API Gateway

Plugin Usage

Product Documentation

Contents

Plugin Usage Overview

Last updated : 2021-05-14 19:11:10

Plugin Overview

Plugins are advanced feature configurations provided by API Gateway. You can create configuration items such as IP access control through plugins and then bind the plugins to APIs for them to take effect.

Plugins have the following advantages over traditional configuration items:

- Feature configuration is decoupled from API configuration. One plugin can be bound to multiple APIs under different services.
- Hot update is supported for configurations. After a plugin is bound to an API, it can take effect without publishing the service.

Directions

Step 1. Create a plugin

1. Log in to the API Gateway console.
2. On the left sidebar, click **Plugin** to enter the plugin list page.
3. Click **Create** in the top-left corner to create a plugin.

Step 2. Bind an API and make the plugin effective

1. Select the just created plugin in the list and click **Bind API** in the **Operation** column.

2. In the **Bind API** pop-up window, select the service, environment, and the API that needs to be bound to the plugin.

3. Click **OK** to bind the plugin to the API. At this time, the configuration of the plugin has taken effect for the API.

Step 3. View the plugins bound to an API

1. On the left sidebar, click **Service** to enter the service list page.

2. In the service list, click the name of the target service to view it.
3. In the API list, click the name of the target API to enter the API details page.
4. On the API details page, click the **Bound Plugin** tab to view the information of plugins bound to the target API.

Supported Plugin Type

- IP access control

> **Note :**
>
> API Gateway currently only supports the IP access control plugin and will offer more plugins such as traffic throttling, circuit breaker, and parameter conversion in the future.

Plugin Rules

- An API can be bound to only one plugin of the same type.
- Plugins are region-specific and can be bound to APIs only in the same region. Cross-region binding is not supported.
- APIs can be bound to plugins only after they are published in the corresponding environment. Unpublished APIs cannot be bound.
- The deactivation of APIs does not affect their binding relationships with plugins, and the plugins will still take effect after the APIs are republished in the environment.
- Plugins support hot updates, and all binding and unbinding operations can take effect without republishing the service.
- After APIs are deleted, their binding relationships with plugins will also be deleted.

IP Access Control

Last updated : 2021-05-14 19:11:11

Overview

IP access control is a security protection capability provided by API Gateway. It is mainly used to restrict the source IPs of API callers. You can allow/reject API requests from a certain source by configuring the IP allowlist/blocklist of an API.

Note :

The original IP access control policy data has been migrated to the IP access control plugin, which can be managed on the Plugin page.

Directions

Step 1. Create a plugin

1. Log in to the API Gateway console.
2. On the left sidebar, click **Plugin** to enter the plugin list page.
3. Click **Create** in the top-left corner to create an IP access control plugin.

Step 2. Bind an API and make the plugin effective

1. Select the just created plugin in the list and click **Bind API** in the **Operation** column.
2. In the **Bind API** pop-up window, select the service, environment, and the API that needs to be bound to the plugin.
3. Click **OK** to bind the plugin to the API. At this time, the configuration of the plugin has taken effect for the API.

Notes

- The IP access control plugin supports blocklist and allowlist modes. When the allowlist is used, requests from IPs not in the allowlist will be rejected by API Gateway; when the blocklist is used,

requests from IPs in the blocklist will be rejected by API Gateway.

- Multiple IPs or CIDR blocks can be entered in the IP access control plugin, which should be separated with semicolons.

Basic Traffic Throttling

Last updated : 2021-06-04 17:59:46

Overview

Basic traffic throttling plugin is a powerful traffic throttling component provided by API Gateway. It can throttle traffic in three dimensions of API, application, and client IP by second, minute, hour, or day. You can create a basic traffic throttling plugin and bind it to your API to take effect and protect your backend services.

