environment where the API has been published. The plug-in takes effect for the API only in the specified environment.
- After you bind a plug-in to an API, unbind the plug-in from the API, or update the plug-in, you do not need to publish the API again.
- Taking an API offline does not affect the plug-ins bound to it. The plug-ins are still bound to the API if the API is published again.
- Plug-ins that have been bound to APIs cannot be deleted.

### Creating a Plug-in

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Plug-ins**.

**Step 5** Click **Create Plug-in**.

In the **Create Plug-in** dialog box, configure the plug-in information.

### Create Plug-in

\* Plug-in Name

\* Plug-in Type

Plug-in Content

Description

0/255

**Table 11-1** Plug-in configuration

Parameter	Description
Plug-in Name	Name of the plug-in you want to create. It is recommended that you enter a name based on certain naming rules to facilitate identification and search.
Plug-in Type	Type of the plug-in, which determines the extension capabilities of the plug-in. <ul style="list-style-type: none"> <li>● <b>CORS</b>: Specifies preflight request headers and response headers and automatically creates preflight request APIs for cross-origin API access.</li> <li>● <b>HTTP Response Headers</b>: Enables you to customize HTTP response headers that will be displayed in an API response.</li> </ul>
Plug-in Content	Content of the plug-in, which can be configured in a form or using a script. The plug-in content varies depending on the plug-in type: <ul style="list-style-type: none"> <li>● <b>CORS Plug-in</b></li> <li>● <b>HTTP Response Header Management Plug-in</b></li> </ul>
Description	Description of the plug-in.

**Step 6** Click **OK**.

After creating the plug-in, **bind it to the API** for which the plug-in will take effect.

----End

## Binding a Plug-in to an API

**Step 1** In the navigation pane, choose **API Publishing > APIs**.

**Step 2** Click the name of the target API to go to the API details page.



**Step 3** On the **Plug-ins** tab page, click **Bind**.

**Step 4** In the **Bind Plug-in** dialog box, select an environment and plug-in type, and select the plug-in to bind.

**Step 5** Click **OK**.

----End

## 11.2 CORS Plug-in

For security purposes, the browser restricts cross-domain requests from being initiated from a page script. In this case, the page can access only the resources from the current domain. CORS allows the browser to send XMLHttpRequest to the server in a different domain. For more information, see [CORS](#).

The CORS plug-in provides the capabilities of specifying preflight request headers and response headers and automatically creating preflight request APIs for cross-origin API access.

### Usage Guidelines

- You have understood the [Guidelines for Using Plug-ins](#).
- APIs with the same request path in an API group can only be bound with the same CORS plug-in.
- If you have enabled CORS for an API and have also bound the CORS plug-in to the API, the CORS plug-in will be used.
- You cannot bind the CORS plug-in to APIs with the same request path as another API that uses the OPTIONS method.
- When you [bind a plug-in to an API](#), ensure that the request method of the API is included in **allow\_methods**.

### Configuration Parameters

**Table 11-2** Configuration parameters

Parameter	Description
allowed origins	<b>Access-Control-Allow-Origin</b> response header, which specifies either a single origin, which tells browsers to allow that origin to access an API; or else — for requests without credentials — the "*" wildcard, to tell browsers to allow any origin to access the API. Separate multiple URIs using commas.
allowed methods	<b>Access-Control-Allow-Methods</b> response header, which specifies the HTTP methods allowed when accessing the API. Separate multiple methods using commas.

Parameter	Description
allowed headers	<p><b>Access-Control-Allow-Headers</b> response header, which specifies request headers that can be used when making an XMLHttpRequest. Separate multiple headers using commas.</p> <p>By default, simple request headers <b>Accept</b>, <b>Accept-Language</b>, <b>Content-Language</b>, and <b>Content-Type</b> (only if the value is <b>application/x-www-form-urlencoded</b>, <b>multipart/form-data</b>, or <b>text/plain</b>) are carried in requests. You do not need to configure these headers in this parameter.</p>
exposed headers	<p><b>Access-Control-Expose-Headers</b> response header, which specifies which response headers can be contained in the response of XMLHttpRequest. Separate multiple headers using commas.</p> <p>By default, basic response headers <b>Cache-Control</b>, <b>Content-Language</b>, <b>Content-Type</b>, <b>Expires</b>, <b>Last-Modified</b>, and <b>Pragma</b> can be contained in the response. You do not need to configure these headers in this parameter.</p>
maximum age	<p><b>Access-Control-Max-Age</b> response header, which specifies for how many seconds the results of a preflight request can be cached. No more preflight requests will be sent within the specified period.</p>
allowed credentials	<p><b>Access-Control-Allow-Credentials</b> response header, which specifies whether XMLHttpRequest requests can carry cookies.</p>

### Example Script

```
{
  "allow_origin": "**",
  "allow_methods": "GET,POST,PUT",
  "allow_headers": "Content-Type,Accept,Accept-Ranges,Cache-Control",
  "expose_headers": "X-Request-Id,X-Apig-Latency",
  "max_age": 172800,
  "allow_credentials": true
}
```

## 11.3 HTTP Response Header Management Plug-in

HTTP response headers are part of the response returned by API Gateway to a client that calls an API. You can customize HTTP response headers that will be contained in an API response.

### Usage Guidelines

You cannot modify the response headers, such as **x-apig-\*** and **x-request-id**, added by API Gateway, or the headers configured for CORS.

