

API Gateway

Operation Guide

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

Instance Management

Instance Parameter Configuration

Service Management

Creating Services

Editing Services

Deleting Services

Service Domain Name

Binding Service to VPC

Migrating Service from Shared Instance to Dedicated Instance

Creating API

API Creation Overview

Creating APIs Connecting to the Public URL/IP Backend

Creating APIs Connecting to the VPC Resource Backend

Creating APIs Connecting to the SCF Backend

Creating API for Interconnecting Backend with COS

Creating APIs Connecting to the Mock Backend

Creating APIs Connecting to the TSF Backend

Importing APIs

API Management

Debugging General APIs

Debugging Microservice APIs

Compressing Responses

Base64 Encoding

Calling API

Calling Key Pair Authentication API

Authentication-Free API

Release and Access

Overview

Service Release and Deactivation

Service Access

Environment Version Switch

Custom Domain Name and Certificate

Configuring Custom Domain Name

Usage Plan

- Overview

- Working with a Usage Plan

- Traffic Control

- Backend Upstream

- VPC Upstream

- Verification and Security

- Overview

- Application-Enabled Authentication Method

- Authentication-Free Mode

- OAuth2.0

- CAM Policy

- Log Statistics

- View Log Analysis

- Exporting Service Logs

- Viewing Operation Logs

- Shipping to CLS

- Access Monitoring

- Viewing Monitoring Charts

- Viewing API Statistics

- Monitoring Metrics

- API Doc Generator

- Application Management

- Permission Management

- Precautions

- Private IP Ranges and Public VIPs of API Gateway Regions

Operation Guide

Instance Management

Instance Parameter Configuration

Last updated : 2024-01-24 17:10:10

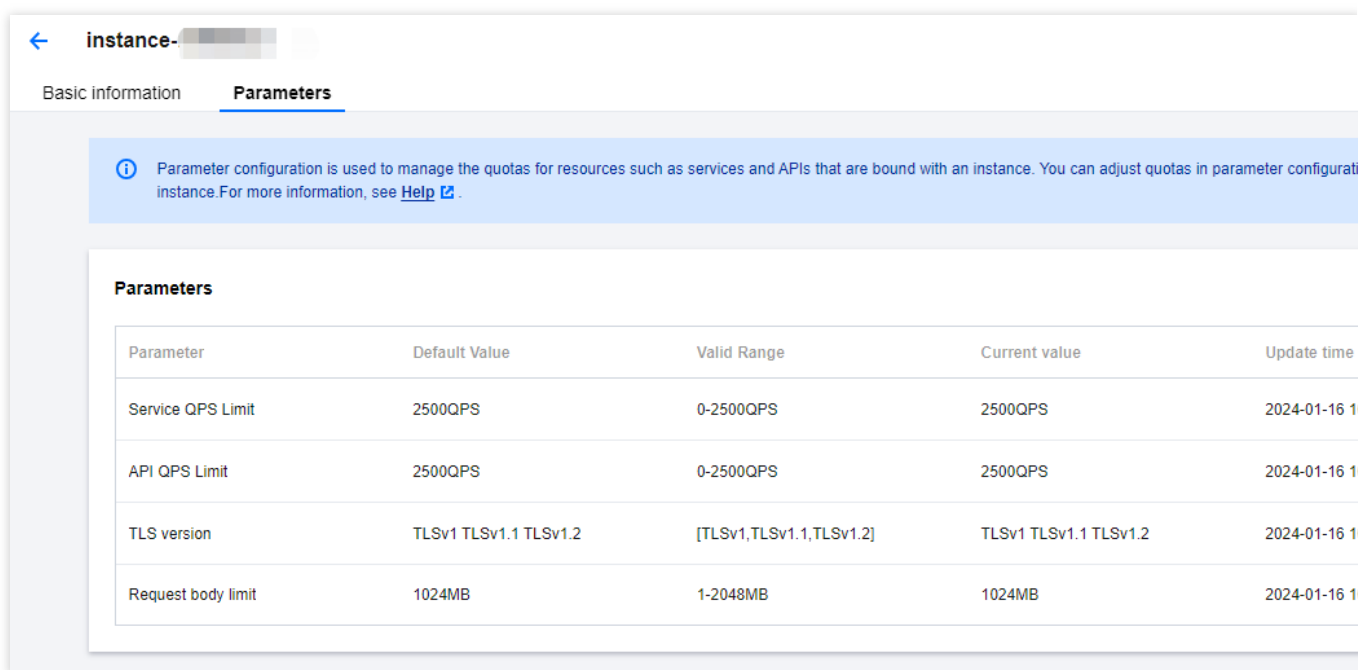
Operation scenarios

This guide instructs you on how to specifically configure parameters for a dedicated Tencent Cloud **API Gateway(apigateway)** instance.

Prerequisites

You have created a dedicated instance.

In the instance list, click the instance ID to enter the instance detail page and switch to **Parameters**.



Parameter	Default Value	Valid Range	Current value	Update time
Service QPS Limit	2500QPS	0-2500QPS	2500QPS	2024-01-16 1
API QPS Limit	2500QPS	0-2500QPS	2500QPS	2024-01-16 1
TLS version	TLSv1 TLSv1.1 TLSv1.2	[TLSv1,TLSv1.1,TLSv1.2]	TLSv1 TLSv1.1 TLSv1.2	2024-01-16 1
Request body limit	1024MB	1-2048MB	1024MB	2024-01-16 1

Note:

The shared version does not support the above adjustments. If necessary, please switch to the dedicated version.

See [Instance Specification](#).

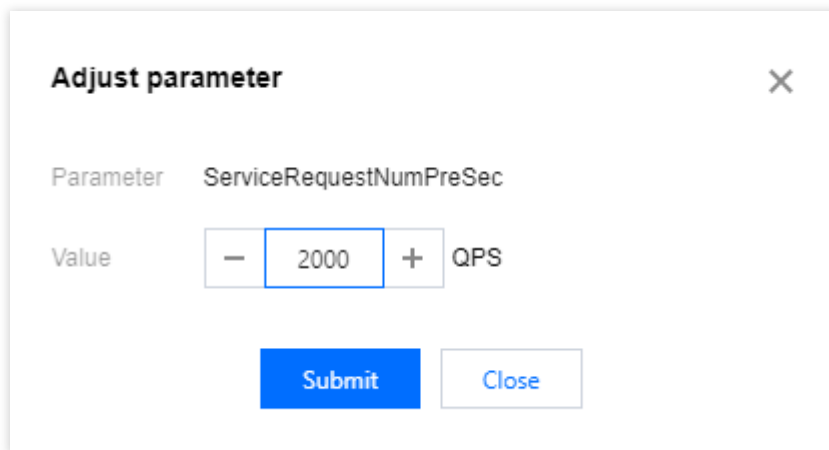
Configure Service QPS

Definition

The QPS of services can be configured uniformly in the instance dimension. For each service's QPS, throttle as needed.

Steps

1. Log into the [API Gateway console](#), click **Instance** to enter the [Instance List](#).
2. Click the instance ID to enter the details page;
3. Click Parameter Configuration List > **Service QPS Limit**, click **Adjust**. A pop-up dialogue box allows you to set uniformly.
4. Once configured, the QPS of each service in this instance adheres to this upper limit, can only be reduced, increment is not feasible. Take an adjustment to 2000 as an example.



The image shows a dialog box titled "Adjust parameter" with a close button (X) in the top right corner. Inside the dialog, there is a table with two rows. The first row has a "Parameter" column with the value "ServiceRequestNumPreSec". The second row has a "Value" column with a numeric input field containing "2000", flanked by minus and plus buttons, and a unit "QPS". At the bottom of the dialog, there are two buttons: "Submit" (in blue) and "Close" (in light blue).

Parameter	Value
ServiceRequestNumPreSec	<div>− <input type="text" value="2000"/> + QPS</div>

Submit Close

5. To view the maximum limit of each service in this instance, click on **Basic Information > View Service**.

← instance- [redacted]

Basic information Parameters

Instance information [View services](#) [Edit](#)

Instance name [redacted]

Instance ID instance-[redacted]

Instance status **Running**

Description N/A

Region Nanjing

Availability Zone Nanjing Zone 2

Tag [redacted]

Log shipping [Manage](#)

Status Not enabled

Billing details

Billing mode Pay as you go

Edition Basic

Creation time 2024-01-16 10:41:25

Network configuration

Network [redacted]

Subnet [redacted]

Public network entry address [redacted]

Private network entry address [redacted]

Egress bandwidth 5000Mbps

6. The page will be redirected to the service list. Click an ID of a service to navigate to the detail page.

Basic information Data query

ⓘ In compliance with relevant national policies and regulations, Tencent Cloud's API Gateway services, created after March 1, 2024, will no longer support public internet access through their

ⓘ Important announcement: Tencent Cloud API Gateway plans to upgrade shared instances in different regions during different periods. During the upgrade, your service can still be accessed will change. For details, see [Private IP Ranges and Public VIPs of API Gateway Regions](#)

[Create](#) [Import APIs](#) [Export APIs](#) [Delete](#) Instance: instance-[redacted] Please

<input type="checkbox"/>	Service name	Service status	Mo...	Network type	Published to	Instance type	Instance ID	Tag	Creation time
1 result found for "Instance: [redacted]" Back to list									
<input type="checkbox"/>	service-[redacted]	Running	[redacted]	Public network	N/A	Dedicated	[redacted]	[redacted]	2024-01-16 16:3...

Total items: 1 20

7. In the Detail Page > **Basic Configuration** > **Service-based throttling**, the traffic control upper limit value here correlates with the value that sets in [Step 4](#) above.

The screenshot displays the 'Basic configurations' tab for a service in the Tencent Cloud API Gateway console. The service is named 'service-...' and is of type 'Dedicated'. The 'Basic information' section shows the service name, ID, region (Guangzhou), published status (N/A), created APIs (0/100), instance, and description. The 'Network' section shows the frontend type (HTTP & HTTPS), public address, IP version (IPv4), and VPC. The 'Billing details' section shows the instance type (Dedicated), billable item (Instance fees + traffic fees), and creation time (2024-01-16 16:33:26). The 'Service-based throttling' section, highlighted with a red box, shows the release to, pre-publish, and test environment QPS limits, all set to 2000 QPS (Max: 2000 QPS). A note at the bottom states: 'Service traffic throttling is effective for all APIs under the service. You can go to the API details page to adjust the throttling policy of a single API. Please use basic traffic throttling plugins for throttling by API/application/ClientIP with a time granularity at the second/minute/hour/day level. For details, see [User Guide](#).'

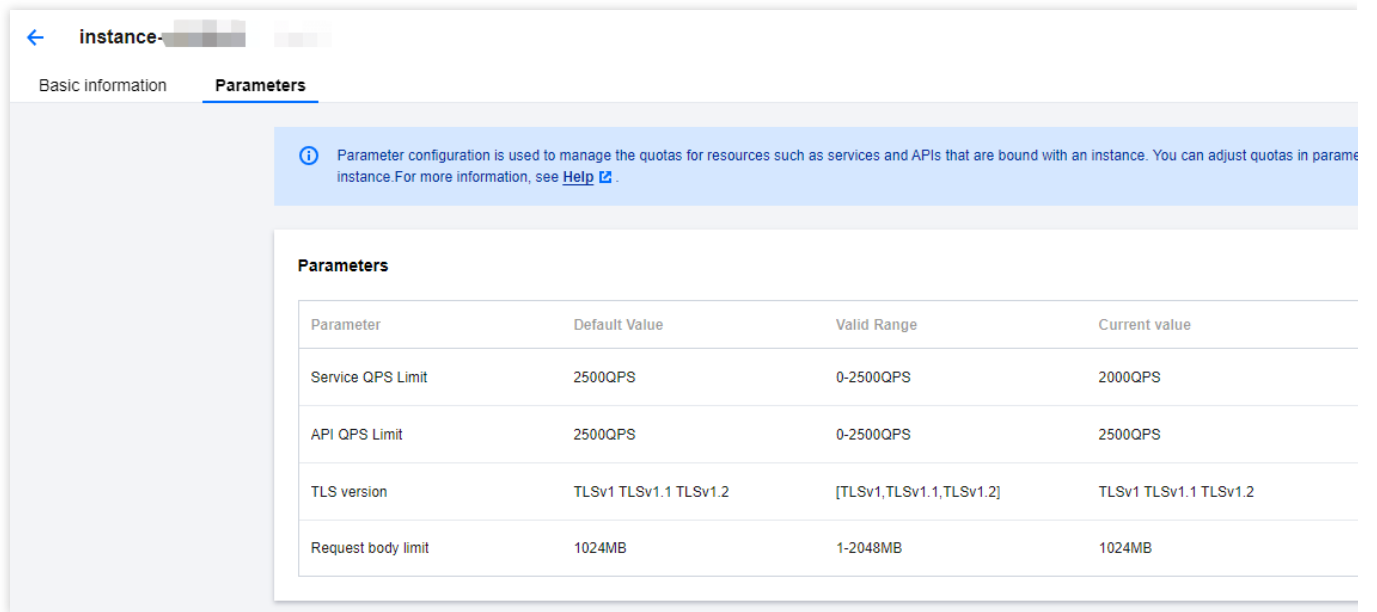
API QPS Configuration

Definition

The QPS for APIs can be set uniformly at the instance level, tailoring the flow control of each API for each service as needed.

Steps

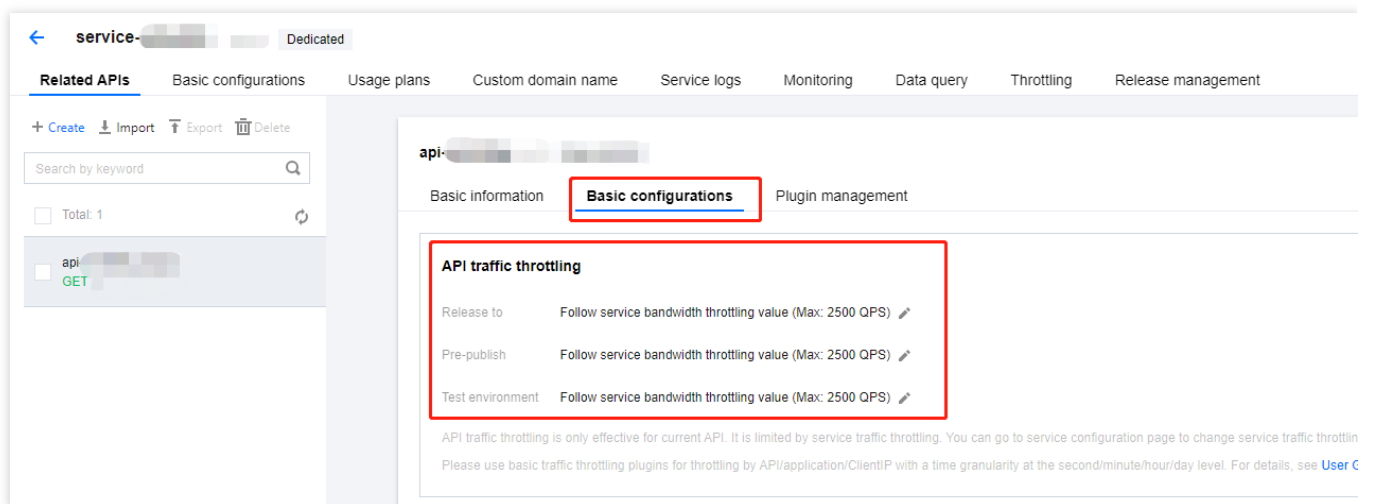
1. On the instance details page within the parameter configuration, it is feasible to uniformly modify the traffic control limit for all APIs under this instance.



In addition, if there is a need to individually set traffic control for each API, please refer to the following steps.

2. Following the steps to configure the QPS for the service, continue by creating an API within this service, please refer to [API Creation Overview](#);

3. On the service detail page, select this API. The details of this API will be displayed on the right. Switch to **Basic configurations**, where you can view the API's flow control limit;



4. By default, the flow control value of each API follows the service flow control value. If individual settings are required, click the edit icon to display the following prompt box, and modify as needed.

Set Call Limit ✕

Bandwidth throttling

Follow service bandwidth throttling value

Max request limit

Throttling limit

2000

QPS

Range: 0-2500

Submit

Close

TLS Version Configuration

The API Gateway instance provides a control feature of TLS version. You can flexibly configure the TLS protocol version according to the needs of the services in different instances. In this chapter, you can understand the concepts of TLS, its use cases, and version configuration methods.

Concept

Transport Layer Security (TLS), a secure transport layer protocol, provides confidentiality and data integrity between two applications, with the most typical application being HTTPS. HTTPS, or HTTP over TLS, constitutes a more secure form of HTTP. It operates below the HTTP layer and above the TCP layer, offering data encryption and decryption services for the HTTP layer.

Protocol version	Description	Supported mainstream browsers	Does the API Gateway support adjustments?
TLSv1.0	RFC2246, released in 1999, is based on SSLv3.0. This version is vulnerable to various attacks such as BEAST and POODLE. Additionally, it supports weaker encryption, which is no longer effective for contemporary network connection security. It does not adhere to the PCI DSS compliance determination standards.	IE6+ Chrome 1+ Firefox 2+	Yes
TLSv1.1	RFC4346, released in 2006, corrects several	IE 11+	Yes

	issues with TLSv1.0.	Chrome 22+ Firefox 24+ Safari 7+	
TLSv1.2	RFC5246 was published in 2008 and is currently the widely used version.	IE 11+ Chrome 30+ Firefox 27+ Safari 7+	Yes
TLSv1.3	RFC8446 is the most recent version of TLS, released in 2018. It supports the 0-RTT mode (faster) and only fully forward secure key exchange algorithms (more secure).	Chrome 70+ Firefox 63+	No (in planning, coming soon)

Steps

1. In Instance Details page > **Parameters**, you can uniformly edit the TLS protocol versions that require support for this instance;

← instance- [redacted]

Basic information **Parameters**

Parameter configuration is used to manage the quotas for resources such as services and APIs that are bound with an instance. You can adjust quotas in parameter configuration. For more information, see [Help](#).