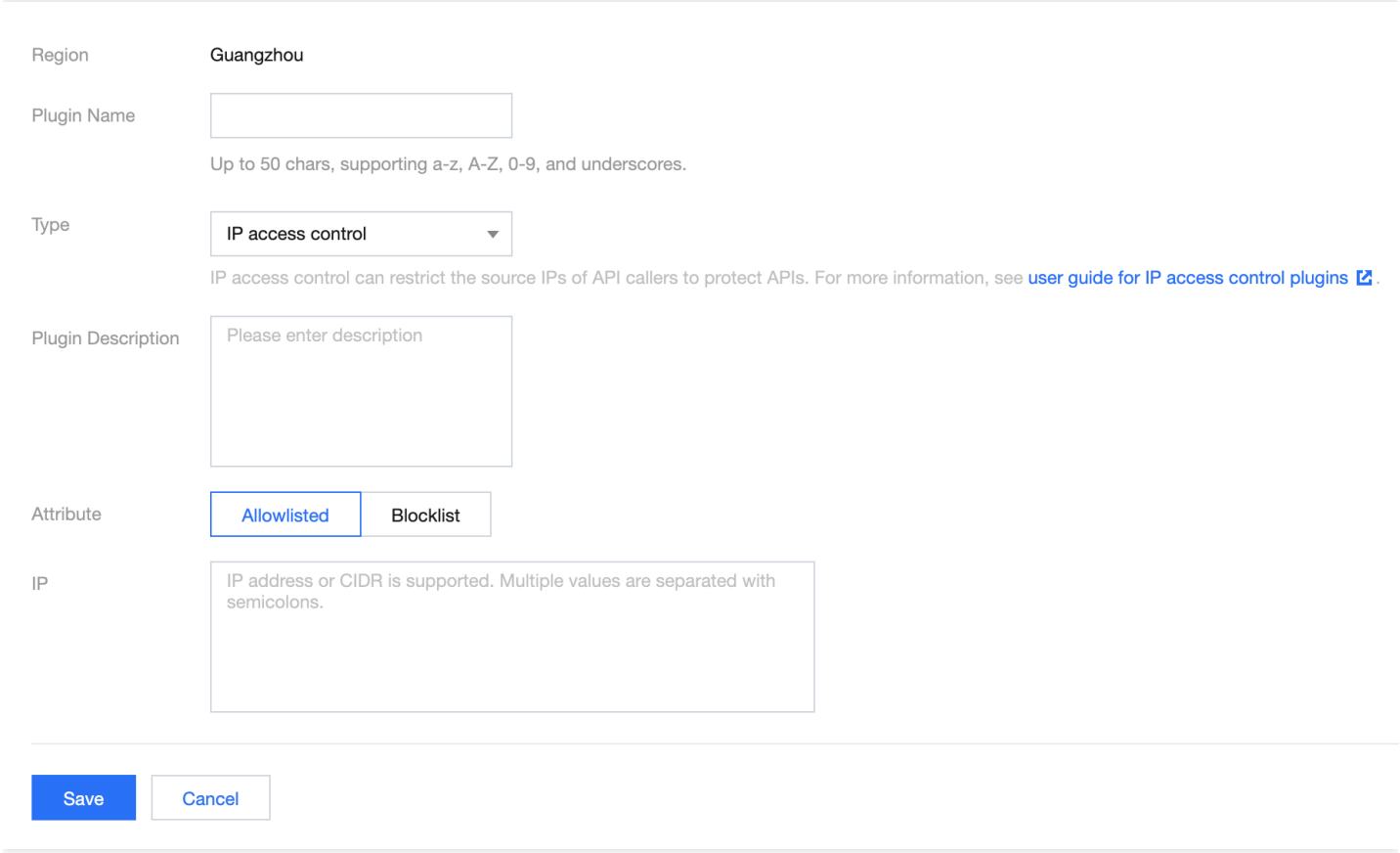
Directions

Step 1. Create a plugin

1. Log in to the API Gateway console.
2. On the left sidebar, click **Plugin** to enter the plugin list page.
3. Click **Create** in the top-left corner of the page and select **Basic Throttling** as the plugin type to create a basic traffic throttling plugin.