## Configuration Parameters

**Table 11-3** Configuration parameters

Parameter	Description
Name	Response header name, which is case-insensitive and must be unique within a plug-in. You can add a maximum of 10 response headers.
Value	Value of the response header. This parameter does not take effect and can be left blank if you set <b>Action</b> to <b>Delete</b> .

Parameter	Description
Action	<p>Response header operation. You can override, append, delete, skip, or add the specified header.</p> <p><b>Override</b></p> <ul style="list-style-type: none"> <li>• The value of this response header will override that of the same header that exists in an API response.</li> <li>• If an API response contains multiple headers with the same name as the one you set here, only the value of the specified header will be returned.</li> <li>• If an API response does not contain the specified header, the value you set here will be returned.</li> </ul> <p><b>Append</b></p> <ul style="list-style-type: none"> <li>• If an API response contains the specified header, the value you set here will be added, following the existing value. The two values will be separated with commas (,).</li> <li>• If an API response contains multiple headers with the same name as the one you set here, values of these headers will be separated with commas (,) and followed by the value of the specified header.</li> <li>• If an API response does not contain the specified header, the value you set here will be returned.</li> </ul> <p><b>Delete</b></p> <ul style="list-style-type: none"> <li>• If an API response contains the specified header, the header will be deleted.</li> <li>• If an API response contains multiple headers with the same name as the one you set here, all these headers will be deleted.</li> </ul> <p><b>Skip</b></p> <ul style="list-style-type: none"> <li>• If an API response contains the specified header, the header will be skipped.</li> <li>• If an API response contains multiple headers with the same name as the one you set here, values of all these headers will be returned without modification.</li> <li>• If an API response does not contain the specified header, the value you set here will be returned.</li> </ul> <p><b>Add</b></p> <p>The value of the specified header will be returned even if the header does not exist in an API response.</p>

### Example Script

```
{
  "response_headers": [
    {
```

```
"name": "test",  
"value": "test",  
"action": "append"  
},  
{  
  "name": "test1",  
  "value": "test1",  
  "action": "override"  
}  
]
```

## 11.4 Deleting a Plug-in

### Scenario


You can delete plug-ins you no longer require. To delete a plug-in that has been bound to APIs, unbind the plug-in from the APIs and then delete it.

### Prerequisites

You have created a plug-in.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Publishing > Plug-ins**.

**Step 5** Click the name of the target plug-in to go to the plug-in details page.

- If the plug-in is not bound to any APIs, click **Delete** in the upper right corner.
- If the plug-in has been bound to APIs, unbind the plug-in from the APIs in the **Bound APIs** area, and then click **Delete** in the upper right corner.

**Step 6** Click **Yes**.

----End

# 12 Monitoring

## 12.1 API Gateway Metrics

### Introduction

This section describes the metrics that API Gateway reports to the Cloud Eye service. You can view metrics and alarms by using the Cloud Eye console.

### Namespace

Dedicated gateway: SYS.APIC

### Metrics

**Table 12-1** Dedicated gateway metrics

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Minute)
requests	Requests	Number of times that all APIs in a dedicated gateway have been called.	$\geq 0$	Dedicated gateway	1
error_4xx	4xx Errors	Number of times that all APIs in the dedicated gateway return a 4xx error.	$\geq 0$	Dedicated gateway	1

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Minute)
error_5xx	5xx Errors	Number of times that all APIs in the dedicated gateway return a 5xx error.	≥0	Dedicated gateway	1
throttled_calls	Throttled API Calls	Number of times that all APIs in the dedicated gateway have been throttled.	≥0	Dedicated gateway	1
avg_latency	Average Latency	Average latency of all APIs in the gateway.	≥0 Unit: ms	Dedicated gateway	1
max_latency	Maximum Latency	Maximum latency of all APIs in the gateway.	≥0 Unit: ms	Dedicated gateway	1
req_count	Requests	Number of times that an API has been called.	≥ 0	API	1
req_count_2xx	2xx Responses	Number of times that the API returns a 2xx response.	≥ 0	API	1
req_count_4xx	4xx Errors	Number of times that the API returns a 4xx error.	≥ 0	API	1
req_count_5xx	5xx Errors	Number of times that the API returns a 5xx error.	≥ 0	API	1
req_count_error	Total Errors	Total number of errors returned by the API.	≥ 0	API	1

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Minute)
avg_latency	Average Latency	Average latency of the API.	≥ 0 Unit: ms	API	1
max_latency	Maximum Latency	Maximum latency of the API.	≥ 0 Unit: ms	API	1
input_throughput	Incoming Traffic	Incoming traffic of the API.	≥ 0 Unit: Byte, KB, MB, or GB	API	1
output_throughput	Outgoing Traffic	Outgoing traffic of the API.	≥ 0 Unit: Byte, KB, MB, or GB	API	1

## Dimension

**Table 12-2** Dedicated gateway monitoring dimensions

Key	Value
instance_id	Dedicated gateway
api_id	API

## 12.2 Creating Alarm Rules

### Scenario

You can create alarm rules to monitor the status of your APIs.

An alarm rule consists of a rule name, monitored objects, metrics, alarm thresholds, monitoring interval, and notification.

### Prerequisites

An API has been called.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.



- Step 3** In the navigation pane, choose **Dedicated Gateways**.
- Step 4** In the navigation pane, choose **API Publishing > APIs**.
- Step 5** Click the name of the target API.
- Step 6** On the **Dashboard** tab page, click **View Metric** to access the Cloud Eye console. Then create an alarm rule. For details, see section "Creating an Alarm Rule" in the *Cloud Eye User Guide*.
- End

## 12.3 Viewing Metrics


### Scenario

Cloud Eye monitors the status of your APIs and allows you to view their metrics.