Parameters

Parameter	Default Value	Valid Range	Current value
Service QPS Limit	2500QPS	0-2500QPS	2000QPS
API QPS Limit	2500QPS	0-2500QPS	2500QPS
TLS version	TLSv1 TLSv1.1 TLSv1.2	[TLSv1,TLSv1.1,TLSv1.2]	TLSv1 TLSv1.1 TLSv1.2
Request body limit	1024MB	1-2048MB	1024MB

2. Click **Adjust** and select the required version in the dialog box.

Adjust parameter ✕

Parameter

TlsVersion

Value

☒ TLSv1

☒ TLSv1.1

☒ TLSv1.2

Submit

Close

Configure Request Body Size

1. In Instance Details page > **Parameters**, you can uniformly edit the TLS protocol versions that require support for this instance;

← instance-

Basic information

Parameters

ⓘ

Parameter configuration is used to manage the quotas for resources such as services and APIs that are bound with an instance. You can adjust quotas in parameter configuration. For more information, see [Help](#).

Parameters

Parameter	Default Value	Valid Range	Current value
Service QPS Limit	2500QPS	0-2500QPS	2000QPS
API QPS Limit	2500QPS	0-2500QPS	2500QPS
TLS version	TLSv1 TLSv1.1 TLSv1.2	[TLSv1,TLSv1.1,TLSv1.2]	TLSv1 TLSv1.1 TLSv1.2
Request body limit	1024MB	1-2048MB	1024MB

2. Click **Adjust** and input the required upper limit value in the dialog box.

Adjust parameter ×

Parameter

MaxRequestBodySize

Value

−

1024

+

MB

Submit

Close

Service Management

Creating Services

Last updated : 2023-12-22 09:46:03

Scenarios

To implement a feature, we usually need to use a group of APIs. In API Gateway, you can add related APIs to a service for easy and efficient management.

Directions

1. Log in to the [API Gateway console](#).
2. Click **Service** on the left sidebar.
3. Under the current region, click **Create** at the top-left corner to create a service.

Note:

Up to 50 services per region.

If the current region is not the one you need, switch the region first. Note that the following regions do not support public IP access: East China (Shanghai Finance

4. Enter the service name and description, and select a frontend type.

Create Service

RegionGuangzhou

Service Name

Up to 50 chars, supporting a-z, A-Z, 0-9, and underscores.

Frontend Typehttp

Billing ModePay as you go

Remarks

Please enter remarks

Fees

Call fee: (tiered pricing)

Traffic fee:

Submit

Close

Note:

Fontend type: The supported protocol. It supports HTTP and HTTPS.

Access type: Public network access, private network VPC access, or both can be selected. API Gateway will generate different domain names accordingly. It's only available to beta users.

Access scope of a private network VPC domain name: CVM instances in VPCs in the same region as the service.

Private network VPC access is in now. To try it out, contact your sales rep or [submit a ticket](#).

The service name can contain up to 50 characters ([a-z], [A-Z], [0-9], and [_]).

5. Click **Submit** to complete the creation.

6. You can click the service name to enter the service details page and create APIs.

For more information, please see [API Creation Overview](#).

Service Name	Service Status	Monitor	Default Domain Name ①	Frontend Type	Release Environment and Status
	Running		Public Network:	http	Test: Not publish Pre-publish: Not publish Publish: Not publish

Editing Services

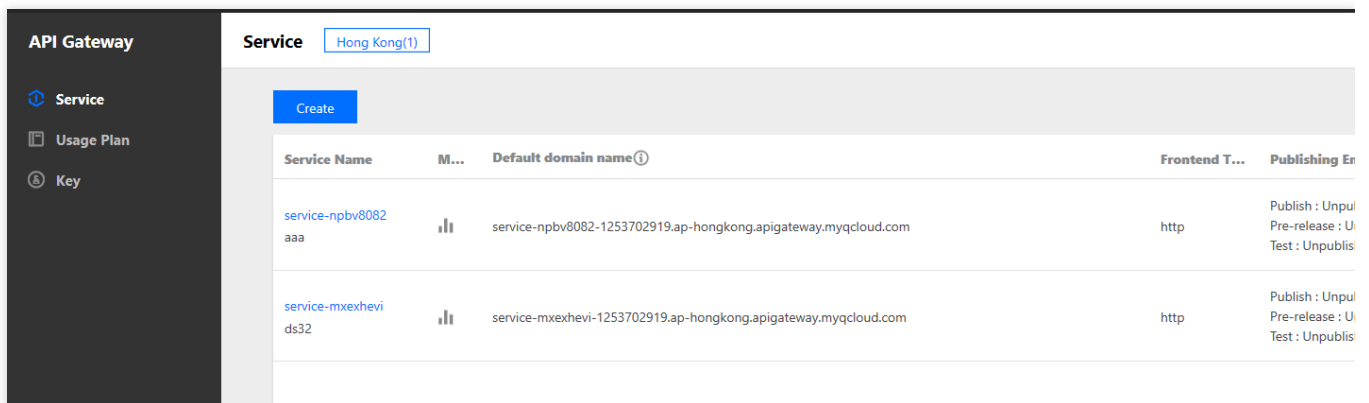
Last updated : 2023-12-22 09:46:12

Operation Scenarios

This document describes how to edit a created service in the API Gateway Console.

Directions

1. Log in to the [API Gateway Console](#) and select **Service** on the left sidebar.
2. Select the service to be edited from the service list and click **Edit** in the "Operation" column.



3. Enter the edited content and click **Submit**.

Edit services✕

Service Name

aaa

Up to 50 characters (including a-z, A-Z, 0-9, and _)

Region

Hong Kong

Frontend Type

http

Notes

Enter remarks

Submit

Disable

Deleting Services

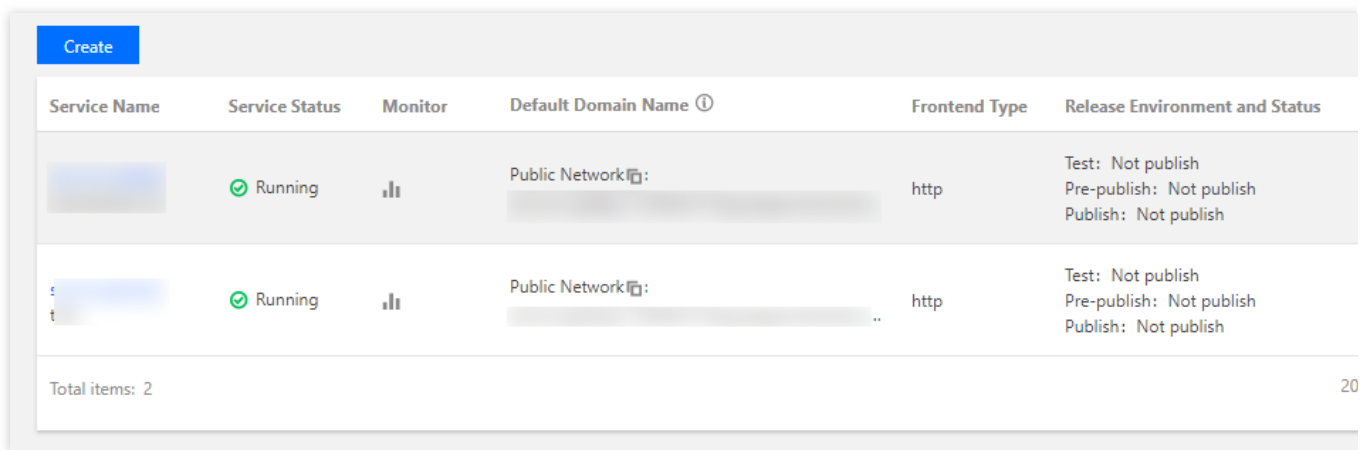
Last updated : 2023-12-22 09:46:22

Operation Scenarios

This document describes how to delete a no longer needed service in the API Gateway Console.

Directions

1. Log in to the [API Gateway Console](#) and select **Service** on the left sidebar.
2. Select the service to be deleted from the service list and click **Delete**.

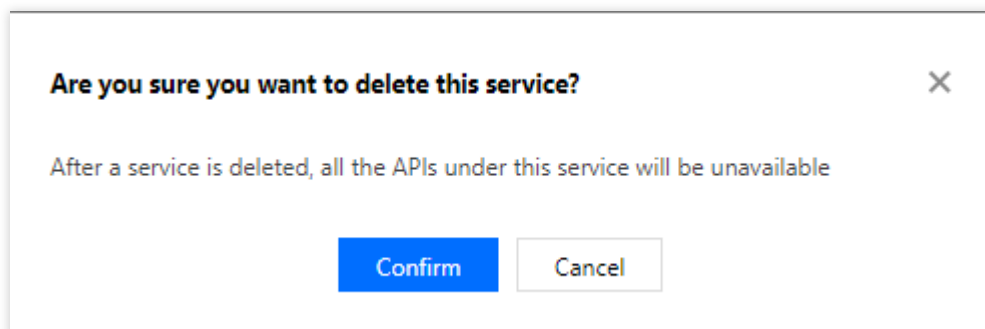


The screenshot shows the 'Service' page in the API Gateway Console. At the top left is a blue 'Create' button. Below it is a table with the following columns: 'Service Name', 'Service Status', 'Monitor', 'Default Domain Name ①', 'Frontend Type', and 'Release Environment and Status'. There are two rows of service data. The first row shows a service with a green 'Running' status and a monitor icon. The second row shows another service with a green 'Running' status and a monitor icon. At the bottom left of the table, it says 'Total items: 2'. At the bottom right, there is a page number '20'.

Service Name	Service Status	Monitor	Default Domain Name ①	Frontend Type	Release Environment and Status
	Running		Public Network	http	Test: Not publish Pre-publish: Not publish Publish: Not publish
	Running		Public Network	http	Test: Not publish Pre-publish: Not publish Publish: Not publish

Total items: 2

3. In the pop-up window, click **Confirm**.



Notes

You can select up to 10 services at a time.

After a service is deleted, APIs under it will no longer provide services, and it cannot be restored. Therefore, please do so with caution.

Tencent Cloud Serverless team provides a quick deletion tool. After simple configuration, you can quickly delete resources in batches. For more information on the tool, please see [Quick Deletion Tool](#).

Service Domain Name

Last updated : 2023-12-22 09:46:32

Operation Scenarios

This document describes how to get a subdomain name for service access in the service details in the API Gateway Console.

Prerequisites

A [service has been created](#).

Directions

1. Log in to the [API Gateway Console](#) and select **Service** on the left sidebar.
2. In the service list, click the target service name to enter the service details page.
3. On the **Service Info** tab, view the subdomain name for accessing the service.

Basic Info

Service Name	exampleservice
Region	Guangzhou
Public Domain	service- XXXXXXXXXX .gz.apigw.tencentcs.com 
IP Version	IPv4
Frontend Type	http
Creation Time	2020-03-11 11:25:59
Remarks	-

Service Domain Name Description

Sample service domain name: `service-a1b2c3d4-1234567890.gz.apigw.tencentcs.com` .

A service subdomain name contains your service name, APPID , and region, such as `http://{your-unique-id}.{region}.apigw.tencentcs.com` .

If both private network VPC and public network are selected as the access method, two subdomain names will be obtained for private network VPC access and public network access, respectively. Private network VPC access is made available through an allowlist. You can [submit a ticket](#) for application.

API Gateway has also enabled the wildcard domain name SSL certificate feature; therefore, the subdomain name of a service can be accessed through the HTTPS protocol at an access address in the format of `https://{your-unique-id}.{region}.apigw.tencentcs.com` .

Binding Service to VPC

Last updated : 2023-12-22 09:46:41

Overview

After binding a [VPC](#) attribute to the service, you can create APIs interconnected with backend resources in the VPC in the service.

Binding Service to VPC in Shared Instance

Binding VPC during service creation

1. Log in to the [API Gateway console](#).
2. Click **Service** on the left sidebar.
3. In the current region, click **Create** in the top-left corner to create a service. In **Basic Information Configuration**, select **Shared Type** for the instance type. In **Network and Optional Configuration**, you can bind the created service to a VPC in the same region.

Binding VPC on service details page

If you don't bind any VPCs during service creation, you can bind one on the service details page to interconnect with VPC resources.

1. Log in to the [API Gateway console](#).
2. Click **Service** on the left sidebar.
3. Select the target service and click the **service name** to enter the service details page.
4. Select **Basic configurations** on the service details page to enter the **Basic configurations** page.
5. On the **Basic configurations** page, find the **VPC** field in **Network**, click the **edit icon** after the field, and then you can bind the service to a VPC in the pop-up window.

Binding Service to VPC in Dedicated Instance

As a dedicated instance has the VPC attribute, you don't need to manually bind it to a VPC. When creating a service in the instance, the VPC field will be automatically set to the VPC of the instance.

Migrating Service from Shared Instance to Dedicated Instance

Last updated : 2023-12-22 09:46:51

Overview

This document describes how to migrate a service from a shared API Gateway instance to a dedicated instance and provides answers to FAQs.

For more information on dedicated instance, see [Instance Specification](#).

Migration Process

Steps	Operation Details	Operator
1	You provide the information of the service to be migrated, including region, <code>appId</code> , and service ID, such as Guangzhou, 123688xxx, and service-0x6u6xxx.	You
2	API Gateway evaluates the configuration of the service to be migrated and outputs an evaluation report.	API Gateway
3	If migration is feasible as evaluated in step 2, you need to evaluate the dedicated instance's specification (based on the business QPS) and VPC parameters (if the original service backend is across VPCs, you also need to configure a CCN instance for VPC interconnection).	You and API Gateway
4	After creating a dedicated instance in step 3, you provide the instance ID to API Gateway for double check.	API Gateway
5	After step 4 is completed, you confirm the migration time.	You
6	At the scheduled time, API Gateway initiates the service migration (usually completed in 1 minute), and you observe the business monitoring data. If the service goes exceptional during or after the migration, API Gateway will roll back the service immediately.	API Gateway and you
7	After the migration is completed, API Gateway will continue to observe the service for a period of time. If there is any exception, it will roll back the service immediately and notify you.	API Gateway

FAQs

Will the service be interrupted during the migration?

In theory, the migration is imperceptible to end users. If an exception occurs, the service will be rolled back immediately.

At present, we guarantee that the domain name of the API Gateway service won't change after the migration, but the ingress and egress IPs will change. If your business client relies on the ingress IP or your backend has an egress IP allowlist, you need to communicate with us in advance.

Can multiple services be migrated to a dedicated instance at the same time?

This needs to be evaluated based on the actual conditions:

Scenario 1: if multiple services to be migrated only have the public network access, they can be migrated to the same dedicated instance.

Scenario 2: if multiple services to be migrated have the private network access (to HTTP 8xxx ports and HTTPS 9xxx ports currently):

If their private network ports are the same, they can be migrated to the same dedicated instance.

Otherwise, they cannot be migrated to the same dedicated instance.

Scenario 3: in scenario 2, if the client supports changing the multiple private network access ports from 8xxx to 80 or from 9xxx to 443, the services can be migrated to the same dedicated instance.

Can service in all shared instances be migrated to a dedicated instance in an imperceptible manner?

No. For historical reasons, if the original service's domain name suffix is `apigateway.myqcloud.com`, we recommend you not migrate it currently, as migration will be perceptible. However, there is an alternative: you can create a service in the dedicated instance, and then use API Gateway's existing API replication feature to sync the API configuration of the original service to the new service. **If the `myqcloud` domain name is configured in a custom domain name or WAF or hardcoded in the client, it needs to be updated synchronously.**

Creating API

API Creation Overview

Last updated : 2023-12-22 09:47:07

Scenarios

This document describes how to create an API in the API Gateway console.

Prerequisites

[Create a service.](#)

Directions

1. Log in to the [API Gateway console](#). Select **Service** in the left sidebar.
2. In the service list, click the name of the target service.
3. In the service details page, click the **Manage API** tab and choose to create a **General API** or **Microservice API** based on the backend service type.
4. Click **Create**.

API Backends

API Gateway supports six types of API backends. General APIs for public URL/IP, VPC, SCF, and Mock, and microservice APIs for TSF. More information is listed below:

API Type	Backend Service	Documentation
General API	Public URL/IP	Creating APIs Connecting to the Public URL/IP Backend
	VPC resources	Creating APIs Connecting to the VPC Resource Backend
	SCF	Creating APIs Connecting to the SCF Backend
	Cloud Object Storage (COS)	Creating APIs Connecting to the Mock Backend

	Mock	Creating APIs Connecting to the Mock Backend
Microservice API	TSF	Creating APIs Connecting to the TSF Backend

Basic API Information

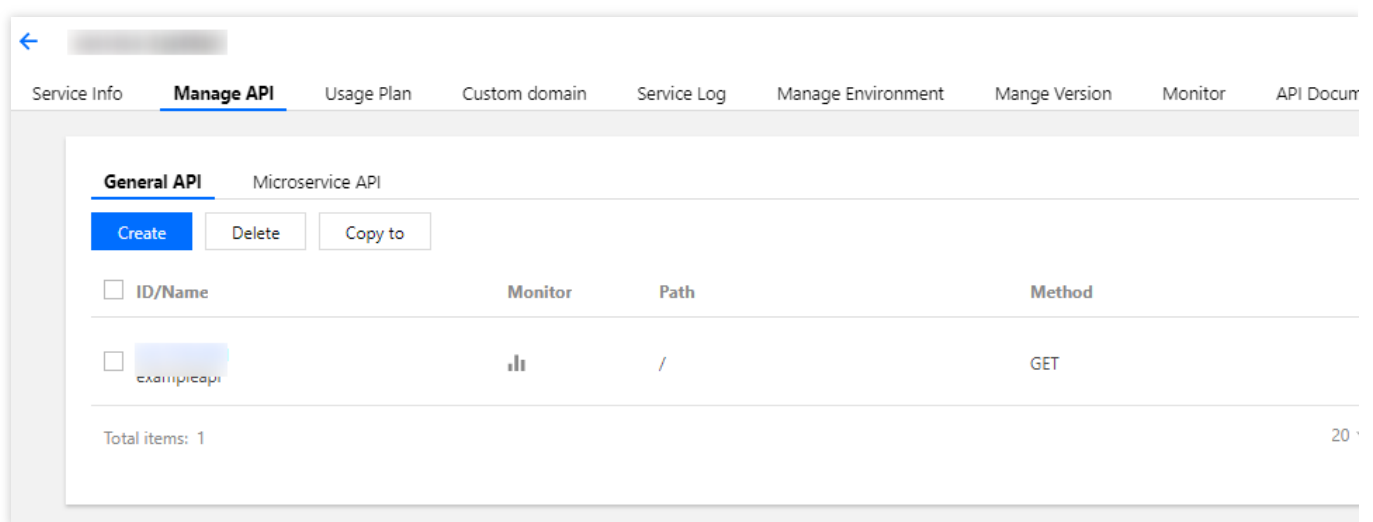
Basic information of an API includes:

API service: a service is a set of APIs for management. Any specific API must belong to a certain service. When creating an API, you can select an existing service or create a new one.