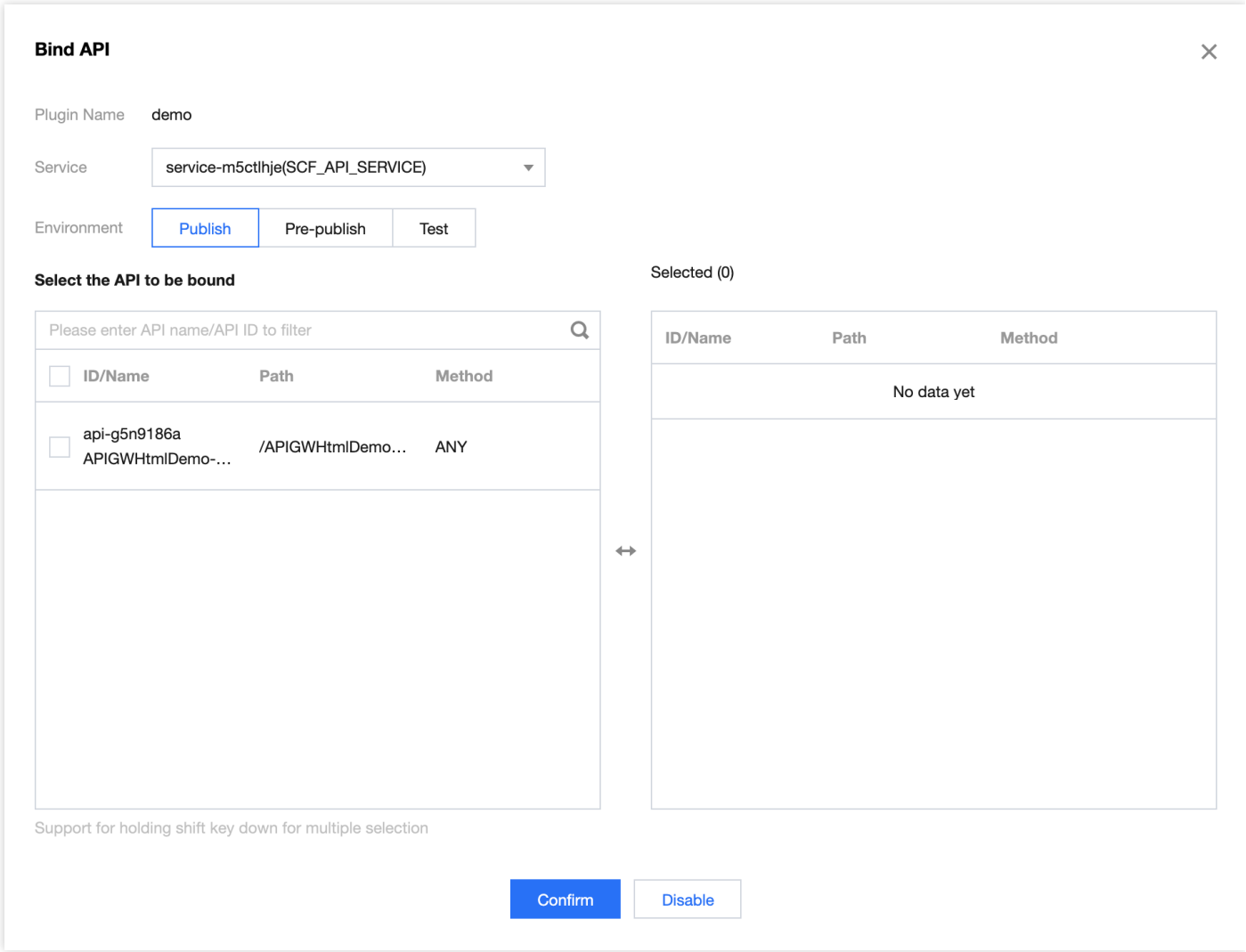
Parameter	Required	Description
Duration	Yes	The duration of traffic throttling, which supports four dimensions: second, minute, hour, and day. It is used in conjunction with "traffic throttling threshold" to indicate the upper limit of the number of requests per unit time.
API Threshold	Yes	The upper limit of the number of times an API can be accessed within a certain period of time.
Application Threshold	No	The upper limit of the number of times an application can be accessed within a certain period of time, which takes effect for all applications bound to this API.
Client IP Threshold	No	The upper limit of the number of times a client IP can be accessed within a certain period of time, which takes effect for all client IPs bound to this API.