### Prerequisites

You have created an API group and API.

### Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Publishing > APIs**.
- Step 5** Click the name of the target API.
- API metrics are displayed on the **Dashboard** tab page.
- Step 6** Click **View Metric** to view more metrics on the Cloud Eye console.

 **NOTE**

The monitoring data is retained for two days. To retain the data for a longer period, save it to an OBS bucket.

----End

# 13 App Management

## 13.1 Creating an App and Obtaining Authorization

### Scenario


For an API that uses app authentication, you can create an app and use the app and its ID and key pair (AppKey and AppSecret) to call the API. You can use an app to call an API only after you bind the app to the API. When you call the API, replace the key pair in the SDK with your own key pair so that API Gateway can authenticate your identity. For details about app authentication, see *Developer Guide*.

#### NOTE

- If the authentication mode of the target API has been set to **None** or **IAM**, you do not need to create apps to call this API.

### Creating an App

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Calling > Apps**.

**Step 5** Click **Create App**, and configure the app information.

**Table 13-1** App information

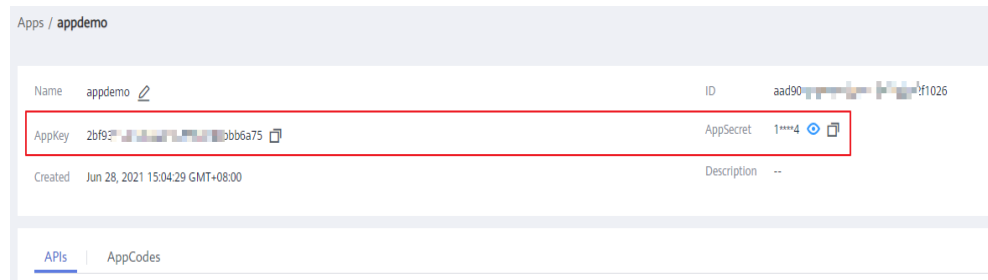
Parameter	Description
Name	App name.
Description	Description of the app.

**Step 6** Click **OK**.

After the app is created, its name and ID are displayed in the app list.

**Step 7** Click the app name, and view the AppKey and AppSecret on the app details page.

**Figure 13-1** App details



----End

## Binding an App to an API

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Calling > Apps**.

**Step 5** Bind an app to an API. You can use one of the following methods:

- In the **Operation** column of the app, click **Bind to API**, and then click **Select API**.
- Click the name of the target app, and click **Select API**.

**Step 6** Select an environment, select an API, and click **OK**.

After the binding is complete, you can view the API on the app details page.

### NOTE

- Only APIs using app authentication can be bound with apps.
- An app can be bound to multiple APIs that use app authentication, and each such API can be bound with multiple apps.
- To debug an API to which the app is bound, click **Debug** in the row containing the API.

----End

## Follow-Up Operations

You can call APIs using different authentication methods. For details, see [Calling APIs](#).

## 13.2 Deleting an App

### Scenario


You can delete apps you no longer require.

### Prerequisites

You have created an app.

### Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Calling > Apps**.

**Step 5** Delete an app. You can use one of the following methods:

- In the **Operation** column of the app you want to delete, click **Delete**.
- Click the name of the target app, and click **Delete App** in the upper right corner of the displayed app details page.

 **NOTE**

If the app has been bound to any APIs, you must unbind the app and then delete it.

**Step 6** Click **Yes**.

----End

## 13.3 Resetting the AppSecret of an App

### Scenario


You can reset the AppSecret of an app. The AppKey is unique and cannot be reset. When you reset the AppSecret, it becomes invalid and APIs bound to the app cannot be called. To enable API calls for that app again, you will need to update the AppSecret of the app you use.

### Prerequisites

You have created an app.

### Procedure

**Step 1** Log in to the management console.

- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Calling > Apps**.
- Step 5** Click the name of the target app.
- Step 6** In the upper right corner of the displayed app details page, click **Reset AppSecret**.
- Step 7** Click **Yes**.
- End

## 13.4 Adding an AppCode for Simple Authentication

### Scenario

AppCodes are identity credentials of an app used to call APIs in simple authentication mode. In this mode, the **X-Apig-AppCode** parameter (whose value is an AppCode on the app details page) is added to the HTTP request header for quick response. API Gateway verifies only the AppCode and the request content does not need to be signed.

When an API is called using app authentication and simple authentication is enabled for the API, AppKey and AppSecret can be used to sign and verify the API request. AppCode can also be used for simple authentication.


#### NOTE

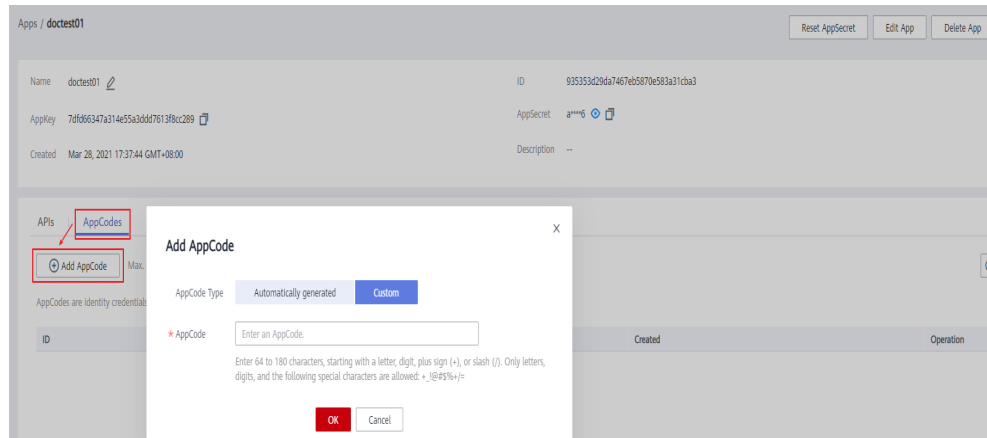
- For security purposes, simple authentication only supports API calls over HTTPS.
- You can create a maximum of five AppCodes for each app.