API path: path of an API request domain name

Method: API request method. The API path + API request method is the unique identifier of an API.

Description: API remarks.



Creating APIs Connecting to the Public URL/IP Backend

Last updated : 2023-12-22 09:47:17

Overview

This document describes how to create an API connecting to the public URL/IP in the backend through the API Gateway console.

Prerequisites

You have [created a service](#).

Directions

Step 1. Create a general API

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. In the service list, click the name of the target service to view it.
3. In the service details, click the **Manage API** tab and choose to create a **General API** based on the backend business type.
4. Click **Create** for subsequent configuration.

Step 2. Perform frontend configuration

The frontend configuration of an API refers to the relevant configurations provided for external access, including the API name, frontend type, request protocol, HTTP method, URL path, and input parameter definition.

Configuring basic frontend information

API Name: name of the API you create, which must be unique within the current service and can contain up to 60 characters.

Frontend Type: supports HTTP&HTTPS and WS&WSS.

Path: You can write a valid URL path as needed. If you need to configure dynamic parameters in the path, please use `{ }` to enclose the parameter names. For example, the `/user/{userid}` path declares the `userid` parameter in the path, which must be defined as a path-type input parameter. Query parameters do not need to be defined in the URL path.

Regular expression match is supported. Taking `/user` as an example of the path:

`=/user` : indicates exact match. When there are multiple APIs with `/user` , APIs configured with `=/user` have the highest matching priority.

`/user/{id}` : indicates that there is a dynamic parameter in the path. When there are multiple APIs with `/user` , APIs configured with a dynamic parameter have the third-highest matching priority.

`/user` : indicates access by exact match or prefix match. `/user` , `/usertest` , and `/user/test/a` all can access APIs with the path of `/user` .

Request Method: supports GET, POST, PUT, DELETE, and HEAD.

Authentication Type: supports [No authentication](#), [Key pair](#), and [OAuth 2.0](#).

CORS is supported: configures cross-origin resource sharing (CORS). If enabled, `Access-Control-Allow-Origin : *` will be added to the response header by default.

Configuring frontend parameters

Input parameters: the input parameters include parameters from the header, query, and path, where a path parameter corresponds to a dynamic parameter defined in the URL path. For any parameter, the parameter name, parameter type, and parameter data type must be specified, and whether it is required, its default value, sample data, and description can be specified optionally. With these configuration items, API Gateway helps you with documentation and preliminary verification of input parameters.

1 Frontend Configuration

2 Backend Configuration

3 Response Result

Service

serverless

API Name

Up to 60 chars

Frontend Type

HTTP

WS

Path

1、 Supports starting with "/" and "/=". Starting with "/" means fuzzy match, while starting with "/" means exact match.

2、 Characters supported in the path: uppercase and lowercase letters, numbers, and symbols -, _, ., /, ~, %, &

3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as /(param)/)

4、 When the path starts with "=", adding request parameter of type Path is not supported.

Request Method

GET

POST

PUT

DELETE

HEAD

ANY

Authentication Type

No authentication

Key pair

OAuth 2.0

1. The authentication-free feature means that anyone who can obtain the API service information will be able to call the API. The API gateway does not authentic

2. For the APIs that require no authentication, it is recommended not to publish them on the cloud marketplace. The API gateway cannot differentiate callers ar

CORS is supported

☐

Notes

Parameter Configuration

Parameter Name	Parameter Location ⓘ	Type	Default Value ⓘ	Required	No
Newly added parameter configuration (0/30)					

Next

Note:

If the request protocol is HTTPS, a request must carry SNI. To ensure request security, API Gateway will deny requests without SNI.

SNI (Server Name Indication) is an extension to TLS. It is used to address situations where a server has multiple domain names and is supported by the protocol since TLSv1.2. Previous SSL handshake messages didn't carry the destination address to be accessed by the client. If a server has multiple virtual hosts, each of which has a unique domain name and uses a unique certificate, then it will not be able to determine which certificate to return to the client. SNI addresses this issue by providing the host information in `Client Hello`.

Step 3. Configure public URL/IP in the backend

The backend configuration of an API refers to the configuration that actually provides the service. API Gateway converts a frontend request based on the backend configuration and forwards the call to the actual service.

If your business is deployed in other cloud vendors, or your local server opens using public URL/IP, please choose public URL/IP as the backend type.

Configuration instructions:

1. If public URL/IP is connected to in the backend, set **Backend Type** to **Public URL/IP**.
2. Enter the backend domain name that starts with `http://` or `https://` and does not include the path behind, for example, `http://api.myservice.com` or `http://108.160.162.30`.
3. Enter the backend path that starts with `/`, for example, `/path` or `/path/{petid}`.
4. Select the request method. The request methods for the frontend and the backend can be different.
5. Set the backend timeout (up to 30 seconds). When API Gateway calls the backend service, but the response is not returned within the specified timeout, API Gateway will terminate the call and return the corresponding error message.
6. Set the backend parameters that map the frontend.
7. Click **Next** and configure the response result.

Frontend Configuration

Backend Configuration

Response Result

Backend Type

Public URL/IP

Provide backend services externally through the public network

VPC resources

Access the CVM and container resources in the VPC through the private CLB

Serverless Cloud Function (SCF)

Serverless computing service provided by Tencent Cloud

Mock

Simulate resp

Backend Domain Name ⓘ

http://

It starts with http or https and contains domain content, and "/" is not required at the end. Only a public domain name is supported.

Backend Path ⓘ

1、 It starts with "/", and supports uppercase and lowercase letters, digits, and \$-_.+!*'(),/%.
 2、 "=" and "^~" in the frontend parameters are used for exact match to the frontend path, which are not available to the backend path.
 3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as /(param)/)

Request Method

GET

POST

PUT

DELETE

HEAD

ANY

Backend timeout ⓘ

15

second(s)

Time range: 1~1,800s

Constant parameter

Parameter Name	Parameter Location ⓘ	Parameter Value	Notes
Newly added constant parameter (0/30)			

Previous

Next

Step 4. Configure the response

API response configuration: includes the configuration of API response data and API error codes.

API response data configuration: indicates the type of returned data, including the data samples of successful and failed calls.

API error code definition: indicates the additional error code, error message, and description.

Note:

Currently, API Gateway directly passes through the response result to the requester without processing it. When the SDK documentation is generated, the entered sample responses will also be displayed in the documentation, which will help users better understand the meanings of APIs.

Creating APIs Connecting to the VPC Resource Backend

Last updated : 2023-12-22 09:47:26

Overview

This document describes how to create an API connecting to the VPC resources in the backend through the API Gateway console.

Prerequisites

You have [created a service](#).

Directions

Step 1. Create a general API

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. In the service list, click the name of the target service to view it.
3. In the service details, click the **Manage API** tab and choose to create a **General API** based on the backend business type.
4. Click **Create** for subsequent configuration.

Step 2. Perform frontend configuration

The frontend configuration of an API refers to the relevant configurations provided for external access, including the API name, frontend type, request protocol, HTTP method, URL path, and input parameter definition.

Configuring basic frontend information

API Name: name of the API you create, which must be unique within the current service and can contain up to 60 characters.

Frontend Type: supports HTTP&HTTPS and WS&WSS.

Path: You can write a valid URL path as needed. If you need to configure dynamic parameters in the path, please use `{ }` to enclose the parameter names. For example, the `/user/{userid}` path declares the `userid` parameter in the path, which must be defined as a path-type input parameter. Query parameters do not need to be defined in the URL path.

Regular expression match is supported. Taking `/user` as an example of the path:

`=/user` : indicates exact match. When there are multiple APIs with `/user` , APIs configured with `=/user` have the highest matching priority.

`/user/{id}` : indicates that there is a dynamic parameter in the path. When there are multiple APIs with `/user` , APIs configured with a dynamic parameter have the third-highest matching priority.

`/user` : indicates access by exact match or prefix match. `/user` , `/usertest` , and `/user/test/a` all can access APIs with the path of `/user` .

Request Method: supports GET, POST, PUT, DELETE, and HEAD.

Authentication Type: supports [No authentication](#), [Key pair](#), and [OAuth 2.0](#).

CORS is supported: configures cross-origin resource sharing (CORS). If enabled, `Access-Control-Allow-Origin : *` will be added to the response header by default.

Configuring frontend parameters

Input parameters: the input parameters include parameters from the header, query, and path, where a path parameter corresponds to a dynamic parameter defined in the URL path. For any parameter, the parameter name, parameter type, and parameter data type must be specified, and whether it is required, its default value, sample data, and description can be specified optionally. With these configuration items, API Gateway helps you with documentation and preliminary verification of input parameters.

1 Frontend Configuration

2 Backend Configuration

3 Response Result

Service serverless

API Name
Up to 60 chars

Frontend Type

HTTP

WS

Path

1、 Supports starting with "/" and "=/". Starting with "/" means fuzzy match, while starting with "=/" means exact match.
2、 Characters supported in the path: uppercase and lowercase letters, numbers, and symbols -, _, ., /, ~, %, { }
3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as /(param)/)
4、 When the path starts with "=/", adding request parameter of type Path is not supported.

Request Method

GET

POST

PUT

DELETE

HEAD

ANY

Authentication Type

No authentication

Key pair

OAuth 2.0

1. The authentication-free feature means that anyone who can obtain the API service information will be able to call the API. The API gateway does not authentic
2. For the APIs that require no authentication, it is recommended not to publish them on the cloud marketplace. The API gateway cannot differentiate callers ar

CORS is supported
☐

Notes

Please enter remarks

Parameter Configuration

Parameter Name	Parameter Location ⓘ	Type	Default Value ⓘ	Required	No
Newly added parameter configuration (0/30)					

Next

Note:

If the request protocol is HTTPS, a request must carry SNI. To ensure request security, API Gateway will deny requests without SNI.

SNI (Server Name Indication) is an extension to TLS. It is used to address situations where a server has multiple domain names and is supported by the protocol since TLSv1.2. Previous SSL handshake messages didn't carry the destination address to be accessed by the client. If a server has multiple virtual hosts, each of which has a unique domain name and uses a unique certificate, then it will not be able to determine which certificate to return to the client. SNI addresses this issue by providing the host information in `Client Hello`.

Step 3. Configure VPC resources in the backend

The backend configuration of an API refers to the configuration that actually provides the service. API Gateway converts a frontend request based on the backend configuration and forwards the call to the actual service.

If your hosts and containers are implemented through the VPC, and you want to open up the service capabilities via API Gateway and private CLB, choose VPC resources in the backend.

1. Set **Backend Type** to **VPC resources**.

2. Select the desired VPC in the **Backend Configuration** step. Currently, API Gateway only supports connecting to VPC resources through private CLB.

The screenshot shows the 'Backend Configuration' step in the Tencent Cloud API Gateway console. The 'Backend Type' is set to 'VPC resources'. The 'Backend Domain Name' is 'http://'. The 'Request Method' is 'GET'. The 'Backend timeout' is '15 second(s)'. The 'Constant parameter' table is empty.

Backend Type	Public URL/IP	VPC resources	Serverless Cloud Function (SCF)	Mock
	Provide backend services externally through the public network	Access the CVM and container resources in the VPC through the private network	Serverless computing service provided by Tencent Cloud	Simulate res

VPC Info: Please select

Backend Domain Name: http://

Backend Path:

Request Method: GET POST PUT DELETE HEAD ANY

Backend timeout: 15 second(s)

Constant parameter:

Parameter Name	Parameter Location	Parameter Value	Notes
Newly added constant parameter (0/30)			

3. Select the CLB instance of the backend domain name and the corresponding listener.

If you choose HTTP or HTTPS listener, please make sure that the backend CVM instance has enabled the public network bandwidth; otherwise, network request failure may occur (traffic generated by this policy is not included in the outbound traffic of the public network).

4. Enter `http://vip+port` or `https://vip+port` as the backend domain name. The requests sent to CLB will be HTTP requests or HTTPS requests depending on the content you enter. The VIP here is the VIP of the private network CLB instance, which can be viewed in its basic information (as shown in the screenshot in step 1).

5. Enter the backend path.

If you select the CLB listening type of HTTP/HTTPS, you must configure the backend path as the path configured in the CLB listener.

Domain name and path configured in the CLB listener:

The backend path in API Gateway must be consistent with that in CLB.

You also need to configure a parameter named `host` in **Constant parameter** and place it in the header. The value of this parameter is the domain name configured in the CLB listener.

If you select the CLB listening type of TCP/UDP, the backend path must be set to the path required by the business in the CVM mounted on the CLB.

If you configure host verification in the CVM, you need to configure a parameter named `host` in **Constant parameter**, and select the address to place parameter according to your own business, just like using a layer-7 listener. Subsequent configurations are the same as those of other APIs.

6. When the backend is connected to CLB, security groups on the real CVM instance should open the IP ranges of `100.64.0.0/10` and `9.0.0.0/8`.

Step 4. Configure the response

API response configuration: includes the configuration of API response data and API error codes.

API response data configuration: indicates the type of returned data, including the data samples of successful and failed calls.

API error code definition: indicates the additional error code, error message, and description.

Note:

Currently, API Gateway directly passes through the response result to the requester without processing it. When SDK documentation is generated, the entered sample responses will also be displayed in the documentation, which will help users better understand the meanings of APIs.

Creating APIs Connecting to the SCF Backend

Last updated : 2023-12-22 09:47:35

Overview

This document describes how to create an API connecting to SCF in the backend through the API Gateway console.

Prerequisites

You have [created a service](#).

Directions

Step 1. Create a general API

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. In the service list, click the name of the target service to view it.
3. In the service details, click the **Manage API** tab and choose to create a **General API** based on the backend business type.
4. Click **Create** for subsequent configuration.

Step 2. Perform frontend configuration

The frontend configuration of an API refers to the relevant configurations provided for external access, including the API name, frontend type, request protocol, HTTP method, URL path, and input parameter definition.

Configuring basic frontend information

API Name: name of the API you create, which must be unique within the current service and can contain up to 60 characters.

Frontend Type: supports HTTP&HTTPS and WS&WSS.

Path: You can write a valid URL path as needed. If you need to configure dynamic parameters in the path, please use `{}` to enclose the parameter names. For example, the `/user/{userid}` path declares the `userid` parameter in the path, which must be defined as a path-type input parameter. Query parameters do not need to be defined in the URL path.

Regular expression match is supported. Taking `/user` as an example of the path:

`=/user` : indicates exact match. When there are multiple APIs with `/user` , APIs configured with `=/user` have the highest matching priority.

`/user/{id}` : indicates that there is a dynamic parameter in the path. When there are multiple APIs with `/user` , APIs configured with a dynamic parameter have the third-highest matching priority.

`/user` : indicates access by exact match or prefix match. `/user` , `/usertest` , and `/user/test/a` all can access APIs with the path of `/user` .

Request Method: supports GET, POST, PUT, DELETE, and HEAD.

Authentication Type: supports [No authentication](#), [Key pair](#), and [OAuth 2.0](#).

CORS is supported: configures cross-origin resource sharing (CORS). If enabled, `Access-Control-Allow-Origin : *` will be added to the response header by default.

Configuring frontend parameters

Input parameters: the input parameters include parameters from the header, query, and path, where a path parameter corresponds to a dynamic parameter defined in the URL path. For any parameter, the parameter name, parameter type, and parameter data type must be specified, and whether it is required, its default value, sample data, and description can be specified optionally. With these configuration items, API Gateway helps you with documentation and preliminary verification of input parameters.

1 Frontend Configuration

2 Backend Configuration

3 Response Result

Service serverless

API Name
Up to 60 chars

Frontend Type

HTTP

WS

Path

1、 Supports starting with "/" and "=/". Starting with "/" means fuzzy match, while starting with "=/" means exact match.
2、 Characters supported in the path: uppercase and lowercase letters, numbers, and symbols -, _, *, ., /, ~, %
3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as {param}/)
4、 When the path starts with "=/", adding request parameter of type Path is not supported.

Request Method

GET

POST

PUT

DELETE

HEAD

ANY

Authentication Type

No authentication

Key pair

OAuth 2.0

1. The authentication-free feature means that anyone who can obtain the API service information will be able to call the API. The API gateway does not authen
2. For the APIs that require no authentication, it is recommended not to publish them on the cloud marketplace. The API gateway cannot differentiate callers a

CORS is supported
☐

Notes

Please enter remarks

Parameter Configuration

Parameter Name	Parameter Location ⓘ	Type	Default Value ⓘ	Required	Nc
Newly added parameter configuration (0/30)					

Next

Note:

If the request protocol is HTTPS, a request must carry SNI. To ensure request security, API Gateway will deny requests without SNI.