Parameter	Required	Description
Special Application	No	Up to 30 items can be entered. For such applications, the basic API traffic throttling of the traffic throttling policy still takes effect, but you need to set an additional traffic throttling threshold for them. Meanwhile, the basic application traffic throttling and user traffic throttling of the traffic throttling policy will stop working for such applications.
Special Client IP	No	Up to 30 items can be entered. For such IPs, the basic API traffic throttling of the traffic throttling policy still takes effect, but you need to set an additional traffic throttling threshold for them. Meanwhile, the basic application traffic throttling and client IP traffic throttling of the traffic throttling policy will stop working for such IPs.



Step 2. Bind an API and make the plugin effective

1. Select the just created plugin in the list and click **Bind API** in the **Operation** column.
2. In the **Bind API** pop-up window, select the service, environment, and the API that needs to be bound to the plugin.



3. Click **OK** to bind the plugin to the API. At this time, the configuration of the plugin has taken effect for the API.

Notes

The basic traffic throttling plugin will be affected by service traffic throttling and API traffic throttling. If multiple traffic throttling rules take effect at the same time, the lowest threshold will prevail.
For example, if the traffic throttling threshold of an API is set to 500 QPS in the basic traffic throttling plugin, the traffic throttling threshold of the service to which the API belongs is 100 QPS, and the traffic throttling threshold of the API itself is 50 QPS, then the actually effective threshold is 50 QPS.

CORS

Last updated : 2021-08-17 15:34:18

Overview

Cross-Origin Resource Sharing (CORS) is a W3C standard. It allows web application servers to perform cross-origin access control, so that cross-origin data transfer can be conducted securely. Currently, API Gateway supports configuring CORS rules to allow or deny corresponding cross-origin requests as needed.

If the default CORS configuration of API Gateway cannot meet your needs, you can configure custom complex CORS rules through the CORS plugin and bind them to APIs for taking effect.