### Prerequisites

You have created an app.

### Generating an AppCode

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and choose **API Gateway**.
- Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 4** In the navigation pane, choose **API Calling > Apps**.
- Step 5** Click the name of the target app.
- Step 6** Click the **AppCodes** tab.
- Step 7** Click **Add AppCode** to generate an AppCode. It can be automatically generated or customized.



----End

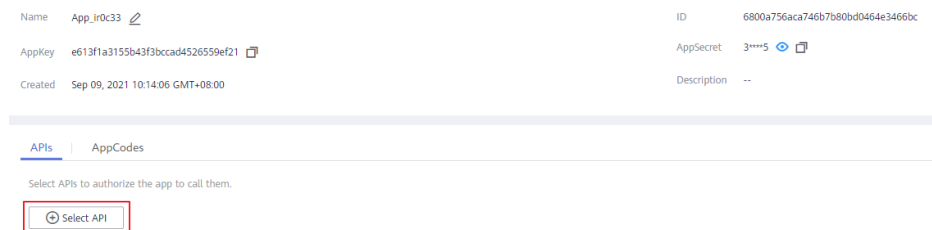
## Using AppCode for Simple Authentication of API Requests

**Step 1** When creating an API, set **Security Authentication** to **App** and enable **Simple Authentication**.

### NOTE

After you enable simple authentication for an existing API, you need to publish the API again to make the configuration take effect.

**Step 2** Bind an app to the API.



**Step 3** When sending a request, add the **X-Apig-AppCode** parameter to the request header and omit the request signature.

For example, when using curl, add the **X-Apig-AppCode** parameter to the request header and set the parameter value to the **generated AppCode**.

```
curl -X GET "https://api.exampledemo.com/testapi" -H "content-type: application/json" -H "host: api.exampledemo.com" -H "X-Apig-AppCode: xhrJVJKABSOxc7d*****FZL4gSHEXkCMQC"
```

----End

## 13.5 Viewing API Details

### Scenario


You can view the details of an API to which an app has been bound.

## Prerequisites

- You have created an app.
- The app has been bound to an API.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.

**Step 4** In the navigation pane, choose **API Calling > Apps**.

**Step 5** Click the name of the target app.

**Step 6** Click the name of the target API to view its details.

----End

# 14 Log Analysis

---




## Scenario

This section describes how to obtain and analyze the API calling logs of dedicated gateways.

## Prerequisites

APIs have been called.

## Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region.
- Step 3** Click  in the upper left corner and choose **API Gateway**.
- Step 4** In the navigation pane, choose **Dedicated Gateways**. Then click **Access Console** in the upper right corner of a dedicated gateway.
- Step 5** Choose **API Calling > Access Logs**, and click **Configure Log Collection**.
- Step 6** Enable log collection (  ).
- Step 7** Specify a log group and log stream, and click **OK**. For details about log groups and log streams, see *Log Tank Service User Guide*.
- Step 8** Click **Log Fields** to view the description of each log field. Then view and analyze logs by referring to the log field descriptions.
- Step 9** To export logs, see section "Log Transfer" in the *Log Tank Service User Guide*.

Fields in access logs are separated using spaces. The following table describes each log field.



**Table 14-1** Log field description

No.	Field	Description
1	remote_addr	Client IP address
2	request_id	Request ID
3	api_id	API ID
4	user_id	Project ID provided by a requester for IAM authentication
5	app_id	App ID provided by a requester for app-based authentication
6	time_local	Time when a request is received
7	request_time	Request latency
8	request_method	HTTP request method
9	host	Domain name
10	router_uri	Request URI
11	server_protocol	Request protocol
12	status	Response status code
13	bytes_sent	Response size in bytes, including the status line, header, and body.
14	request_length	Request length in bytes, including the start line, header, and body.
15	http_user_agent	User agent ID
16	http_x_forwarded_for	X-Forwarded-For header field
17	upstream_addr	Backend address
18	upstream_uri	Backend URI
19	upstream_status	Backend response code
20	upstream_connect_time	Time taken for establishing a connection with the backend
21	upstream_header_time	Duration from the beginning of the establishment of a connection to receiving the first byte from the backend
22	upstream_response_time	Duration from the beginning of the establishment of a connection to receiving the last byte from the backend

No.	Field	Description
23	region_id	Region ID

----End

# 15 SDKs

---

API Gateway supports API authentication based on IAM, apps, and custom authorizers. You can also choose not to authenticate API requests. For details about the differences between the authentication modes, see "How Do I Choose an Authentication Mode" in the *API Gateway Developer Guide*.

This section describes how to download SDKs and view related instructions.


For details about IAM authentication, see section "Calling APIs Through IAM Authentication" in the *API Gateway Developer Guide*.