SNI (Server Name Indication) is an extension to TLS. It is used to address situations where a server has multiple domain names and is supported by the protocol since TLSv1.2. Previous SSL handshake messages didn't carry the destination address to be accessed by the client. If a server has multiple virtual hosts, each of which has a unique domain name and uses a unique certificate, then it will not be able to determine which certificate to return to the client. SNI addresses this issue by providing the host information in `Client Hello`.

Step 3: Configure SCF in the backend

The backend configuration of an API refers to the configuration that actually provides the service. API Gateway converts a frontend request based on the backend configuration and forwards the call to the actual service.

If your business is implemented in SCF and you want to open up your service capabilities through API Gateway, you can select SCF as the backend connection type.

Frontend Configuration

Backend Configuration

Response Result

Backend Type

Public URL/IP

Provide backend services externally through the public network

VPC resources

Access the CVM and container resources in the VPC through the private network

Serverless Cloud Function (SCF)

Serverless computing service provided by Tencent Cloud

Cloud Function

Namespace

default

Name

nextjs_component_otnul8g

Version

Alias: Default Traffic

[create one](#)

Backend timeout

15

second(s)

Time range: 1-1,800s

Response Integration

☐

Base64 Encoding

☐

Previous

Next

When connecting to SCF in the backend, you need to enter the following parameters:

No.	Parameter	Description
1	Namespace	Namespace of the connected function, which is <code>default</code> by default
2	Name	Name of the connected function
3	Version	Version of the connected function. Default value: <code>\$LATEST</code>
4	Backend timeout	The default value is <code>15</code> (seconds).
5	Response integration	See Response integration for more information.

Response integration:

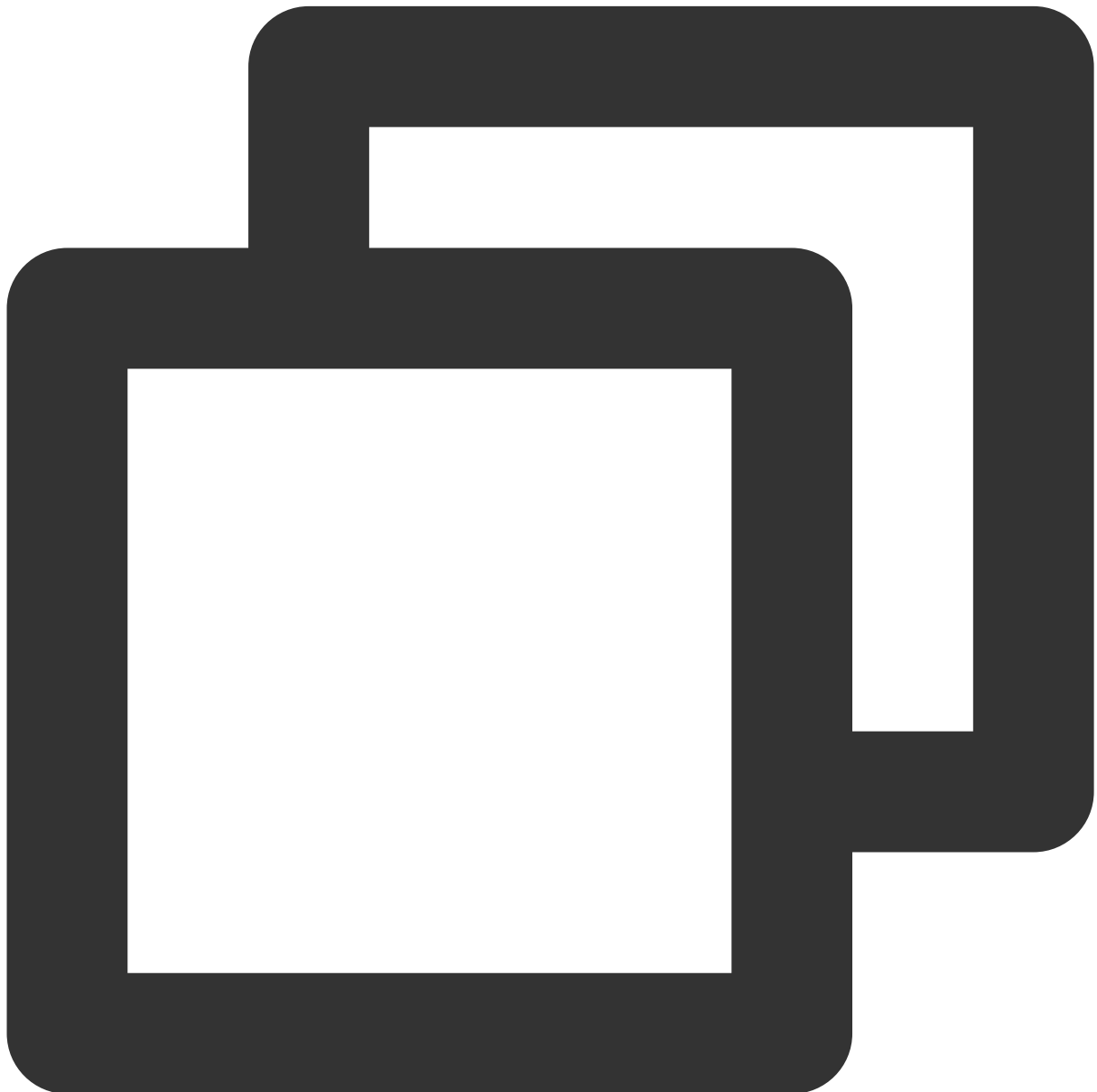
Request method is the method to process requests sent from API Gateway to SCF, and response method is the method to process the returned values sent from SCF to API Gateway. Request method and response method can be implemented using pass-through or integration.

If response integration is not enabled, the pass-through mode will be used. That is, when API Gateway sends a request to SCF, the request information will be combined into a fixed structure, which will be received by SCF. The

response will be passed through without being processed and can only be in JSON format.

If response integration is enabled, the integration mode will be used. That is, when API Gateway sends a request to SCF, the request information will be combined into a fixed structure, and SCF also returns a fixed structure. API Gateway will map the structure returned by SCF to `statusCode`, `header`, `body` as well as other fields, and then return it to the client.

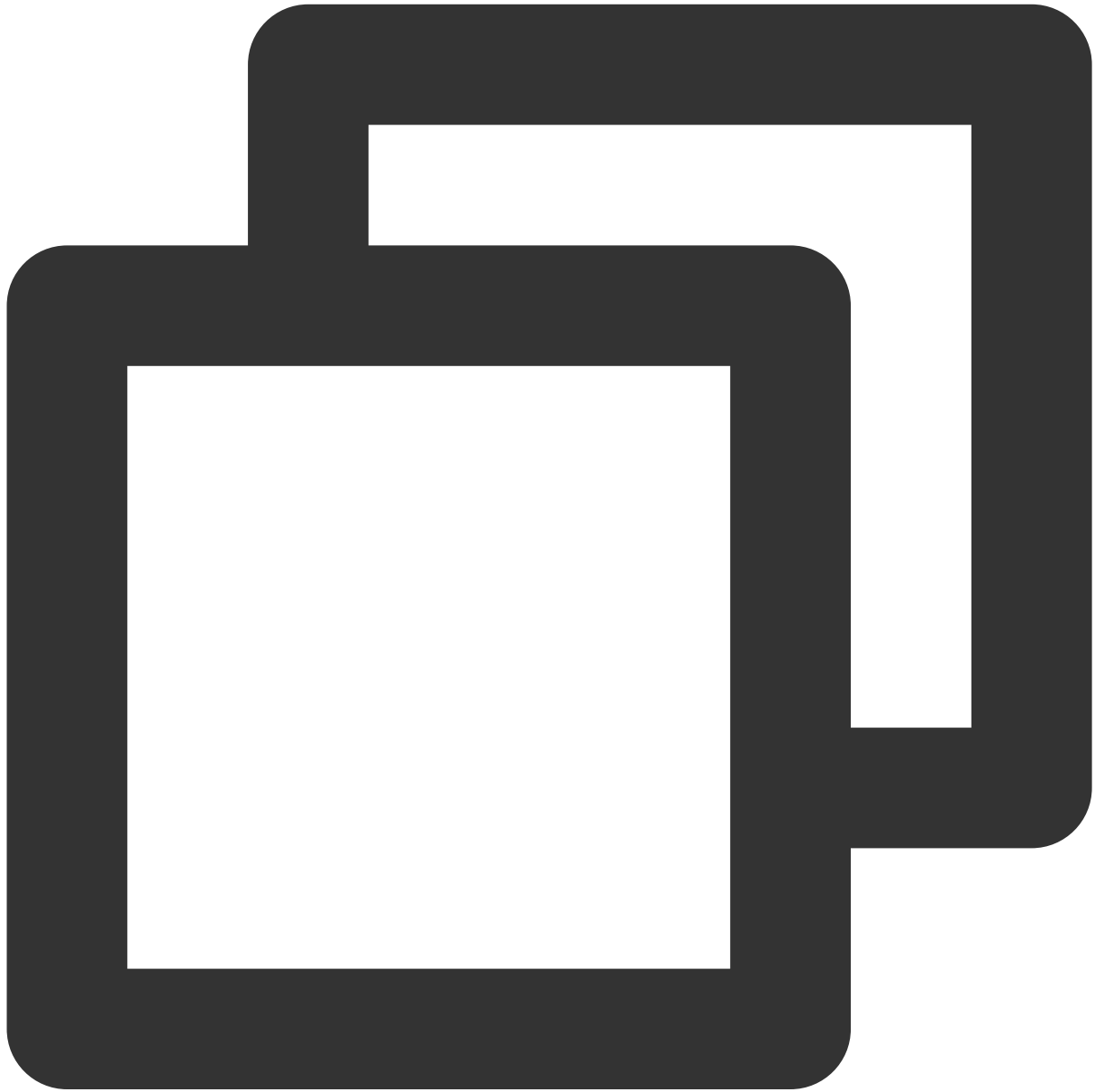
If response integration is enabled, you must configure the SCF to return data in the following format to API Gateway for parsing:



```
{  
    "isBase64Encoded": false, // Whether Base64 encoding is used. Valid values:
```

```
"statusCode": 200, // HTTP request status code
"headers": {"Content-Type": "text/html"}, // `Content-Type` can contain only
"body": "<html><body><h1>Heading</h1><p>Paragraph.</p></body></html>"
}
```

The structure format of requests sent from API Gateway to SCF is as follows:



```
{
  "requestContext": {
    "serviceId": "service-f94sy04v",
    "path": "/test/{path}",
    "httpMethod": "POST",
```

```
{
  "identity": {
    "secretId": "abdcxxxxxxxxsdfs"
  },
  "sourceIp": "10.0.2.14",
  "stage": "release"
},
"headers": {
  "Accept-Language": "en-US,en,cn",
  "Accept": "text/html,application/xml,application/json",
  "Host": "service-3ei3tii4-251000691.ap-guangzhou.apigateway.myqcloud.com",
  "User-Agent": "User Agent String",
  "x-api-requestid": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef"
},
"body": "{\\"test\\":\\"body\\"}",
"pathParameters": {
  "path": "value"
},
"queryStringParameters": {
  "foo": "bar"
},
"headerParameters":{
  "Refer": "10.0.2.14"
},
"stageVariables": {
  "stage": "release"
},
"path": "/test/value",
"queryString": {
  "foo" : "bar",
  "bob" : "alice"
},
"httpMethod": "POST",
"isBase64Encoded": "true"
}
```

Note:

You can implement backend web services by writing SCF functions and providing services through API Gateway which will pass the request content as parameters to the function and return the result from the function back to the requester as the response. For more information, please see [API Gateway Trigger Overview](#).

Step 4. Configure the response

API response configuration: includes the configuration of API response data and API error codes.

API response data configuration: indicates the type of returned data, including the data samples of successful and failed calls.

API error code definition: indicates the additional error code, error message, and description.

Note:

Currently, API Gateway directly passes through the response result to the requester without processing it. When SDK documentation is generated, the entered sample responses will also be displayed in the documentation, which will help users better understand the meanings of APIs.

Creating API for Interconnecting Backend with COS

Last updated : 2023-12-22 09:47:46

Overview

This document describes how to create an API connecting to [COS](#) on the backend in the API Gateway console.

Solution Strengths

API Gateway and COS are interconnected over the private network, which has an excellent performance and doesn't incur public network traffic fees.

API Gateway can calculate the COS signature, so there is no need to transform the client.

Prerequisites

You have [created an API Gateway service](#).

You have [created a COS bucket](#) and [uploaded an object](#).

Directions

Step 1. Create a general API

1. Log in to the [API Gateway console](#) and click **Service** on the left sidebar.
2. In the service list, click the name of the target service.
3. In the service details, click the **Manage API** tab and choose to create a **general API** based on the backend business type.
4. Click **Create** for subsequent configuration.

Step 2. Configure the frontend

The frontend configuration of an API refers to the configuration provided for external access, including the API name, frontend type, request protocol, HTTP method, URL path, and input parameter definition. You need to set the following fields:

Parameter	Required	Description
-----------	----------	-------------

API name	Yes	Name of the API you create, which must be unique within the current service and can contain up to 60 characters.
Frontend type	Yes	Protocol for the client to access API Gateway. API Gateway supports two frontend types: HTTP&HTTPS and WS&WSS .
URL path	Yes	<p>You can write a valid URL path as needed.</p> <p>Configure a dynamic parameter in the path: Use {} to enclose the parameter name. For example, the /user/{userid} path declares the `userid` parameter in the path, which must be defined as a path-type input parameter. A query parameter does not need to be defined in the URL path.</p> <p>Regular expression match is supported for the path: Taking /user as an example of the path:</p> <p><code>=/user</code> : Indicates exact match. When there are multiple APIs with /user , APIs configured with <code>=/user</code> will be matched first.</p> <p><code>/user/{id}</code> : Indicates that there is a dynamic parameter in the path. When there are multiple APIs with /user , APIs configured with a dynamic parameter will be matched first.</p> <p><code>/user</code> : Indicates access by exact match or prefix match.</p> <p><code>/user</code> , <code>/usertest</code> , and <code>/user/test/a</code> all can access APIs with the path of /user .</p>
Request method	Yes	You can select GET, POST, PUT, DELETE, or HEAD.
Authentication type	Yes	API authentication type. Multiple types are supported.
CORS is supported	Yes	It is used to configure cross-origin resource sharing (CORS). After it is enabled, <code>Access-Control-Allow-Origin : *</code> will be added to the response header by default.
Input parameter	No	The input parameters include parameters from the header, query, and path, where a path parameter corresponds to a dynamic parameter defined in the URL path. For any parameter, the parameter name, parameter type, and parameter data type must be specified, and whether it is required, its default value, sample data, and description can be specified optionally. With these configuration items, API Gateway helps you with documentation and preliminary verification of input parameters.

Click **Next** for backend configuration.

Step 3. Configure the backend to interconnect with COS

The backend configuration of an API refers to the configuration that actually provides the service. API Gateway converts a frontend request based on the backend configuration and forwards the call to the actual service. If your

business is deployed in COS and the backend needs to be interconnected with COS, you need to select COS as the backend type.

Configuration instructions:

Parameter	Required	Description
Action	Yes	COS API to be interconnected with API Gateway, which is linked with the request method in the frontend configuration. For more information, see Notes .
Bucket	Yes	Only a bucket in the same region as API Gateway can be interconnected with.
Calculate COS signature	Yes	After it is enabled, the COS signature will be calculated in API Gateway, so the system doesn't need to request the client to calculate the signature.
Backend Path	Yes	It must start with "/". API Gateway will match objects in the bucket based on the path.
Path Match Method	Yes	<p>Full path match: Objects in the bucket are matched by the combination of the frontend path and backend path. This is suitable for scenarios where objects to be manipulated are at different paths passed in by the client. For the specific match rules, see Notes.</p> <p>Backend path match: An object in the bucket is matched by only the frontend path. This is suitable for scenarios where only a fixed object needs to be manipulated. No matter which path the client uses for access, the request will be always forwarded to the COS object at the backend path.</p>

Notes

Linkages between request method and Action

Linkages between the request method and `Action` are as detailed below:

Request Method Selected in Frontend Configuration	Action in Backend Configuration
GET	GetObject
PUT	PutObject
POST	PostObject, AppendObject
HEAD	HeadObject
DELETE	DeleteObject

Full path match rules

Passthrough path = backend path + request path - frontend path. Below are sample paths in four cases:

Path	Case 1	Case 2	Case 3	Case 4
Frontend path	/	/a	/a	/a
Backend path	/	/b	/	/b
Request path	/a/b	/a/b	/a/b	/a/c
Passthrough path	/a/b	/b/b	/b	/b/c

Creating APIs Connecting to the Mock Backend

Last updated : 2023-12-22 09:48:04

Overview

This document describes how to create an API connecting to Mock in the backend through the API Gateway console.

Caution:

If the API connects to Mock in the backend, only fixed data can be returned. Therefore, you are advised to use Mock for testings but not in actual business scenarios.

Prerequisites

You have [created a service](#).

Directions

Step 1. Create a general API

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. In the service list, click the name of the target service to view it.
3. In the service details, click the **Manage API** tab and choose to create a **General API** based on the backend business type.
4. Click **Create** for subsequent configuration.

Step 2. Perform frontend configuration

The frontend configuration of an API refers to the relevant configurations provided for external access, including the API name, frontend type, request protocol, HTTP method, URL path, and input parameter definition.

Configuring basic frontend information

API Name: name of the API you create, which must be unique within the current service and can contain up to 60 characters.

Frontend Type: supports HTTP&HTTPS and WS&WSS.

Path: You can write a valid URL path as needed. If you need to configure dynamic parameters in the path, please use `{ }` to enclose the parameter names. For example, the `/user/{userid}` path declares the `userid`

parameter in the path, which must be defined as a path-type input parameter. Query parameters do not need to be defined in the URL path.