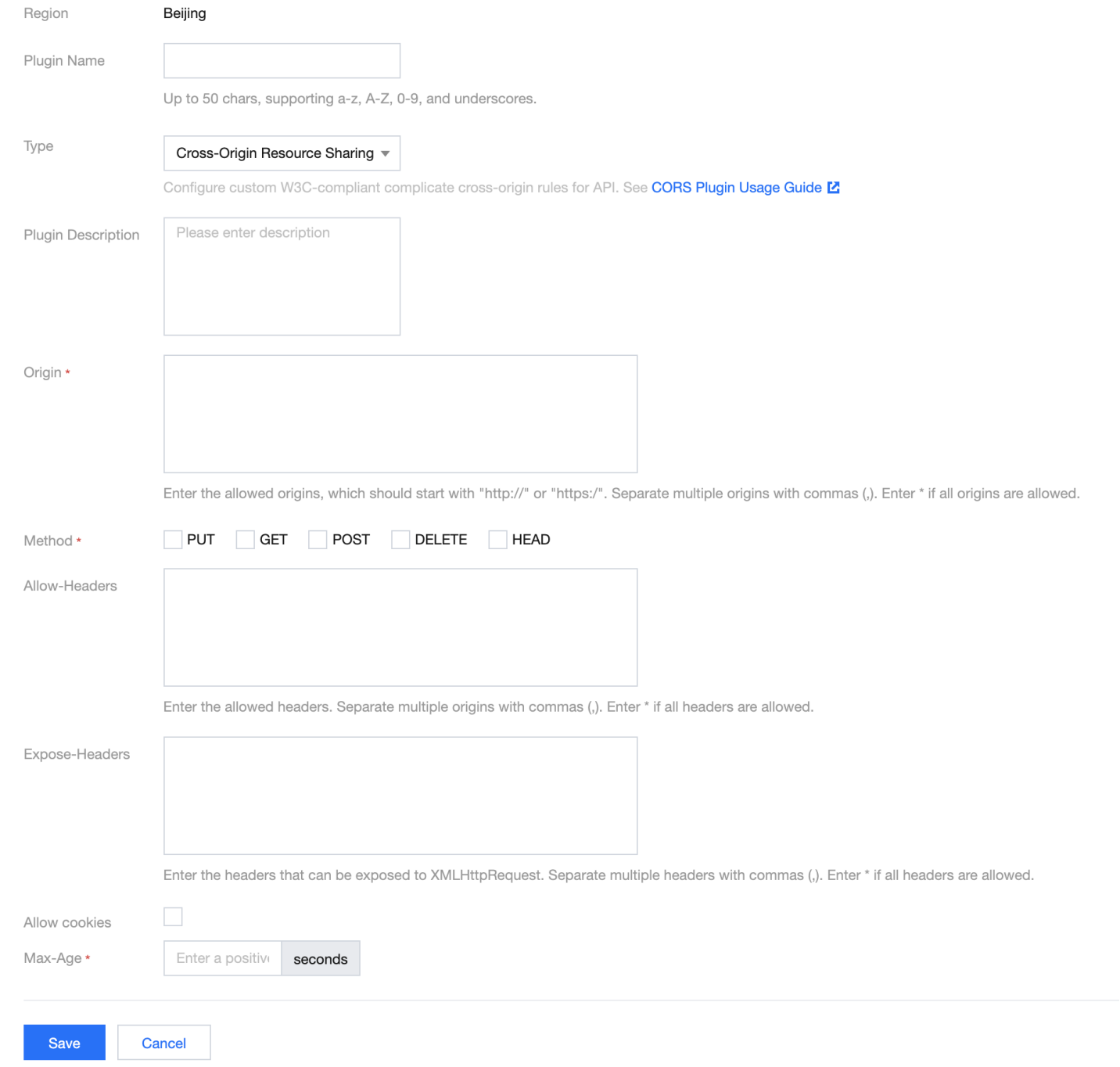
Directions

Step 1. Create a plugin

1. Log in to the API Gateway console.
2. On the left sidebar, click **Plugin** to enter the plugin list page.
3. Click **Create** in the top-left corner of the page and select **Cross-Origin Resource Sharing (CORS)** as the plugin type to create a CORS plugin.