## Scenario

SDKs are used when you call APIs through app authentication. Download SDKs and related documentation and then call APIs by following the instructions in the documentation.

## Procedure

**Step 1** Log in to the management console.

**Step 2** Click  in the upper left corner and choose **API Gateway**.

**Step 3** Choose **Help Center > SDK Process Flow**.

**Step 4** Click **Download SDK** of the desired language.

----End

# 16 Calling Published APIs

---

## 16.1 Calling APIs

### Obtaining APIs and Documentation

Before calling APIs, obtain the request information from the API provider, including the access domain name, protocol, method, path, and request parameters.

Obtain APIs: from your company or from a partner

Obtain related documentation from the help center of the API provider's official website:

The authentication information to be obtained varies with the API authentication mode.

- App authentication:
  - Signature authentication: Obtain the key and secret (or client AppKey and AppSecret) of the app authorized for the API from the API provider as well as the SDK for calling the API.
  - Simple authentication: Obtain the AppCode of the app authorized for the API from the API provider.
  - Other authentication modes: Obtain the key and secret (or client AppKey and AppSecret) of the app authorized for the API from the API provider.
- IAM authentication: The account credential (token or AK/SK obtained with the account and password) obtained on the cloud service platform is used for authentication. If the AK/SK is used for authentication, you also need to obtain the SDK from the API provider for calling the API.
- Custom authentication: Obtain the custom authentication information to be carried in the request parameters from the API provider.
- None: No authentication information is required.

## Calling an API

### NOTE

This section describes only the configuration of the request path and authentication parameters. For other parameters, such as timeout and SSL, configure them as required. To avoid service loss due to incorrect parameters, configure them by referring to the industry standards.

#### Step 1 Set the request path.

Scenario	Request Parameter Configuration
Calling an API with a domain name	Call the API using <b>the subdomain name allocated to the API group or a domain name bound to the group</b> . No additional configuration is required.
Calling an API in the DEFAULT group with an IP address	In the shared gateway, call an API in the DEFAULT group with an IP address. No additional configuration is required.
Calling an API in a non-DEFAULT group with an IP address	<ul style="list-style-type: none"> <li>To call APIs using an IP address, ensure that the <b>app_route</b> parameter has been set to <b>on</b> on the <b>Configuration Parameters</b> tab page of the dedicated gateway.</li> <li>To use an IP address to call an API that uses app authentication in a non-DEFAULT group, add the header parameters <b>X-HW-ID</b> and <b>X-HW-APPKEY</b> and set the parameter values to the key and secret of an app authorized for the API or a client AppKey and AppSecret.</li> <li>To use an IP address to call an API that does not use app authentication in a non-DEFAULT group, add the header parameter <b>host</b>.</li> </ul>

#### Step 2 Set the authentication parameters.

Authentication Mode	Request Parameter Configuration
App authentication (with a signature)	Use the SDK to sign API requests. For details, see section "Calling APIs Through App Authentication" in the <i>API Gateway Developer Guide</i> .

Authentication Mode	Request Parameter Configuration
App authentication (with app_secret)	<ul style="list-style-type: none"> <li>On the <a href="#">Configuration Parameters</a> tab page of a dedicated gateway, the <b>app_secret</b> parameter has been set to <b>on</b> to enable app_secret authentication and <b>app_api_key</b> has been set to <b>off</b> to disable app_api_key authentication.</li> <li>Add the header parameter <b>X-HW-ID</b> and set the parameter value to the key of the app authorized for the API or the client AppKey.</li> <li>Add the header parameter <b>X-HW-AppKey</b> and set the parameter value to the secret or AppSecret obtained in <a href="#">Obtaining APIs and Documentation</a>.</li> </ul>
App authentication (with app_basic)	<ul style="list-style-type: none"> <li>To enable app_basic authentication, ensure that the <b>app_basic</b> parameter has been set to <b>on</b> on the <a href="#">Configuration Parameters</a> tab page of the dedicated gateway.</li> <li>Add the header parameter <b>Authorization</b> and set the parameter value to "Basic + base64 (<i>appkey</i> + : + <i>appsecret</i>)", in which <i>appkey</i> and <i>appsecret</i> are the key and secret (or AppKey and AppSecret) obtained in <a href="#">Obtaining APIs and Documentation</a>.</li> </ul>
IAM authentication (with a token)	Obtain a token from the cloud platform and carry the token in API requests for authentication. For details, see section "Token Authentication" in the <i>API Gateway Developer Guide</i> .
IAM authentication (with AK/SK)	Use an SDK to sign API requests. For details, see section "AK/SK Authentication" in the <i>API Gateway Developer Guide</i> .
Custom authentication	Carry authentication information in API request parameters for authentication.
None	Call APIs without authentication.

----End

## 16.2 Response Headers

The following table describes the response headers that API Gateway adds to the response returned when an API is called.