Regular expression match is supported. Taking `/user` as an example of the path:

`=/user` : indicates exact match. When there are multiple APIs with `/user` , APIs configured with `=/user` have the highest matching priority.

`/user/{id}` : indicates that there is a dynamic parameter in the path. When there are multiple APIs with `/user` , APIs configured with a dynamic parameter have the third-highest matching priority.

`/user` : indicates access by exact match or prefix match. `/user` , `/usertest` , and `/user/test/a` all can access APIs with the path of `/user` .

Request Method: supports GET, POST, PUT, DELETE, and HEAD.

Authentication Type: supports [No authentication](#), [Key pair](#), and [OAuth 2.0](#).

CORS is supported: configures cross-origin resource sharing (CORS). If enabled, `Access-Control-Allow-Origin : *` will be added to the response header by default.

Configuring frontend parameters

Input parameters: the input parameters include parameters from the header, query, and path, where a path parameter corresponds to a dynamic parameter defined in the URL path. For any parameter, the parameter name, parameter type, and parameter data type must be specified, and whether it is required, its default value, sample data, and description can be specified optionally. With these configuration items, API Gateway helps you with documentation and preliminary verification of input parameters.

Note:

If the request protocol is HTTPS, a request must carry SNI. To ensure request security, API Gateway will deny requests without SNI.

SNI (Server Name Indication) is an extension to TLS. It is used to address situations where a server has multiple domain names and is supported by the protocol since TLSv1.2. Previous SSL handshake messages didn't carry the destination address to be accessed by the client. If a server has multiple virtual hosts, each of which has a unique domain name and uses a unique certificate, then it will not be able to determine which certificate to return to the client. SNI addresses this issue by providing the host information in `Client Hello`.

Mock returns a response that has a fixed configuration to an API request. It is typically used for development testing. API configuration and response can be completed in advance before the backend service is completed. To connect with Mock, you only need to configure your returned data and click **Complete**.



Frontend Configuration



Backend Configuration

Backend Type

Public URL/IP

Provide backend services externally through the public network

VPC resources

Access the CVM and container resources in the VPC through the private CLB

Serverless Cloud Function (SCF)

Serverless computing service provided by Tencent Cloud

Return data

Please enter the return data

Previous

Complete

Creating APIs Connecting to the TSF Backend

Last updated : 2023-12-22 09:48:13

Overview

This document describes how to create an API connecting to the TSF in the backend through the API Gateway console.

Prerequisites

You have [created a service](#).

Directions

Step 1. Create a microservice API

1. Log in to the [API Gateway console](#).
2. In the service list, click the name of the target service to view it.
3. In the service details, click the **Manage API** tab and choose to create a **General API** or **Microservice API** based on the backend business type. If your business is implemented in TSF, select **Microservice API**.
4. Click **Create** for subsequent configuration.

Step 2. Perform frontend configuration

1. Enter the API name.
2. Enter the URL path.

You can write a valid URL path as needed. If you need to configure a dynamic parameter in the path, use `{ }` to enclose the parameter name. For example, the `/user/{userid}` path declares the `userid` parameter in the path, which must be defined as a path-type input parameter. A query parameter does not need to be defined in the URL path.

3. Select the request method.

The request method is HTTP method. You can choose from GET, POST, PUT, DELETE, HEAD, and ANY.

4. Select the authentication type: no authentication or key pair.
5. Select whether to support CORS.

6. Enter the parameter configuration.

1 Frontend Configuration

2 Backend Configuration

3 Response Result

Service serverless

API Name

Up to 60 chars

Frontend Type

HTTP WS

The microservice API currently supports the http protocol.

Path

1、 Supports starting with "/" and "=/". Starting with "/" means fuzzy match, while starting with "=/" means exact match.
2、 Characters supported in the path: uppercase and lowercase letters, numbers, and symbols -, _, *, ., /, ~, %
3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as {param}/)
4、 When the path starts with "=/", adding request parameter of type Path is not supported.

Request Method

GET POST PUT DELETE HEAD ANY

Authentication Type

No authentication Key pair OAuth 2.0

1. The authentication-free feature means that anyone who can obtain the API service information will be able to call the API. The API gateway does not authenti
2. For the APIs that require no authentication, it is recommended not to publish them on the cloud marketplace. The API gateway cannot differentiate callers at

CORS is supported
☐

Notes

Please enter remarks

Parameter Configuration

Parameter Name	Parameter Location ⓘ	Type	Default Value ⓘ	Required	No
X-NameSpace-Code ⓘ	Header	string		<input checked="" type="checkbox"/>	T
X-MicroService-Name ⓘ	Header	string		<input checked="" type="checkbox"/>	T

Newly added parameter configuration (2/30)

Input parameters include parameters from the header, query, and path locations, where a path parameter corresponds to a dynamic parameter defined in the URL path.

For any parameter, the parameter name, parameter type, and parameter data type must be specified. Whether a parameter is required and its default value, sample data, and description can be specified optionally. Using these configuration items, API Gateway helps you with the documentation and preliminary verification of input parameters.

Two required parameters `X-NameSpace-Code` and `X-MicroService-Name` need to be passed in for the call. They control which microservice the API Gateway request will be sent to and can be placed in header, path, or query. If the parameters are placed in path, just like for general APIs, you need to configure the path parameter in the path, such as `/ {X-NameSpace-Code} / {X-MicroService-Name}`. If the variable `X-NameSpace-Code` is

`crgt` and `X-MicroService-Name` is `coupon-activity` , then the access URL will be `https://access domain name/crgt/coupon-activity/` . Except these 2 fixed parameters, other parameters can be configured in the same way as the general APIs are.

In the namespace of [Tencent Service Framework](#), the `X-NameSpace-Code` path parameter is the code value of the namespace selected for the backend configuration.

In the service management of [Tencent Service Framework](#), the `X-MicroService-Name` path parameter is the microservice name of the cluster selected for the backend configuration.

7. Click **Next** and configure the backend.

Step 3. Configure TSF in the backend

1. Select the cluster and namespace of the microservices to be interconnected with.
2. Select the microservices. The API publisher can integrate multiple microservices in 1 API.

Please make sure that the added microservices, including those deployed on CVMs and containers, can be accessed by API Gateway (over the public network and NodePort).

Frontend Configuration

Backend Configuration

Response Result

Cluster

Please select cluster

Namespace

Please select namespace

Service List

Please select microservice

Please enter a keyword

Microservice Name

No data yet

Microservice Name

Nc

Please ensure that the added service can be accessed by the API gateway and the service instance must be deployed in the same way.

Backend Path

1、 It starts with "/", and supports uppercase and lowercase letters, digits, and \$-_.+!*()/%.

2、 "=" and "^~" in the frontend parameters are used for exact match to the frontend path, which are not available to the backend path.

3、 The Path parameter must be wrapped with curly braces {} as a separate part of the path (such as /(param)/)

4、 The backend path does not need to contain Path types, such as X-NameSpace-Code and X-MicroService-Name.

Backend timeout

15

second(s)

Time range: 1-1,800s

Load balancing method

RoundRobinRule

Session Persistence

Enable

Note:

Currently, API Gateway only supports forwarding requests to service instances of the same deployment type (virtual machine or container) in TSF. If there are microservice instances deployed on both virtual machines and containers under a service, API Gateway cannot be used as the request entry.

3. Configure the backend path.

This refers to the specific backend service request path. If you need to configure a dynamic parameter in the path, use `{ }` to enclose the parameter name. In the parameter mapping configuration, this parameter name will be configured as the input parameter from the frontend configuration. The path here can be different from the frontend path. The backend path is the actual service request path.

4. Set the backend timeout period.

This refers to the timeout period of the backend service call initiated by API Gateway (up to 30 seconds). During a call, if there is no response within the timeout period, API Gateway will terminate the call and return the corresponding error

message.

5. Select the load balancing method.

6. Set the session persistence.

7. Set the parameters.

Backend parameters: `X-NameSpace-Code` and `X-MicroService-Name` are fixed parameters and cannot be mapped. Other configured parameters can be mapped.

If your `Body` parameter only has a form format, you can directly map the frontend and backend parameters when configuring them. If it is in JSON format, the JSON parameter will be directly passed through by API Gateway.

Mapped parameters: parameter mapping is used to map the input parameters from the frontend to the parameters of the actual backend service. By default, parameter mapping will map an input parameter by using the same name and parameter location. In addition, you can change the parameter mapping method as needed. For example, you can map the input parameter from path to the query parameter in the backend service.

Constant parameters: You can add custom constant parameters as needed. They remain the same in each API call. In addition, you can use system parameters to pass certain required system information to the backend service.

8. Click **Next** and configure the response result.

Step 4. Configure the response

API response configuration: includes the configuration of API response data and API error codes.

API response data configuration: indicates the type of returned data, including the data samples of successful and failed calls.

API error code definition: indicates the additional error code, error message, and description.

Note:

Currently, API Gateway directly passes through the response result to the requester without processing it. When SDK documentation is generated, the entered sample responses will also be displayed in the documentation, which will help users better understand the meanings of APIs.

Instructions

To allow access to the backend microservice through API Gateway, you need to open the security group of the virtual machine where the microservice resides to the internet. You can set the source, protocol port, and access policy of the security group access.

When setting the access source, you need to at least open the IP ranges 9.0.0.0/8 and 100.64.0.0/10 where API Gateway resides. You can also open other sources as needed.

For an application deployed on a virtual machine, the corresponding service port of the virtual machine must be opened. For an application deployed on a container, the service port of the virtual machine where the container resides, instead of NodePort, needs to be opened.

For container applications, IP drifting occurs often. Therefore, we recommend that you open service ports that run on containers and need to be exposed externally on all machines in the cluster.

Importing APIs

Last updated : 2023-12-22 09:48:22

Overview

This document describes how to create an API by importing an OpenAPI file in the API Gateway console.

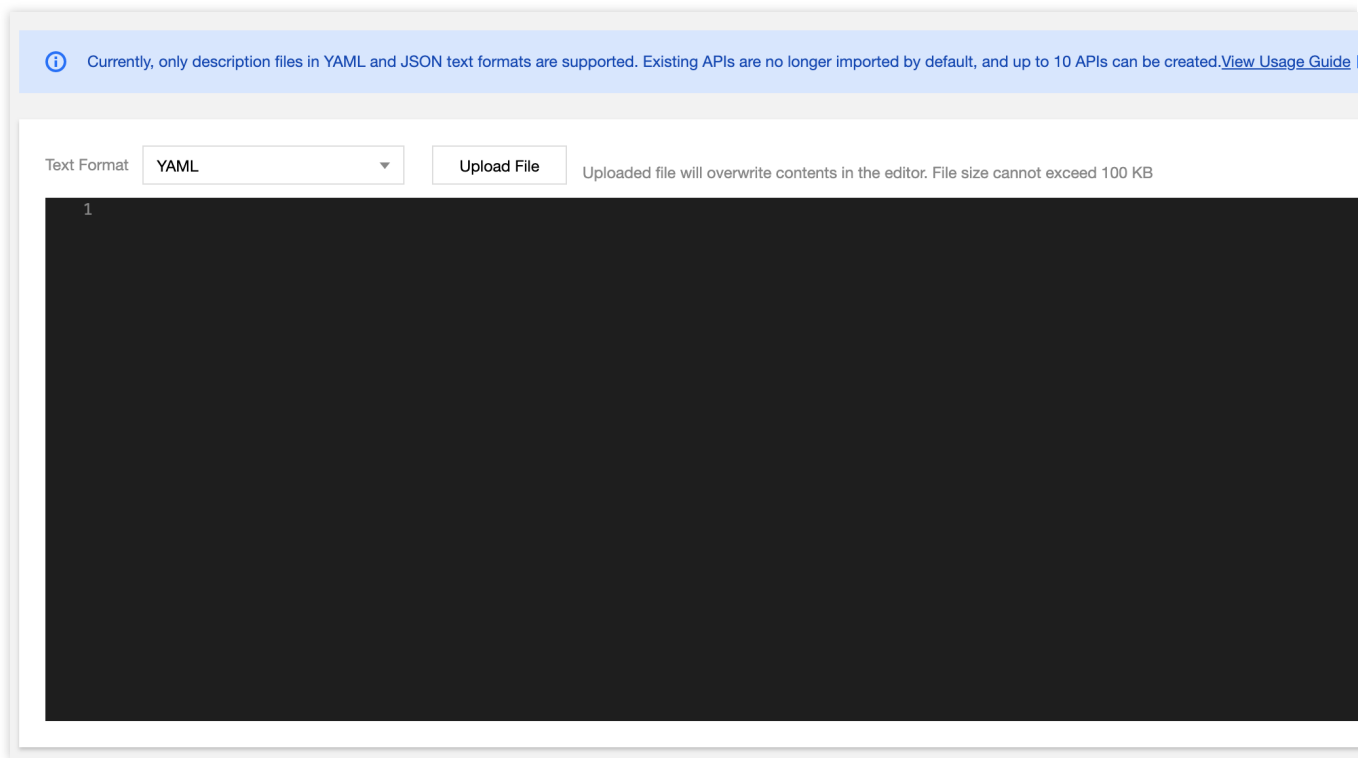
Prerequisites

You have [created a service](#).

You have an OpenAPI 3.0.0 API description file in YAML or JSON format.

Directions

1. Log in to the [API Gateway console](#) and click **Service** on the left sidebar.
2. In the service list, click the name of the target service.
3. In the service details, click the **Manage API** tab to view the API list of the service.
4. Click **Import API** above the API list to enter the API importing page.
5. Select the **Text format** (YAML or JSON), click **Upload File**, and select API description files, or directly enter the API descriptions in the code editor.
6. Click **Save**, and API Gateway will create APIs based on the descriptions and then return a list of successfully created APIs.



Notes

OpenAPI 3.0.0 API description files in YAML or JSON format are supported.

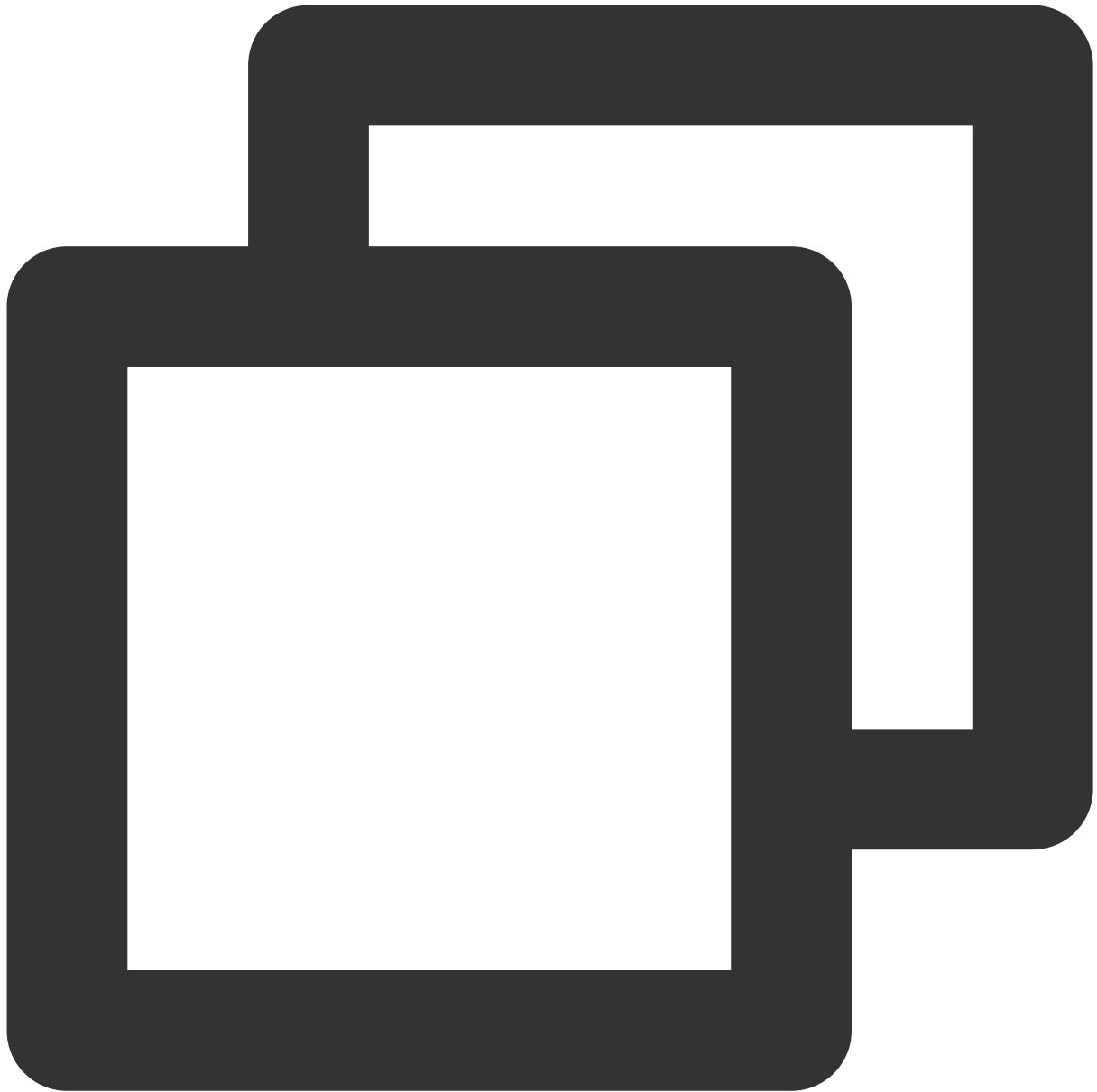
You can import up to ten APIs at a time.

The uploaded API description files must be suffixed with ".yaml" or ".json", and each file can be up to 100 KB in size.

The uploaded description files will overwrite the content in the code editor.

An API created successfully will not be published automatically and needs to be published manually to take effect.