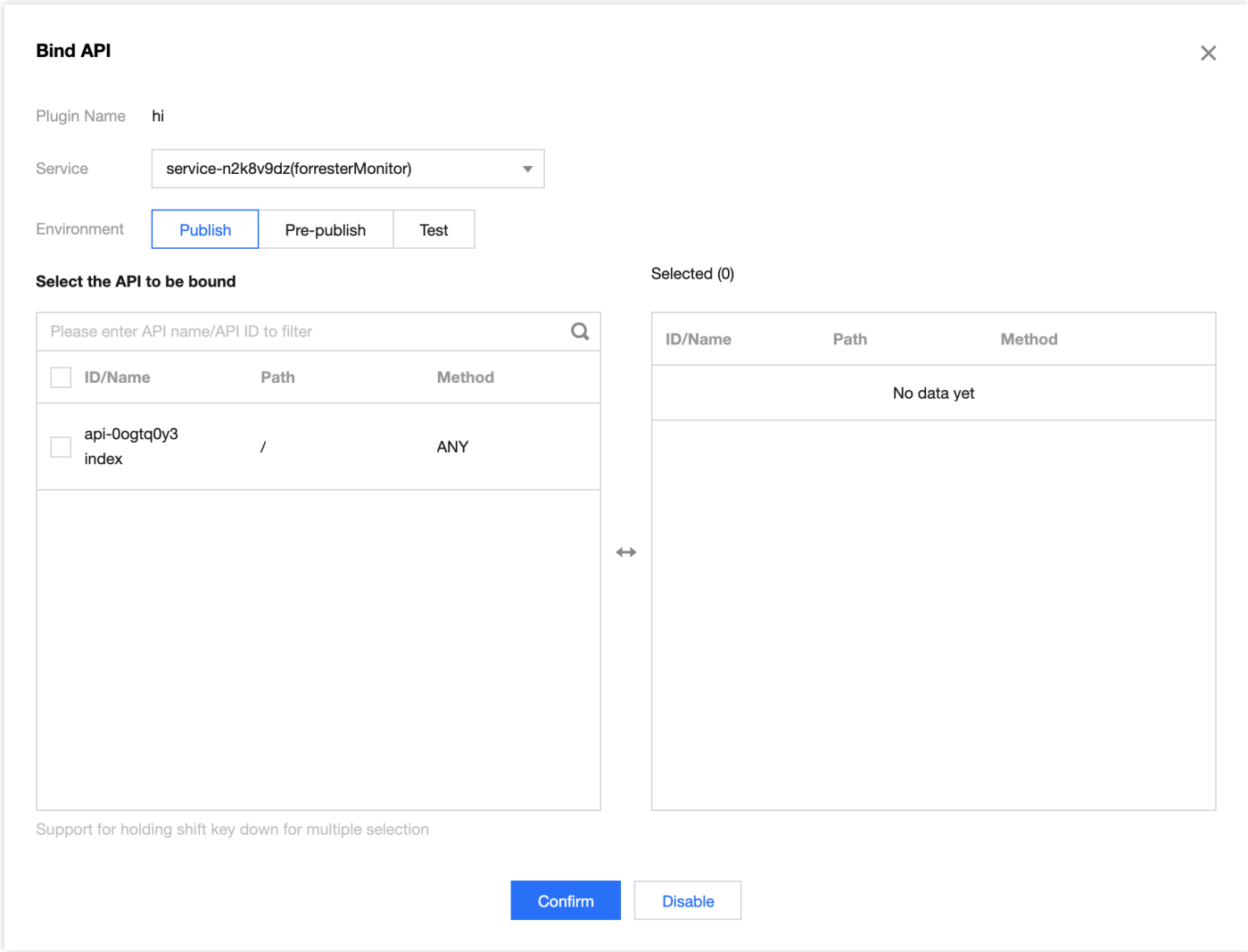
Parameter	Required	Description
Origin	Yes	Specify the origins of allowed cross-origin requests; You can specify multiple origins and separate them by commas; You can configure `*`, which means that all domain names are allowed; Be careful not to omit the protocol name `http` or `https`. If the port is not the default port 80, you also need to include the port.
Method	Yes	GET, PUT, POST, DELETE, and HEAD methods are supported. You can enumerate one or more allowed CORS request methods.
Allow-Headers	No	Specify the custom HTTP request headers that can be used for subsequent `OPTIONS` requests; You can specify multiple headers and separate them by commas; You can configure `*`, which means that all header are allowed; If you leave this parameter empty, all headers will be denied.

Parameter	Required	Description
Expose-Headers	No	Specify the headers that can be exposed to the `XMLHttpRequest` object; You can specify multiple headers and separate them by commas; You can configure `*`, which means that all header are allowed; If you leave this parameter empty, all headers will be denied.
Allow Cookies	No	Specify whether to allow cookies.
Max-Age	Yes	Set the validity period of the result obtained by `OPTIONS`. The value must be a positive integer, such as 600.



Step 2. Bind an API and make the plugin effective

1. Select the just created plugin in the list and click **Bind API** in the **Operation** column.
2. In the **Bind API** pop-up window, select the service, environment, and the API that needs to be bound to the plugin.



3. Click **OK** to bind the plugin to the API. At this time, the configuration of the plugin has taken effect for the API.

Notes

Currently, there are two places in API Gateway where you can set CORS rules:

- Create API > frontend configuration > CORS is supported: enable the **CORS is supported** configuration item when creating an API, and API Gateway will add `Access-Control-Allow-Origin : *` in the response header by default.
- The CORS plugin as described in this document. For more information, see Directions.

The CORS plugin has a higher priority than the **CORS is supported** configuration item. When the former is bound to an API, the latter of the API will not take effect.