**X-Apig-Mode: debug** indicates API debugging information.

Response Header	Description	Remarks
X-Request-Id	Request ID.	Returned for all valid requests.
X-Apig-Latency	Duration from the time when API Gateway receives a request to the time when the backend returns a message header.	Returned only when the request header contains <b>X-Apig-Mode: debug</b> .
X-Apig-Upstream-Latency	Duration from the time when API Gateway sends a request to the backend to the time when the backend returns a message header.	Returned only when the request header contains <b>X-Apig-Mode: debug</b> and the backend type is not Mock.
X-Apig-RateLimit-api	API request limit information. Example: <b>remain:9,limit:10,time:10 second</b> .	Returned only when the request header contains <b>X-Apig-Mode: debug</b> and a limit has been configured for the number of times the API can be called.
X-Apig-RateLimit-user	User request limit information. Example: <b>remain:9,limit:10,time:10 second</b> .	Returned only when the request header contains <b>X-Apig-Mode: debug</b> and a limit has been configured for the number of times the API can be called by a user.
X-Apig-RateLimit-app	App request limit information. Example: <b>remain:9,limit:10,time:10 second</b> .	Returned only when the request header contains <b>X-Apig-Mode: debug</b> and a limit has been configured for the number of times the API can be called by an app.
X-Apig-RateLimit-ip	IP address request limit information. Example: <b>remain:9,limit:10,time:10 second</b> .	Returned only when the request header contains <b>X-Apig-Mode: debug</b> and a limit has been configured for the number of times the API can be called by an IP address.
X-Apig-RateLimit-api-allenv	Default API request limit information. Example: <b>remain:199,limit:200,time:1 second</b> .	Returned only when the request header contains <b>X-Apig-Mode: debug</b> .

## 16.3 Error Codes

**Table 16-1** lists the error codes that you may encounter when calling APIs.

 **NOTE**

- For details about the error codes that may occur when you manage APIs, see section "Error Codes" in the *API Gateway API Reference*.
- If an error occurs when you use API Gateway, find the error message and description in the following table according to the error code, for example, APIG.0101. The error messages are subject to change without prior notice.

**Table 16-1** Error codes

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0101	The API does not exist or has not been published in the environment.	404	The API does not exist or has not been published in the environment.	Check whether the domain name, method, and path are consistent with those of the registered API. Check whether the API has been published. If it has been published in a non-production environment, check whether the X-Stage header in the request is the environment name. Check whether the domain name used to call the API has been bound to the group to which the API belongs.
APIG.0101	The API does not exist.	404	The API request method does not exist.	Check whether the API request method is the same as the method defined by the API.



Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0103	The backend does not exist.	500	The backend service was not found.	Contact technical support.
APIG.0104	The plug-ins do not exist.	500	No plug-in configurations were found.	Contact technical support.
APIG.0105	The backend configurations do not exist.	500	No backend configurations were found.	Contact technical support.
APIG.0106	Orchestration error.	400	An orchestration error occurred.	Check whether the frontend and backend parameters of the API are correct.
APIG.0201	API request error.	400	Invalid request parameters.	Set valid request parameters.
APIG.0201	Request entity too large.	413	The request body exceeds 12 MB.	Reduce the size of the request body.
APIG.0201	Request URI too large.	414	The request URI exceeds 32 KB.	Reduce the size of the request URI.
APIG.0201	Request headers too large.	494	The request headers are too large because one of them exceeds 32 KB or the total length exceeds 128 KB.	Reduce the size of the request headers.
APIG.0201	Backend unavailable.	502	The backend service is unavailable.	Check whether the backend address configured for the API is accessible.
APIG.0201	Backend timeout.	504	The backend service has timed out.	Increase the timeout duration of the backend service or shorten the processing time.

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0201	An unexpected error occurred	500	An internal error occurred.	Contact technical support.
APIG.0202	Backend unavailable	502	The backend is unavailable.	Check whether the backend request protocol configured for the API is the same as the request protocol used by the backend service.
APIG.0203	Backend timeout.	504	The backend service has timed out.	Increase the timeout of the backend service or shorten its processing time.
APIG.0204	SSL protocol is not supported: TLSv1.1	400	The SSL protocol version is not supported.	Use a supported SSL protocol version.
APIG.0301	Incorrect IAM authentication information.	401	The IAM authentication details are incorrect.	Check whether the token is correct.
APIG.0302	The IAM user is not authorized to access the API.	403	The IAM user is not allowed to access the API.	Check whether the user is controlled by a blacklist or whitelist.

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0303	Incorrect app authentication information.	401	The app authentication details are incorrect.	Check whether the request method, path, query strings, and request body are consistent with those used for signing; check whether the date and time on the client are correct; and check whether the signing code is correct by referring to section "Calling APIs Through App Authentication" in the <i>Developer Guide</i> .
APIG.0304	The app is not authorized to access the API.	403	The app is not allowed to access the API.	Check whether the app has been authorized to access the API.
APIG.0305	Incorrect authentication information.	401	The authentication information is incorrect.	Check whether the authentication information is correct.
APIG.0306	API access denied.	403	Access to the API is not allowed.	Check whether you have been authorized to access the API.
APIG.0307	The token must be updated.	401	The token needs to be updated.	Obtain a new token from IAM.

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0308	The throttling threshold has been reached.	429	The throttling threshold has been reached.	Try again after the throttling resumes. If the number of subdomain requests per day is reached, bind an independent domain name to the API.
APIG.0310	The project is unavailable.	403	The project is currently unavailable.	Select another project and try again.
APIG.0311	Incorrect debugging authentication information.	401	The debugging authentication details are incorrect.	Contact technical support.
APIG.0401	Unknown client IP address.	403	The client IP address cannot be identified.	Contact technical support.
APIG.0402	The IP address is not authorized to access the API.	403	The IP address is not allowed to access the API.	Check whether the IP address is controlled by a blacklist or whitelist.
APIG.0404	Access to the backend IP address has been denied.	403	The backend IP address cannot be accessed.	Check whether the backend IP address or the IP address corresponding to the backend domain name is accessible.
APIG.0502	The app has been frozen.	405	The app has been frozen.	Check whether your account balance is sufficient.
APIG.0601	Internal server error.	500	An internal error occurred.	Contact technical support.
APIG.0602	Bad request.	400	Invalid request.	Check whether the request is valid.