To import an API from an OpenAPI 3.0.0 description file, you need to manually add the `servers` field to define the backend address to be accessed by the API, for example:



```
servers:  
- url: 'http://localhost:8080'  
  description: Generated server url
```

Note:

For more information on the mapping between OpenAPI specifications and API Gateway, see [Defining APIs](#).

For detailed directions, see [Importing APIs](#).

API Management

Debugging General APIs

Last updated : 2023-12-22 09:48:51

Overview

The API debugging page allows you to verify the correctness of an API immediately after completing its configuration by initiating a simulated API call and viewing the specific request response. If the API fails to work as expected, you can modify the configuration according to the response to make it meet your design expectations.

Prerequisites

You have [created a general API](#).

Directions

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to enter the service list page.
2. Click a service name to enter the service details page, and click **Manage API** at the top to manage the API.
3. Select the API to be debugged in the general API list. Click **Debug** in the operation column to go to the debugging page.

On this page, you can see the frontend configuration information of the API to be debugged, including the path, request method and Content-Type. For the request parameters (not displayed if you do not configure them):

If you configure a parameter to have a default value, the default value will be populated in the input box by default;

If you configure a parameter to be required, a check will be performed to verify whether any value has been entered for it before testing;

If your request method is POST, PUT, or DELETE, you will be required to enter a value for the `Body` parameter.

Note:

If a parameter is optional and the user does not enter any value for the parameter, API Gateway will send a `null` to the backend by default.

4. Click **Send request**, you will see the response body and response headers.

Response: the response code and data actually returned to the frontend after the API call.

API Info

API Name	exampleapi
Path	/
Request Method	GET
Content - Type	<input type="text" value="application/x-www-form-urlencoded"/>

Send Request

Return Result

Return Code	200
Response latency	9ms
Response body	hello world, hello apigatw
Response Headers	<pre>HTTP/1.1 200 OK Date: Mon, 23 Mar 2020 07 Content-Type: application/ Transfer-Encoding: chunke Connection: keep-alive X-API-RequestId: ! X-API-ID: X-Service-RateLimit: 500/5 X-API-RateLimit: unlimited Server: apigw/1.0.15 Access-Control-Allow-Orig Access-Control-Expose-He gePlan-RateLimit,X-UsageF tent-Disposition,Date,Keep Accept-Encoding,Accept-La m,Host,If-Match,If-Modifie fied-Since,Range,Origin,Rel arded-Host,X-Forwarded-P ntent-Security-Policy,ETag,i Cookie,Trailer,Transfer-En</pre>

Debugging Microservice APIs

Last updated : 2023-12-22 09:49:01

After an API is created, you can use the debugging feature on the API details page.

On the debugging page, enter corresponding request parameters and send the request.

The `X-NameSpace-Code` and `X-MicroService-Name` parameters here are required. If you want to configure other parameters, please see [Debugging General APIs](#) for value rules.

Compressing Responses

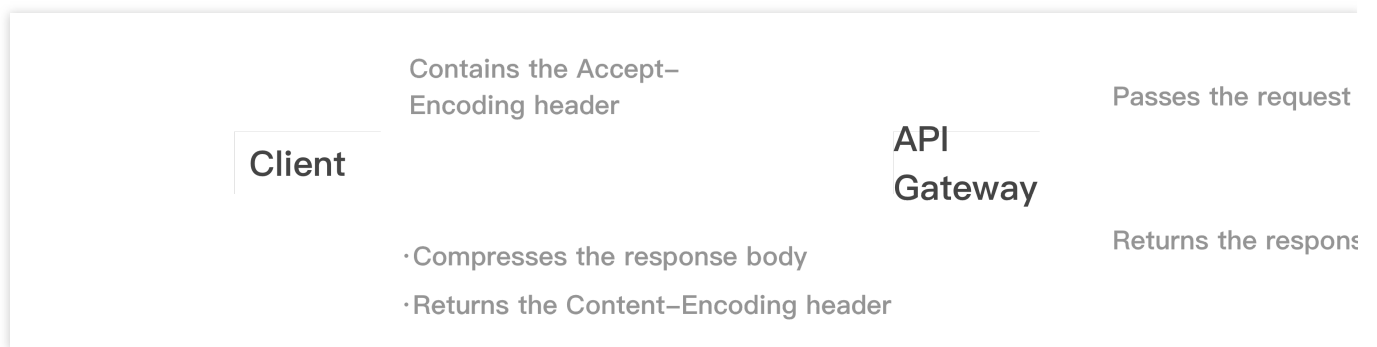
Last updated : 2023-12-22 09:49:10

Overview

For the HTTP protocol, compressing the response data can reduce the amount of data transferred, thereby shortening response time, saving server bandwidth, and improving client performance.

This task guides you on how to compress a response through API Gateway.

Interaction Process



Operation Description

By default, API Gateway supports response compression based on the GZIP compression algorithm. To implement this feature, the following requirements need to be met:

The client request contains the **Accept-Encoding** header and its value contains **gzip**.

The response body size of the client is greater than 1 KB.

Content-Type of the response body is **text/xml**, **text/plain**, **text/css**, **application/javascript**, **application/x-javascript**, **application/rss+xml**, **application/xml**, **application/json**, or **application/octet-stream**.

If the client request meets the requirements above, API Gateway will return the compressed response body to the client and contain the **Content-Encoding: gzip** header in the response.

Notes

Response compression supports only the HTTP and HTTPS protocols but not WebSocket.

If the backend is connected to SCF and response integration is enabled, the structure that SCF returns to API Gateway cannot contain the `Content-Encoding: gzip` header. Otherwise, response compression will not take effect.

Base64 Encoding

Last updated : 2023-12-22 09:49:20

Overview

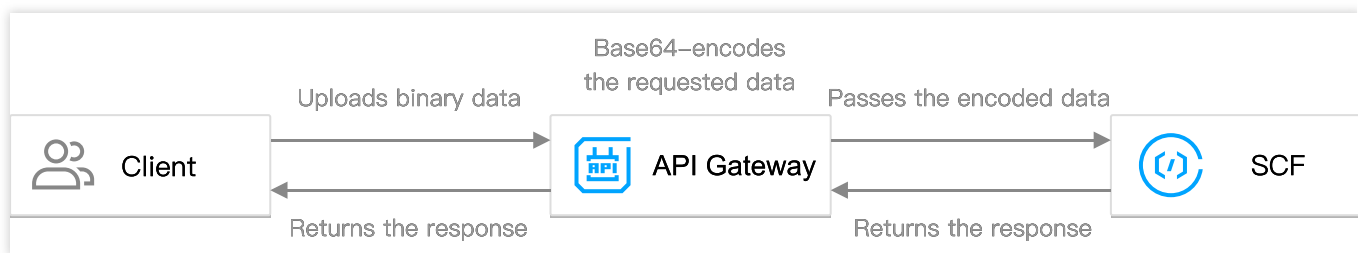
If Tencent Cloud API Gateway is connected with the SCF backend, binary data cannot be directly uploaded due to the trigger limits. Therefore, binary data must be Base64-encoded before the upload. API Gateway allows you to Base64-encode the client's request body before it is passed to SCF. Once the Base64 encoding feature is enabled, you can upload binary data to SCF without modifying the client code.

The Base64 encoding feature provides the following two trigger modes to facilitate the needs of different scenarios:

Trigger All: Base64-encodes all requested content for all requests before passing the content to SCF.

Trigger by Header: Base64-encodes the requested content based on the trigger rules (required) configured. API Gateway will verify the request headers according to the trigger rules. Only requests with the specific `Content-Type` or `Accept` header will be Base64-encoded before being passed to SCF. Requests that do not meet the rules will be passed to SCF without being Base64-encoded.

Interaction Process



Directions

Configuring trigger all

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. On the service list page, click the desired service name to view the API list.
3. Click **Create**, configure the API frontend, and click **Next**.

4. In the **Backend Configuration** step, choose **Serverless Cloud Function (SCF)** in the **Backend Type** filed, enable **Base64 Encoding**, and configure other fields as needed. In this way, Base64 encoding will be enabled for the created API with the default trigger mode **Trigger All**.

Frontend Configuration > **Backend Configuration** > Response Result

Backend Type

Public URL/IP Provide backend services externally through the public network	VPC resources Access the CVM and container resources in the VPC through the private network	Serverless Cloud Function (SCF) Serverless computing service provided by Tencent Cloud
---	--	--

Cloud Function

Namespace: default

Name: nextjs_component_otnul8g

Version: Version: \$LATEST

Backend timeout ⓘ: 15 second(s)
Time range: 1-1,800s

Response Integration ⓘ ☒

Base64 Encoding ⓘ ☐

Previous Next Complete Now

Configuring trigger by header

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar.
2. On the service list page, click the desired service name to view the API list.
3. Click the desired API (backend of the desired API must be SCF) to go to the API detail page. Then, choose the **Basic Configurations** tab and find the **Base64 Encoding** area.
4. Click **Edit** on the right of **Base Encoding** and click **Trigger by Header > Add Trigger Rule**. Then, select a header in **Parameter** and set the value as needed.
5. Click **Save**.

Base64 Encoding

Current Status

Trigger All

Trigger by Header

Close

Trigger Rule

Parameter	Value	Operation
Current list is empty		
Add Trigger Rule (0/10)		

Save

Cancel

After you enable this feature, API Gateway will Base64-encode your request content before sending it to SCF to support binary file upload.

You can configure Base64 encoding to be triggered for all requests or based on specific Content-Type and Accept headers. For more information, please see [Base64 Encoding Instruction](#)

Notes

For requests that have successfully triggered Base64 encoding, API Gateway will Base64-encode the request body and set the value of the `isBase64Encoded` field to `True` before passing the requests to SCF. You can determine whether the requests are Base64-encoded according to the value of this field. (For more information, please see [Structures Passed by API Gateway to Backend](#)).

Due to the trigger limits, the requested content that API Gateway can pass to SCF at a time cannot be larger than 6 MB. In other words, the Base64 encoding feature only supports passing files that are smaller than 6 MB after being Base64-encoded. To pass files larger than 6 MB, please upload with COS by referring to [Uploading Files](#).

The **Trigger by Header** mode adopts fuzzy match for the trigger rules. Only the `Content-Type` and `Accept` request headers are supported. API Gateway applies a logical OR across multiple trigger rules. That is, Base64 encoding will be triggered as long as one of the trigger rules is met.

Calling API

Calling Key Pair Authentication API

Last updated : 2023-12-22 09:49:43

Overview

This document describes how to call the key pair authentication API.

Prerequisites

Creating key pair authentication API

1. In the [API Gateway console](#), create an API and select the authentication type as "key pair authentication" (for more information, see [API Creation Overview](#)).
2. Release the service where the API resides to an environment. See [Service Release and Deactivation](#).
3. Create a key pair on the key management page in the console.
4. Create a usage plan on the usage plan page in the console and bind it to the created key pair (for more information, see [Sample Usage Plan](#)).
5. Bind the usage plan to the API or the service where the API resides.

Confirming information

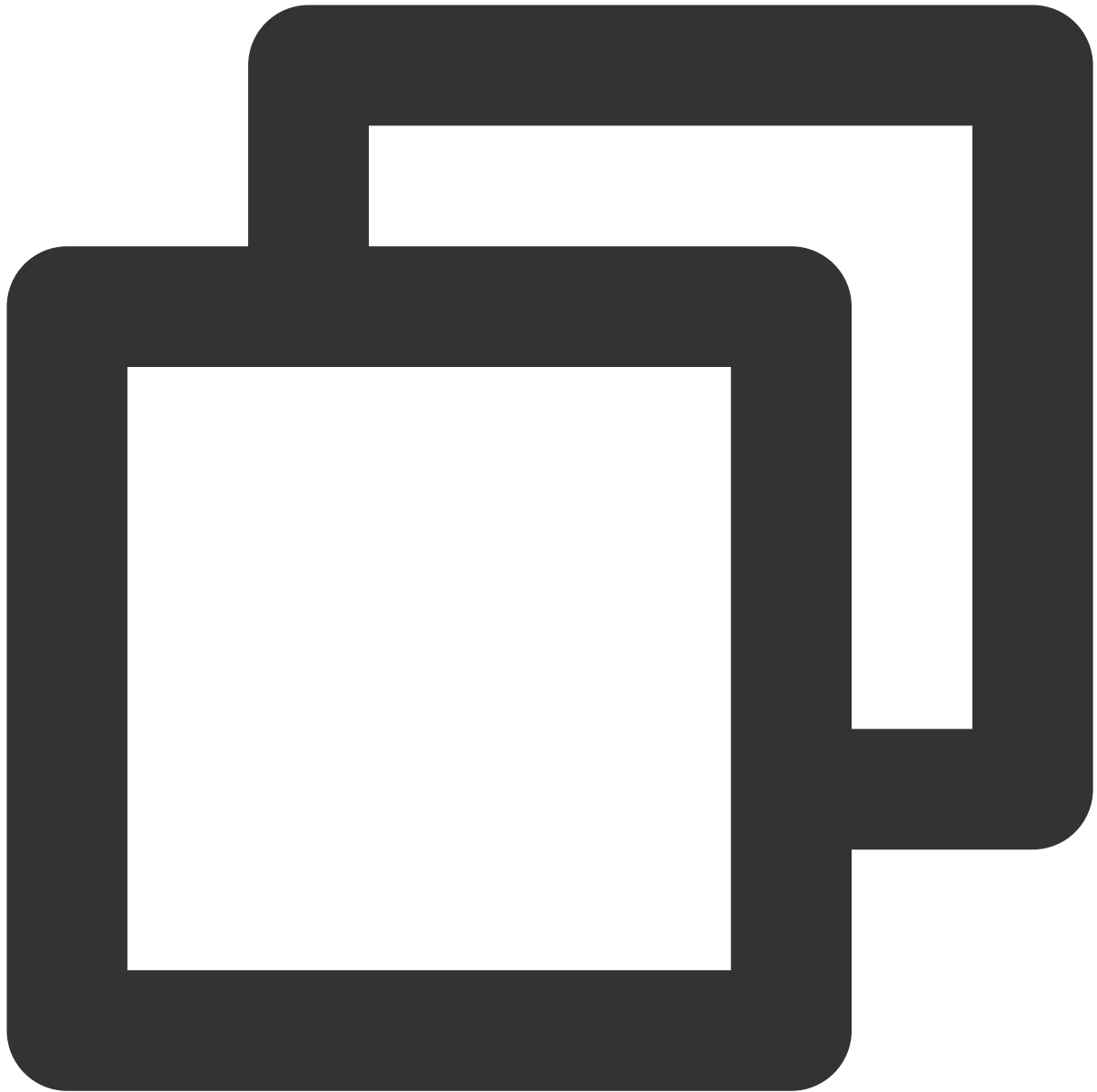
Before calling the API, you must have the `SecretId` and `SecretKey` of the API to be called and understand the API request path, method, and parameters.

Preparing tools

You can initiate requests from sources including browsers, browser plugins, Postman, and clients. Postman is recommended for simple validation.

Sample Call

Address



```
http://service-xxxxxxx-1234567890.ap-guangzhou.apigateway.myqcloud.com/release  
// Enter the URL of the API service you want to call
```

URL concatenation rule: service path + environment parameter + API path

Methods



POST

Request body



```
QueryParam_a=value1&QueryParam_b=value2
```

Request headers



```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-cn
Connection: Keep-Alive
Host: service-xxxxxxx-1234567890.ap-guangzhou.apigateway.myqcloud.com/release
User-Agent: Mozilla/4.0 (compatible;MSIE5.01;Window NT5.0)
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded;charset=utf-8
// Request body type, which should be set according to the actual request body cont
X-Client-Proto: http
X-Client-Proto-Ver: HTTP/1.1
X-Real-IP: 163.xxx.xx.244
```

```
X-Forwarded-For: 106.xxx.xx.102, 163.xxx.xx.244
Date: Sun, 21 Sep 2017 06:18:21 GMT
Authorization: hmac id="AKIDCgXXXXXXXXX48pN", algorithm="hmac-sha1", headers="Date H
// Signature. For specific signature algorithms, see the key calculation method in
```

The eventually delivered HTTP request contains at least two headers: `Date` or `X-Date` and `Authorization`. More optional headers can be added in the request. If `Date` is used, the server will not check the time; if `X-Date` is used, the server will check the time.

The value of `Date` header is the construction time of the HTTP request in GMT format, such as Fri, 09 Oct 2015 00:00:00 GMT.

The value of `X-Date` header is the construction time of the HTTP request in GMT format, such as Mon, 19 Mar 2018 12:08:40 GMT. It cannot deviate from the current time for more than 15 minutes.

For the value of `Authorization` header, see [Key Pair Authentication](#). For sample signatures generated in different programming languages, see [API Gateway](#).

Response Processing

Response code

Response Code	Description
$200 \leq \text{code} < 300$	Success
$300 \leq \text{code} < 400$	Redirect, which requires subsequent operations to complete the request
$400 \leq \text{code} < 500$	Client error
$\text{code} > 500$	Server error

Response headers



```
Content-Type: text/html; charset=UTF-8
Content-Length: 122
Date: Sun, 21 Sep 2017 06:46:04 GMT
Server: squid/3.5.20
Connection: close
Set-Cookie:1P_JAR=2017-09-18-06; expires=Mon, 25-Sep-2017 06:46:04 GMT; path=/; dom
X-Secret-ID:AKIDXXXXXXXXX48pN
// `secret_id` in key pair
X-UsagePlan-ID:XXXXXXXX
// ID of the usage plan bound to the key pair
X-RateLimit-Limit:500
```



```
// Throttling configuration in the usage plan  
X-RateLimit-Used:100/125  
// Throttling usage in the usage plan
```

Authentication-Free API

Last updated : 2023-12-22 09:49:56

Overview

This document describes how to call an API that requires no authentication.

Prerequisites

Creating a no-auth API

1. In the [API Gateway console](#), create an API with **Authentication Type** set to **No authentication** (see [API Creation Overview](#)).
2. Publish the service to which the API belongs to the release environment (see [Service Release and Deactivation](#)).

Confirming information

Before calling the API, you must obtain information including the API's request path, request method, and request parameters, which can be found in the **Default Access Address** section of the API's details page.

The screenshot displays the 'Basic Information' tab of an API in the Tencent Cloud API Gateway console. The breadcrumb navigation shows 'service-b6mulwqw / api-17ood9c2'. The 'Basic Information' section includes fields for Path (/), Method (ANY), Apid (api-17ood9c2), and Creation Time (2020-09-11 10:22:26). The 'Frontend Configuration' section includes Frontend Method (HTTP), Path (/), Request Method (ANY), Authentication Type (No authentication), CORS is supported (No), and Notes (-). The 'Default Access Address' section on the right shows the Public Domain Name (http://service-b6mulwqw-1259347776.), Published Environments (release), Access Path (/), and Request Method (ANY). A note at the bottom of this section explains how to generate the default access address.

Basic Information	
Path	/
Method	ANY
Apid	api-17ood9c2
Creation Time	2020-09-11 10:22:26

Frontend Configuration	
Frontend Method	HTTP
Path	/
Request Method	ANY
Authentication Type	No authentication
CORS is supported	No
Notes	-

Default Access Address	
Public Domain Name	http://service-b6mulwqw-1259347776.
Published Environments	release
Access Path	/
Request Method	ANY

You can generate the default access address of the API in the form: `name/[release environment]/[access path]*`. You must include the port number when accessing the address. Otherwise, the address cannot be accessed.

Preparing tools

You can initiate requests from sources including browsers, browser plugins, Postman, and clients. Postman is recommended for simple validation.

Directions

1. Below is the basic information of an API.

Public domain name: `http://service-p52nqnd0-1253970226.gz.apigw.tencentcs.com`

Published environment: release

Access path: `/api`

Request method: GET

The default access address of an API is in the format of “public domain name or VPC domain name/published environment/access path”, so the default access address of the above API is: `http://service-p52nqnd0-1253970226.gz.apigw.tencentcs.com/release/api`.

2. Enter the access address in Postman, select `GET` as the request method, set the request parameters, and click **Send** to call the API.

The screenshot shows the Postman interface with a GET request configured. The URL is `http://service-p52nqnd0-1253970226.gz.apigw.tencentcs.com/release/api`. The 'Params' tab is selected, showing a table for Query Params.

KEY	VALUE	DESCRIPTION
Key	Value	Descriptor

Release and Access

Overview

Last updated : 2023-12-22 09:50:07

A created and configured API can be accessed only after it is published. Otherwise, it can only be used for internal trial. After an API is published, its public network access address will be generated. A user can access it via the bound user domain name.

Service is the basic unit for publishing. In each publishing, all the APIs in the service will be published to the specified environment and can be accessed from the public network. Three environments are supported: testing, pre-publishing and publishing.

Service Release and Deactivation

Last updated : 2023-12-22 09:51:46





Operation Scenarios

Once APIs are configured in a service, the service can be published. The system will use the system time as the release log to facilitate release rollback as needed.

Directions

Service release

1. Log in to the [API Gateway Console](#) and click **Service** on the left sidebar.
2. Select the name of the service to be published from the service list and click **Publish** in the "Operation" column.

Service Name	Service Status	Monitor	Default Domain Name ⓘ	Frontend Type	Release Environment and Status
service-3yj48pii exampleservice	✓ Running		Public Network  : service-3yj48pii-1259347776.gz.apigw.tencentcs.com	http	Test: Not publish Pre-publish: Not publish Publish: Not publish
service-jgf1sjai test	✓ Running		Public Network  : service-jgf1sjai-1259347776.gz.apigw.tencentcs.com	http	Test: Not publish Pre-publish: Not publish Publish: Not publish

3. Select the release environment and enter remarks.

Release environment: currently supported environments include test, pre-release, and release.

Remarks: this field can contain up to 200 characters, which is required.

4. Click **Submit**.




Service deactivation

Note :

After a service is deactivated, it cannot be accessed externally.

Unpublished services cannot be deactivated.

After a service is published to a specific environment, if you want to unpublish it, click **Deactivate** in the "Operation" column on the environment management page.

Environment Name	Access Path	Release Status	Running Version
▶ Test	service-3yj48pii-1259347776.gz.apigw.tencentcs.com/test 	Published	20200406141058dfa97eaa-482a-4f81-b0df-6e08b0ec
▶ Pre-publish	service-3yj48pii-1259347776.gz.apigw.tencentcs.com/prepub 	Not publish	-
▶ Publish	service-3yj48pii-1259347776.gz.apigw.tencentcs.com/release 	Not publish	-

Service Access

Last updated : 2024-01-12 10:46:04

Operation scenarios

After the service is successfully published, it can be tested by using the service's default subdomain name. If you want to publish the service for public access, please bind it to your own domain name for API access.

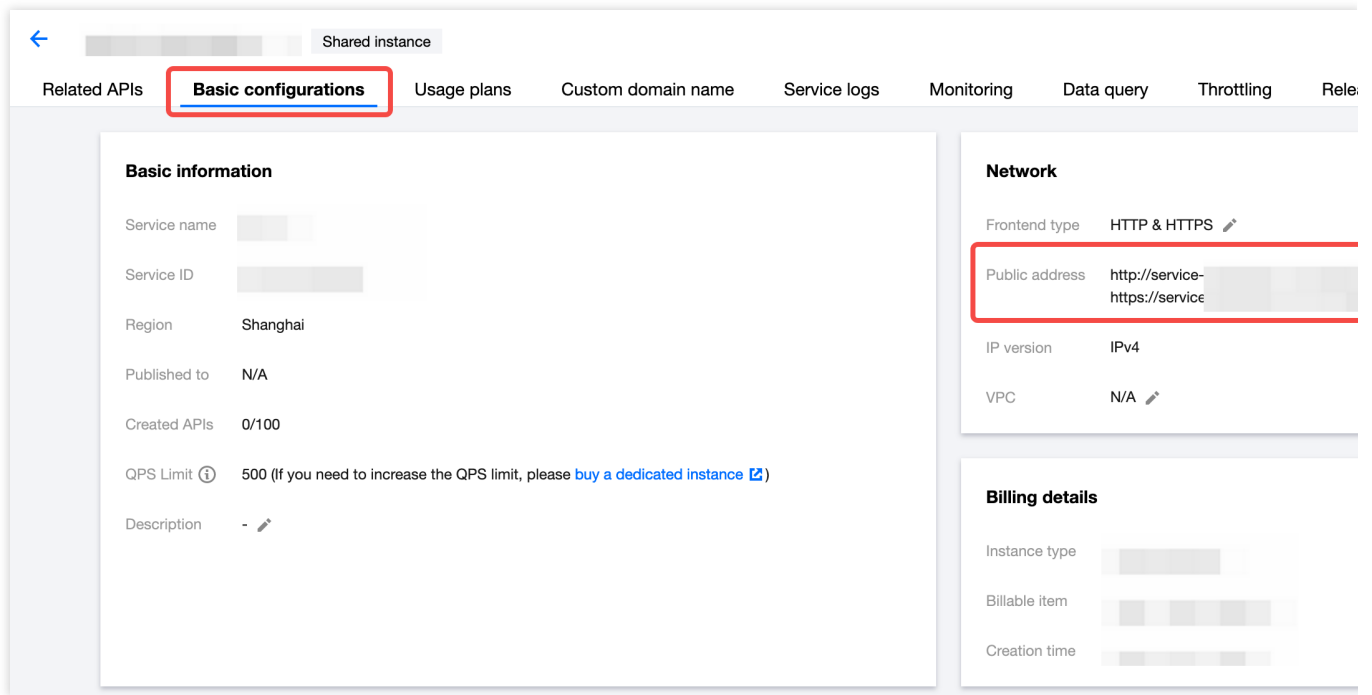
Note:

The default subdomain name provided by API Gateway is solely for integration testing and not recommended for use in your official production service. Once your testing is complete, we highly encourage you to utilise your own domain name. For more information, please refer to [Binding A Custom Domain Name](#). After the binding, your API paths can be accessed via the custom domain name.

Operation step

Default subdomain name rule of a service

The default subdomain name of the service appears automatically after a service is created, as shown in the figure below.



Each service in API Gateway provides a default subdomain name, containing the following rules:

First type: From **November 22, 2023**, a brand new subdomain name will be officially put into use

`http://{service-id}-{your-unique-id}.{region}.tencentapigw.com`

Second type: Users utilizing this domain name will gradually use the new domain name `tencentapigw.com`

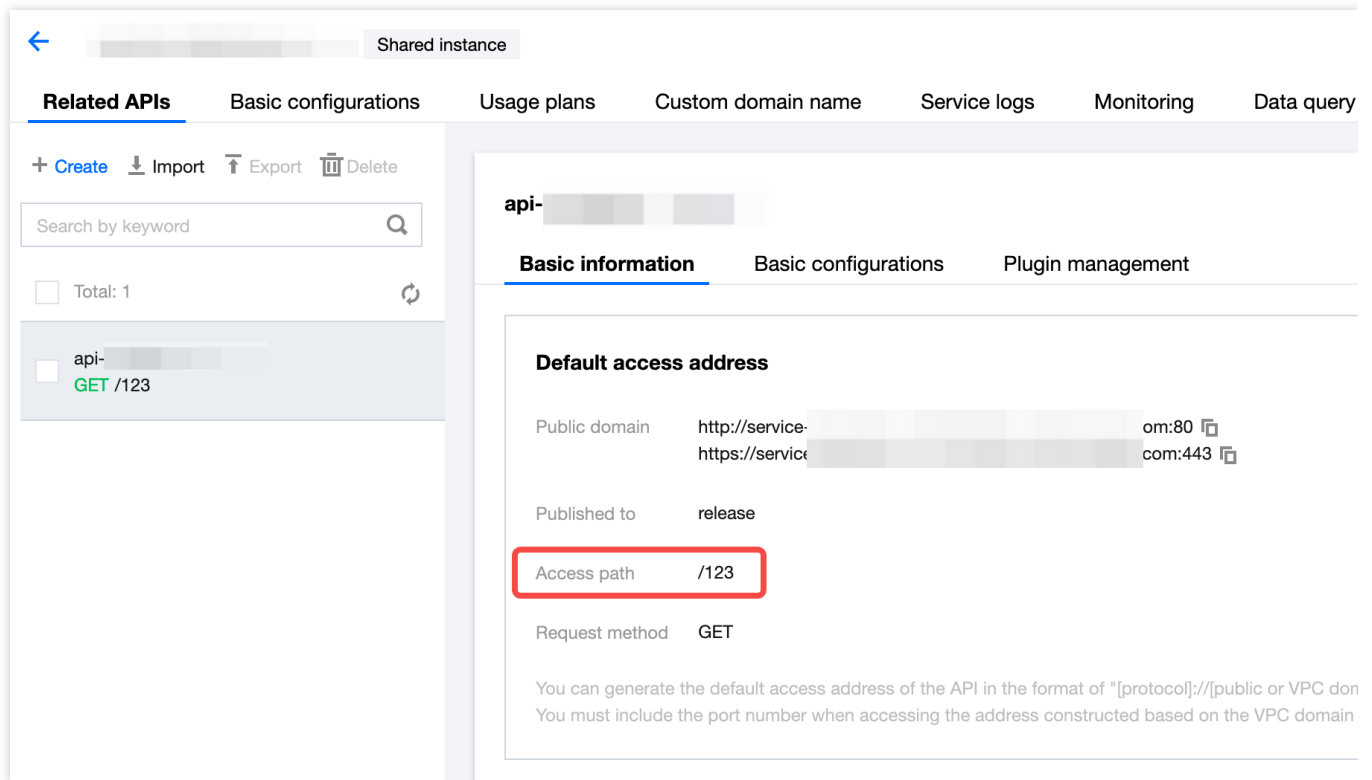
`http://{service-id}-{your-unique-id}.{region}.apigw.tencentcs.com`

Third type: service only appeared before 2021

`http://{your-unique-id}.{region}.apigateway.myqcloud.com`

API path access rules of a service

The API path is a custom-defined part of the front-end access path when a user creates an API. Once the API creation is complete, it can be viewed in the following positions.



The following section provides an example using the aforementioned first type of default sub-domain name:

After a service is published, the access path to the specific API is as follows

When there is only an environment for a service, the three supported environments are distinguished as follows:

Release Environment `http://{your-unique-id}.{region}.tencentapigw.com/release` .

Prerelease Environment `http://{your-unique-id}.{region}.tencentapigw.com/prepub` .

Test Environment `http://{your-unique-id}.{region}.tencentapigw.com/test` .

When there is an environment and API path for a service, an environment should be designated first, followed by the path:

For instance, to access the `/getUser` path in the release environment, use `http://{your-unique-id}.{region}.tencentapigw.com/release/getUser` .

Example

Your user ID is 123456789. You have created a service named `register` in Guangzhou (gz), which has a service ID of `service-n904iiau`. You have also added an API with a path of `/user` . The `register` service is now published on `3 different environments` . Under these circumstances, if other users, applications or terminals require access to the `/user` API, the accurate access path are as follows:

Release environment: `http://service-n904iiau-123456789.gz.tencentapigw.com/release/user` .

Pre-release environment: `http://service-n904iiau-123456789.gz.tencentapigw.com/prepub/user` .

Test environment: `http://service-n904iiau-123456789.gz.tencentapigw.com/test/user` .

Note

Due to the quality issues of international links, which are impacted by various factors and hard for the ISP to optimize and improve over a short period, there might be instances of packet loss or timeouts during cross-border access with API Gateway. If you mainly access overseas businesses, we suggest you respectively create resources domestically and overseas. The domestic services cover local businesses, and overseas services cover foreign businesses. So you can avoid congested cross-border network segments for a better business experience. We appreciate your understanding and support.

Environment Version Switch

Last updated : 2023-12-22 09:52:18

Operation Scenarios

API Gateway logs the version of each service published in an environment, allowing you to switch versions with ease.

Directions

Viewing a version

A release log contains the release time and release notes you enter. You can view the release log of any release on the release page.

1. Log in to the [API Gateway Console](#) and click **Service** on the left sidebar.
2. In the service list, click the target service name to enter the service details page.
3. At the top of the service details page, click **Manage Environment** to view the release history.

Service Info

Manage API

Usage Plan

Custom domain

Service Log

Manage Environment

Mange Version

Monitor

API Docu

As you embed the client or debug, please provide an accurate access path to avoid errors.

Environment Name	Access Path	Release Status	Running Version
▼ Test	service- <div> </div> gz.apigw.tencentcs.com/test	Published	2 <div> </div> 5d6553540-5069-4f9f-aecd-d8389c54

Published Version

Version ID	Release Time	Remarks
20200406141058dfa97eaa- <div> </div> 6e08b0ec7393	2020-04-06T06:10:58Z	123
20200406144035d6553540- <div> </div> -d8389c54d6b8(Current Version)	2020-04-06T06:40:35Z	test

Total items: 2

Switching versions

1. Select the environment for which you want to switch versions on the environment management page and click **Switch Version** in the "Operation" column.
2. Select the version you want to switch to and click **Submit** to switch the latest version in the current environment to another version.

Switch Version

SelectEnvironment: Test version to publish

Please select a version

Version	Remarks
<input type="radio"/> 20200406141058dfa97eaa-482a-4f81-b0df-6e08b0ec7393	123

Submit

Close

Custom Domain Name and Certificate

Configuring Custom Domain Name

Last updated : 2023-12-22 09:52:30

Scenarios

The domain name binding feature allows you to bind your own domain name to a service, so that the service can be accessed at it.

Prerequisite

Complete resolution of the domain name to be bound.

Directions

Associating a custom domain name

1. Log in to the [API Gateway console](#) and click **Service** on the left sidebar.
2. In the service list, click a service ID to enter the service details page.
3. On the service details page, select **Custom domain**, click **Create** in the top-left corner, enter the configuration information, and click **Submit**.

Note:

If you need the HTTPS protocol that supports independent domain names, submit the SSL certificate for your domain name. You can upload the certificate or enter the certificate name, content, and private key.

After a CNAME record is modified, it takes a while for it to take effect. Be sure to proceed with the configuration after it takes effect; otherwise, the configuration will fail.

If you select HTTPS for a independent domain, WSS is also supported by default.


4. After configuring CNAME resolution, configure the domain name in the service (be sure to configure CNAME resolution first).
5. To unbind a domain name, delete it from the service and then delete its CNAME record.

Configuring forced HTTPS

On the custom domain name configuration page, if the protocol is HTTPS&HTTPS or HTTPS, the forced HTTPS feature can be enabled. After it is enabled, API Gateway will redirect requests using the custom domain name over the


HTTP protocol to the HTTPS protocol.

Add Custom Domain Name

i Please make sure that the domain name to bind has been resolved and configured, and has pointed to the second-level service. You should confirm that your domain name has been ICP licensed according to [Domain Name Registration Re](#)
Public Second-Level Domain: service-b6mulwqw-1259347776.gz.apigw.tencentcs.com 

Service **serverless**

Domain

After modifying the CNAME record, it needs to be greater than the TTL time to take effect. Please make sure to
Please see [official documentation](#) 

Access Mode ☒ Public Network

Protocol

HTTP & HTTPS

HTTP

HTTPS

Certificate

Please select 

If there is no applicable certificate, please upload it on the console. [Click to Upload](#) 

Force HTTPS

☐

After you check this option, API Gateway will redirect HTTP requests of the custom domain name to HTTPS.

Path Mapping

☒ Default path mapping 

☐ Custom path mapping

Submit

Close

Configuring domain name path mapping

1. On the custom domain name list page, click **Edit Path Mapping** in the **Operation** column.

Create					
Domain Name	Path Mapping	Protocol	Network Type	Resolution Status	SS
www.wuyizhan.xyz	Use default path mapping ⓘ	http	Public Network	Resolved successful	No
Total items: 1					

2. Select the path mapping type:

Default path mapping: the URL of the path is `custom domain name/environment name`. For example, `www.XXXXXX.com/release` points to the release environment in the service, `www.XXXXXX.com/prepub` points to the pre-release environment in the service, and `www.XXXXXX.com/test` points to the test environment in the service.