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0605	Domain name resolution failed.	500	Domain name resolution failed.	Check whether the domain name is correct and has been bound to a correct backend address.
APIG.0606	Failed to load the API configurations.	500	API configurations could not be loaded.	Contact technical support.
APIG.0607	The following protocol is supported: {xxx}	400	The protocol is not supported. Only xxx is supported. xxx is subject to the actual value in the response.	Use HTTP or HTTPS to access the API.
APIG.0608	Failed to obtain the admin token.	500	The tenant details cannot be obtained.	Contact technical support.
APIG.0609	The VPC backend does not exist.	500	The VPC backend service cannot be found.	Contact technical support.
APIG.0610	No backend available.	502	No backend services are available.	Check whether all backend services are available. For example, check whether the API calling information is consistent with the actual configuration.
APIG.0611	The backend port does not exist.	500	The backend port was not found.	Contact technical support.

Error Code	Error Message	HTTP Status Code	Description	Solution
APIG.0612	An API cannot call itself.	500	An API cannot call itself.	Modify the backend configurations, and ensure that the number of layers the API is recursively called does not exceed 10.
APIG.0613	The IAM service is currently unavailable.	503	IAM is currently unavailable.	Contact technical support.
APIG.0705	Backend signature calculation failed.	500	Backend signature calculation failed.	Contact technical support.
APIG.0802	The IAM user is forbidden in the currently selected region	403	The IAM user is disabled in the current region.	Contact technical support.
APIG.1009	AppKey or AppSecret is invalid	400	The AppKey or AppSecret is invalid.	Check whether the AppKey or AppSecret in the request is correct.

# 17 Permissions Management

## 17.1 Creating a User and Granting API Gateway Permissions

This topic describes how to use Identity and Access Management (IAM) to implement permissions control for your API Gateway resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing API Gateway resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or a cloud service to perform O&M on your API Gateway resources.

If your account does not require individual IAM users, skip this chapter.

This section describes the procedure for granting permissions (see [Figure 17-1](#)).

### Prerequisites

Learn about the permissions (see [Table 17-1](#)) supported by API Gateway and choose policies or roles according to your requirements. For the permissions of other services, see "System Permissions".

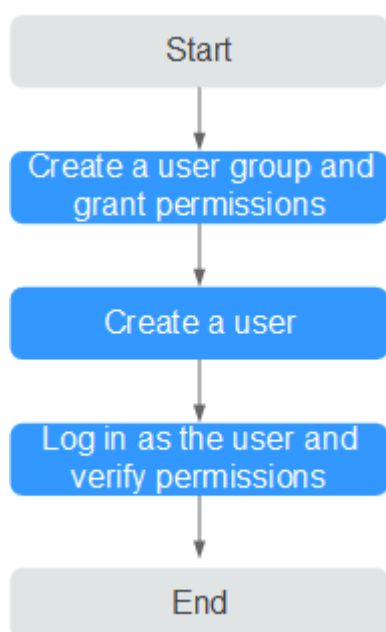
**Table 17-1** System-defined roles and policies supported by APIG

Role/ Policy Name	Description	Type	Dependency
APIG Administrator	Administrator permissions for APIG. Users granted these permissions can use all functions of <b>dedicated</b> gateways.	System-defined role	None

Role/ Policy Name	Description	Type	Dependency
APIG FullAccess	Full permissions for APIG. Users granted these permissions can use all functions of <b>dedicated</b> gateways.	System-defined policy	None
APIG ReadOnly Access	Read-only permissions for APIG. Users granted these permissions can only view <b>dedicated</b> gateways.	System-defined policy	None

## Process Flow

**Figure 17-1** Process for granting API Gateway permissions



1. Create a user group and assign permissions.  
Create a user group on the IAM console, and attach the **APIG Administrator** role or the **APIG FullAccess** policy to the group.
2. Create an IAM user.  
Create a user on the IAM console and add the user to the group created in **1**.
3. Log in and verify permissions.  
Log in to the API Gateway console as the created user, and verify that the user has administrator permissions for API Gateway.



## 17.2 API Gateway Custom Policies

Custom policies can be created to supplement the system-defined policies of API Gateway. For the actions that can be added to custom policies, see [Permissions Policies and Supported Actions](#).

You can create custom policies using one of the following methods:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see section "Creating a Custom Policy" in the *Identity and Access Management User Guide*. The following section contains examples of common API Gateway custom policies.

### NOTE

Only dedicated API gateways support system-defined policies and custom policies.

### Example Custom Policies

- Example 1: Allow users to create and debug APIs

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apig:apis:create",
        "apig:apis:debug"
      ]
    }
  ]
}
```

- Example 2: Deny API group creation

A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

The following method can be used if you need to assign permissions of the **APIG FullAccess** policy to a user but you want to prevent the user from creating API groups. Create a custom policy for denying API group creation, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on API gateways except creating API groups. The following is an example of a deny policy:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "apig:groups:create"
      ]
    }
  ]
}
```

```
] }  
}
```

# 18 Key Operations Recorded by CTS

## 18.1 API Gateway operations that can be recorded by CTS

### Enabling CTS

If you want to collect, record, or query operation logs for API Gateway in common scenarios such as security analysis, audit, and problem locating, enable Cloud Trace Service (CTS). For details, see section "Enabling CTS" in the *Cloud Trace Service User Guide*.

CTS provides the following functions:

- Recording audit logs
- Querying audit logs
- Dumping audit logs
- Encrypting trace files
- Enabling notifications of key operations

### Viewing Key Operations

With CTS, you can record operations associated with API Gateway for future query, audit, and backtracking.

**Table 18-1** API Gateway operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating an API group	ApiGroup	createApiGroup
Deleting an API group	ApiGroup	deleteApiGroup
Updating an API group	ApiGroup	updateApiGroup
Binding a domain name	ApiGroup	createDomainBinding

Operation	Resource Type	Trace Name
Change minimum TLS version	ApiGroup	modifySecureTransmission
Unbinding a domain name	ApiGroup	relieveDomainBinding
Adding a domain certificate	ApiGroup	addDomainCertificate
Deleting a domain certificate	ApiGroup	deleteDomainCertificate
Creating an API	Api	createApi
Deleting an API	Api	deleteApi
Deleting multiple APIs	Api	batchDeleteApi
Updating an API	Api	updateApi
Publishing an API	Api	publishApi
Taking an API offline	Api	offlineApi
Publishing multiple APIs or taking APIs offline	Api	batchPublishOrOfflineApi
Switching API versions	Api	switchApiVersion
Taking an API version offline	Api	offlineApiByVersion
Debugging an API	Api	debugApi
Creating an environment	Environment	createEnvironment
Deleting an environment	Environment	deleteEnvironment
Updating an environment	Environment	updateEnvironment
Creating an environment variable	EnvVariable	createEnvVariable
Updating an environment variable	EnvVariable	updateEnvVariable
Deleting an environment variable	EnvVariable	deleteEnvVariable
Creating an app	App	createApp
Deleting an app	App	deleteApp
Updating an application	App	updateApp
Resetting AppSecret	App	resetAppSecret

Operation	Resource Type	Trace Name
Binding a client to an API	AppAuth	grantAuth
Unbinding a client from an API	AppAuth	relieveAuth
Creating a signature key	Signature	createSignature
Deleting a signature key	Signature	deleteSignature
Updating a signature key	Signature	updateSignature
Binding a signature key	SignatureBinding	createSignatureBinding
Unbinding a signature key	SignatureBinding	relieveSignatureBinding
Creating an access control policy	Acl	createAcl
Deleting an access control policy	Acl	deleteAcl
Deleting access control policies	Acl	batchDeleteAcl
Updating an access control policy	Acl	updateAcl
Creating an access control blacklist	Acl	addAclValue
Deleting an access control blacklist	Acl	deleteAclValue
Binding an access control policy to an API	AclBinding	createAclBinding
Unbinding an access control policy from an API	AclBinding	relieveAclBinding
Unbinding multiple access control policies from APIs	AclBinding	batchRelieveAclBinding
Creating a request throttling policy	Throttle	createThrottle
Deleting a request throttling policy	Throttle	deleteThrottle
Deleting multiple request throttling policies	Throttle	batchDeleteThrottle

Operation	Resource Type	Trace Name
Updating a requesting throttling policy	Throttle	updateThrottle
Binding a request throttling policy	ThrottleBinding	createThrottleBinding
Unbinding a request throttling policy	ThrottleBinding	relieveThrottleBinding
Unbinding multiple request throttling policies	ThrottleBinding	batchRelieveThrottleBinding
Creating an excluded request throttling configuration	ThrottleSpecial	createSpecialThrottle
Deleting an excluded request throttling configuration	ThrottleSpecial	deleteSpecialThrottle
Updating an excluded request throttling configuration	ThrottleSpecial	updateSpecialThrottle
Creating a load balance channel	Vpc	createVpc
Deleting a load balance channel	Vpc	deleteVpc
Updating a load balance channel	Vpc	updateVpc
Adding members to a load balance channel	Vpc	addVpcMember
Deleting members from a load balance channel	Vpc	deleteVpcMember
Exporting an API	Swagger	swaggerExportApi
Exporting multiple APIs	Swagger	swaggerExportApiList
Exporting all APIs in a group	Swagger	swaggerExportApi-ByGroup
Importing APIs to a new group	Swagger	swaggerImportApiTo-NewGroup
Importing APIs to an existing group	Swagger	swaggerImportApiTo-ExistGroup
Exporting all custom backends	Swagger	SwaggerExportLdApi

Operation	Resource Type	Trace Name
Importing custom backends	Swagger	SwaggerImportLdApi
Creating a custom authorizer	Authorizer	createAuthorizer
Deleting a custom authorizer	Authorizer	deleteAuthorizer
Updating a custom authorizer	Authorizer	updateAuthorizer
Creating a plug-in	Plugin	createPlugin
Updating a plug-in	Plugin	updatePlugin
Deleting a plug-in	Plugin	deletePlugin
Binding a plug-in to an API	Plugin	pluginAttachApi
Unbinding an API from a plug-in	Plugin	pluginDetachApi
Binding a plug-in to an API	Plugin	apiAttachPlugin
Unbinding a plug-in from an API	Plugin	apiDetachPlugin

## Disabling CTS

Disable CTS by following the procedure in section "Deleting a Tracker" in the *Cloud Trace Service User Guide*.

## 18.2 Querying Audit Logs

Query audit logs by following the procedure in section "Querying Real-Time Traces" in the *Cloud Trace Service User Guide*.

**Figure 18-1** Viewing logs