Add Custom Domain Name

① Please make sure that the domain name to bind has been resolved and configured, and has pointed to the second-level domain of the service.
Public Second-Level Domain: `service-3yj48pii-1259347776.gz.apigw.tencentcs.com`

Service `exampleservice`

Domain Name

Access Mode ☒ Public Network

Protocol

Path Mapping ☒ Default path mapping ⓘ ☐ Custom path mapping

Custom path mapping: the URL is `custom domain name/custom path`. This URL points to the environment that you map. For example, if the configured path is `/mypath` and the environment is the release environment, the URL of the release environment will be `www.XXXXXX.com/mypath`. If you want to use the root path, directly configure the path as `/`.

Add Custom Domain Name

① Please make sure that the domain name to bind has been resolved and configured, and has pointed to the second-level domain of the service.
Public Second-Level Domain: service-3yj48pii-1259347776.gz.apigw.tencentcs.com

Service exampleservice

Domain Name

Access Mode ☒ Public Network

Protocol

http

Path Mapping ☒ Default path mapping ⓘ ☐ Custom path mapping

Submit

Close

N:

When a custom path mapping is used, the default path mapping (`custom domain name/environment name`) will not take effect.

Both the custom path mapping and default path mapping can be modified after configuration.

3. Click **Submit** to complete the configuration.

Usage Plan

Overview

Last updated : 2023-12-22 09:52:59

API Gateway uses usage plans to control the authentication, request traffic, quotas, and API user permissions after a service is published.

Currently, the content that can be adjusted and controlled by a usage plan includes:

Authentication and security control

Traffic control

A usage plan will take effect after it is bound to a service environment.

Multiple usage plans can be bound to the same service environment, and one usage plan can be bound to multiple service environments.

Note :

Two or more usage plans bound to the same key cannot be bound to the same environment in the same service.

Similarly, if two or more usage plans have been bound to the same environment in the same service, they cannot be bound to the same key.

Working with a Usage Plan

Last updated : 2023-12-22 09:53:14

Scenarios

To call a service after it is published, you need to create a key-value pair secret and a usage plan and bind them to the service environment.

This document describes how to configure a user-specific usage plan and make it available to the users.

Prerequisite

1. [Create a service](#) and [create and debug an API](#).
2. [Publish the service](#) to an environment, such as the release environment.

Directions

Creating a key-value pair secret

1. Log in to the [API Gateway console](#), and click **Application** on the left sidebar to enter the application management page.
2. On the application management page, click **Secret** on the top to open the secret management page.
3. Click **Create**, select the secret type and complete the following information.

Auto-generated

Custom secret

Enter the secret name.

Secret name: Up to 50 characters ([a-z], [A-Z], [0-9] and [_])

Enter the secret name, SecretId and SecretKey.

Secret name: Up to 50 characters ([a-z], [A-Z], [0-9] and [_-])

SecretId: 5-50 characters ([a-z], [A-Z], [0-9] and [_-])

SecretKey: 10-50 characters ([a-z], [A-Z], [0-9] and [_-])

4. Click **Submit** to generate a secret or save the custom key-value pair secret (SecretId and SecretKey).

Bind Key

Please select a secret key

Please enter keywords

☒ Secret Key N... Secret Key

☒

Support for holding shift key down for multiple selection

Submit

Close

(1) selected

Secret Key ...	Secret Key
<div></div>	<div></div>

Creating a usage plan

1. On the left sidebar, click [Usage Plan](#) to enter the usage plan list page.
2. Click **Create** in the top-left corner and enter the configuration information as prompted.
3. Click **Submit** to complete the creation.

Binding usage plan to secret

1. On the [Usage Plan](#) page, click the ID of the target usage plan to enter the usage plan details page.
2. On the usage plan details page, click **Bind Key**.
3. Check the `SecretId` to be bound and click **Submit** to complete the binding.

Bind Key

Please select a secret key

☒ Secret Key N... Secret Key

☒

Support for holding shift key down for multiple selection

(1) selected

Secret Key ...	Secret Key

Binding a usage plan with a service environment

1. Select a created service on the [Service](#) list page, switch to the **Usage Plan** tab, and click **Bind**.

←

Service Info Manage API **Usage Plan** Custom domain Service Log Manage Environment Mange Version Monitor API Documentation/SDK Policy

By environment By API

Usage Plan Name	Environment	Max Calls per Second	Quota Details	Creati
No data yet				

Total items: 0

2. In the usage plan binding window, select an effective environment and usage plan to be bound.
Effective environments: publish, pre-publish, and testing

3. Click **Submit** to complete the binding.

Bind Usage Plan (by environment)

Region

Guangzhou

Effective Environment

Test

Usage Plan

Please select usage plan

(0) selected

Please enter usage plan name/ID

☐ Name/ID

☐ PlanA

Name/ID

No data yet

Support for holding shift key down for multiple selection

Submit

Close

Note:

Usage plans bound with the same key-value pair cannot be bound to the same environment.

Now you can share the SecretId and SecretKey to the end user. The end user can use the provided `SecretId` and `SecretKey` through the second-level domain name of the service (or a bound private domain name) to access the APIs published in the service.

Traffic Control

Last updated : 2023-12-22 09:53:31

You can limit the maximum number of calls under a usage plan by setting the QPS in it and binding a key.

For example, if you create a `secret_id` and `secret_key` pair and a usage plan with 1,000 QPS, bind the `secret_id` and `secret_key` pair to the usage plan, and bind the usage plan to the environment where you need to limit the traffic, such as the release environment, then an API in the release environment can be called by a user with the `secret_id` and `secret_key` pair at a frequency of up to 1,000 QPS.

Currently, up to 2,000 QPS can be set for each usage plan. Because the architecture of API Gateway is designed to be highly available, forwarded requests will be processed by different underlying nodes. If the traffic control value is too small (such as less than 5 QPS), there will be a certain probability that traffic control will be inaccurate, and the actual number of requests allowed to pass will be slightly greater than the set value.

Backend Upstream

VPC Upstream

Last updated : 2023-12-22 09:53:41

Overview

A VPC upstream is to open the services deployed in the VPC for external access through API Gateway. It provides load balancing capabilities and can connect to multiple services in the VPC at the same time.

Solution Strengths

There is no need to purchase private network CLBs. With VPC upstreams, you can use API Gateway alone to open services in the VPC for external access. The VPC upstream feature itself is free of charge, resulting in low costs. VPC subnets are used for communication, resulting in low network latency and high performance.

Prerequisites

You have purchased a [dedicated API Gateway instance](#).

You have created VPC resources such as [CVMs](#).

Note:

Currently, VPC upstreams support only APIs mounted under dedicated instances but not shared instances.

Directions

Step 1. Create a VPC upstream

1. Log in to the [API Gateway console](#).
2. On the left sidebar, click **VPC Upstreams**. The VPC upstream list page is displayed.
3. Click **Create** in the top-left corner of the page to create a VPC upstream.

Parameter	Required	Description
Upstream Name	No	Name of the VPC upstream. An upstream name can contain up to 50 characters out of a-z, A-Z, 0-9, and _.
VPC	Yes	VPC of the backend resource to be connected.

Description	No	Remarks of the VPC upstream.
Load Balancing Algorithm	No	Only the weighted polling algorithm is supported currently.
Protocol	Yes	Protocol for API Gateway to interact with backend resources. HTTP and HTTPS are supported.
Host Header	No	Request header Host of backend requests, usually the backend service domain name and port number.
Node List	Yes	List of backend nodes to which API Gateway forwards messages. A list can contain up to 200 nodes. You need to set the node address, port number, and weight for each node.
Retry Attempts	Yes	Number of retry attempts for a failed node request of API Gateway. The default value is 5. Enter an integer ranging from 1 to 100.

[←](#) **Create VPC Upstream**

Basic Info

Upstream Name

Optional

Up to 50 chars, supporting a-z, A-Z, 0-9, and underscores.

Select VPC

forrester(vpc-dbxu830k)

Description

Please enter description

Load Balancer

Load Balancing Algorithm

Weighted Round Robin

Protocol

HTTP

HTTPS

Host Header

Specifies the host request header for backend requests. It's usually the backend service domain name and port

Node List

No.	Node Address	Port
1	<div>Exiting CVM</div> <div>Please select</div>	

Add Node (1/200)

Retry Attempts

5

times

Please enter a positive integer between 1 to 100. Note that if the maximum number of retry attempts is set too h

Save

Cancel

Step 2. Create an API for the backend to connect to resources in the VPC and associate it with the VPC upstream

1. Log in to the [API Gateway console](#) and click **Service** on the left sidebar.
2. In the service list, click a service name **mounted under a dedicated instance** to view the service details.
3. In the service details, click the **Manage API** tab and choose to create a **general API** based on the backend business type.
4. Click **Create**, enter the API frontend configuration, and click **Next**.
5. In **Backend Configuration**, set **Backend Type** to **VPC resources**, set **Connection Mode** to **VPC Upstream**, and select the VPC upstream created in step 1.

6. Complete the rest configuration.

Step 3. Call the API

Call the API created in step 2. The API can be called successfully.

Notes

On the **Associated APIs** tab on the VPC upstream details page, you can view the APIs that use the current VPC channel as the backend. You need to delete these APIs before deleting the VPC upstream to avoid affecting API calling.

A dedicated instance runs in a VPC. If the VPC where the dedicated instance resides is inconsistent with the VPC connected to the VPC upstream, connect the two VPCs via [Cloud Connect Network](#) to avoid affecting API calling.

Verification and Security

Overview

Last updated : 2023-12-22 09:53:53

Various API verification methods and protection policies are provided to protect your APIs, helping avoid data and asset losses caused by malicious access, unauthorized access, application vulnerabilities, and attacks.

Currently, API Gateway supports [application-enabled](#), [OAuth2.0](#), and [key pair](#) authentication methods. The key pair method is a historical feature, and the application-enabled method is recommended.

Application-Enabled Authentication Method

Last updated : 2023-12-22 09:54:07

If a published API uses the application-enabled authentication method (`ApiAppKey` and `ApiAppSecret`), when a client calls the API, it needs to use the signature key to perform signature calculation on the request content and transfer the signature to the server for signature verification. This document describes how to implement the signature calculation process on the client.

Note:

For application-enabled authentication signature demos for common programming languages, see [Development Guide - Generating Application Authentication Signature](#).

Overview

API Gateway provides a frontend signature calculation and verification feature, which can:

- Verify the validity of a client request and confirm that the request carries the signature generated by the authorized `ApiAppKey` .

- Prevent the request data from being tampered with during network transfer.

The owner of an API can generate an application on the application management page in the API Gateway console. Each application carries a signature key pair (`ApiAppKey` and `ApiAppSecret`). After the API owner authorizes the API to the specified application (which can be issued by the API owner or owned by an API caller), the API caller can use the application's signature key to call the API.

When the client calls the API, it needs to use the authorized signature key to perform encrypted signature calculation on the critical request content, put the `ApiAppKey` and the encrypted signature string in the header of the request, and transfer it to API Gateway. API Gateway will read the header information of the `ApiAppKey` in the request, query the value of the `ApiAppSecret` corresponding to the value of the `ApiAppKey` , use the `ApiAppSecret` to perform signature calculation on the critical data in the received request, and compare the generated signature with the signature sent by the client to verify the correctness of the signature. Only if the request passes the signature verification will it be sent to the backend service; otherwise, API Gateway will deem the request invalid and directly return an error.

Signature Generation and Verification Process

Prerequisites

The security authentication type of the called API is "application authentication".

The API caller needs to get the authorization granted to the application by the API before calling the API.

Signature generation on client

1. Extract the critical data from the original request and get a string for signing.
2. Use the encryption algorithm plus `ApiAppSecret` to encrypt the critical data signature string to get a signature.
3. Add all headers related to the signature to the original HTTP request to get the final HTTP request.

Signature verification on server

1. Extract the critical data from the received request and get a string for signing.
2. Read the `ApiAppKey` from the received request and query the corresponding `ApiAppSecret` through the `ApiAppKey`.
3. Use the encryption algorithm plus `ApiAppSecret` to encrypt the critical data signature string to get a signature.
4. Read the client signature from the received request and check whether the server signature and the client signature are the same.

Signature Generation and Transfer

Signature string extraction

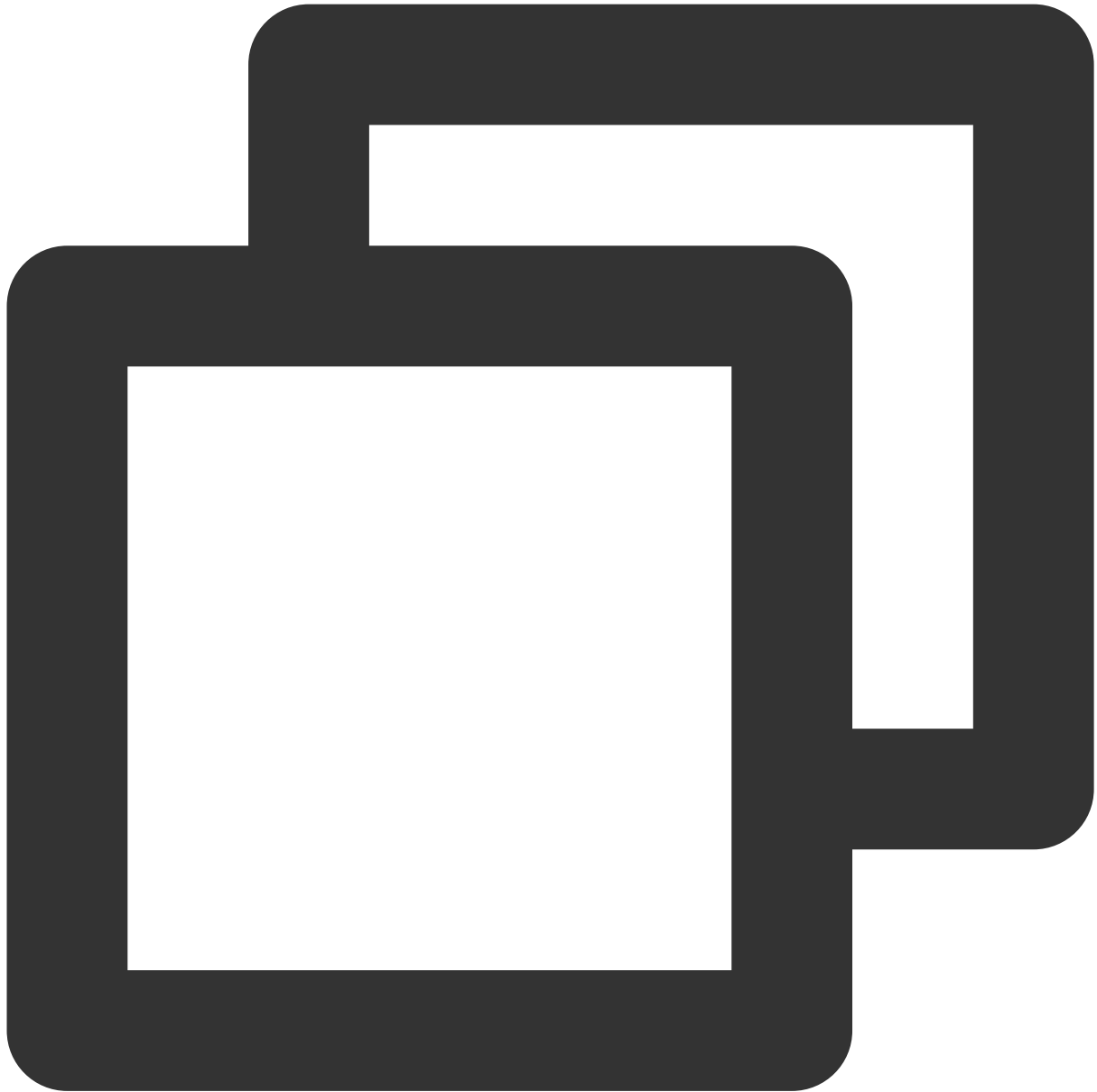
The client needs to extract the critical data from the HTTP request and splice it into a signature string in the following format:



```
Headers
HTTPMethod
Accept
Content-Type
Content-MD5
PathAndParameters
```

The above 6 fields constitute the entire signature string. They need to be separated with `\\n`. `Headers` must contain `X-Date`. There is no need to add `\\n` after `PathAndParameters`. Even if other fields are empty, `\\n` should still be retained. The signature is case sensitive. The extraction rules for each field are as follows:

Headers: you can select specified headers to participate in the signature calculation. The keys of the selected headers are sorted in lexicographical order and then spliced as follows:



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

The headers in `Authorization` are the ones involved in signature calculation. We recommend you convert them to the lowercase and separate them by ASCII spaces. For example, if the headers involved in the calculation are

`date` and `source` , the format should be `headers="date source"` ; if only the `x-date` header participates in the calculation, the format should be `headers="x-date"` .

HTTPMethod: HTTP method. The value should be in all caps (such as POST).

Accept: value of the `Accept` header in the request, which can be empty. We recommend you explicitly set the `Accept` header. If it is empty, some HTTP clients will set the default value of `/` for it, causing signature verification to fail.

Content-Type: value of the `Content-Type` header in the request, which can be empty.

Content-MD5: value of the `Content-MD5` header in the request, which can be empty. The `Content-MD5` header is calculated only when the request has a `Body` in a `non-Form` format. The calculation method of the `Content-MD5` value in Java is as follows:



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

`PathAndParameters`: it contains all the parameters in `Path` , `Query` , and `Form` in the following format:

`path` does not contain release environment (release, prepub, test) information.

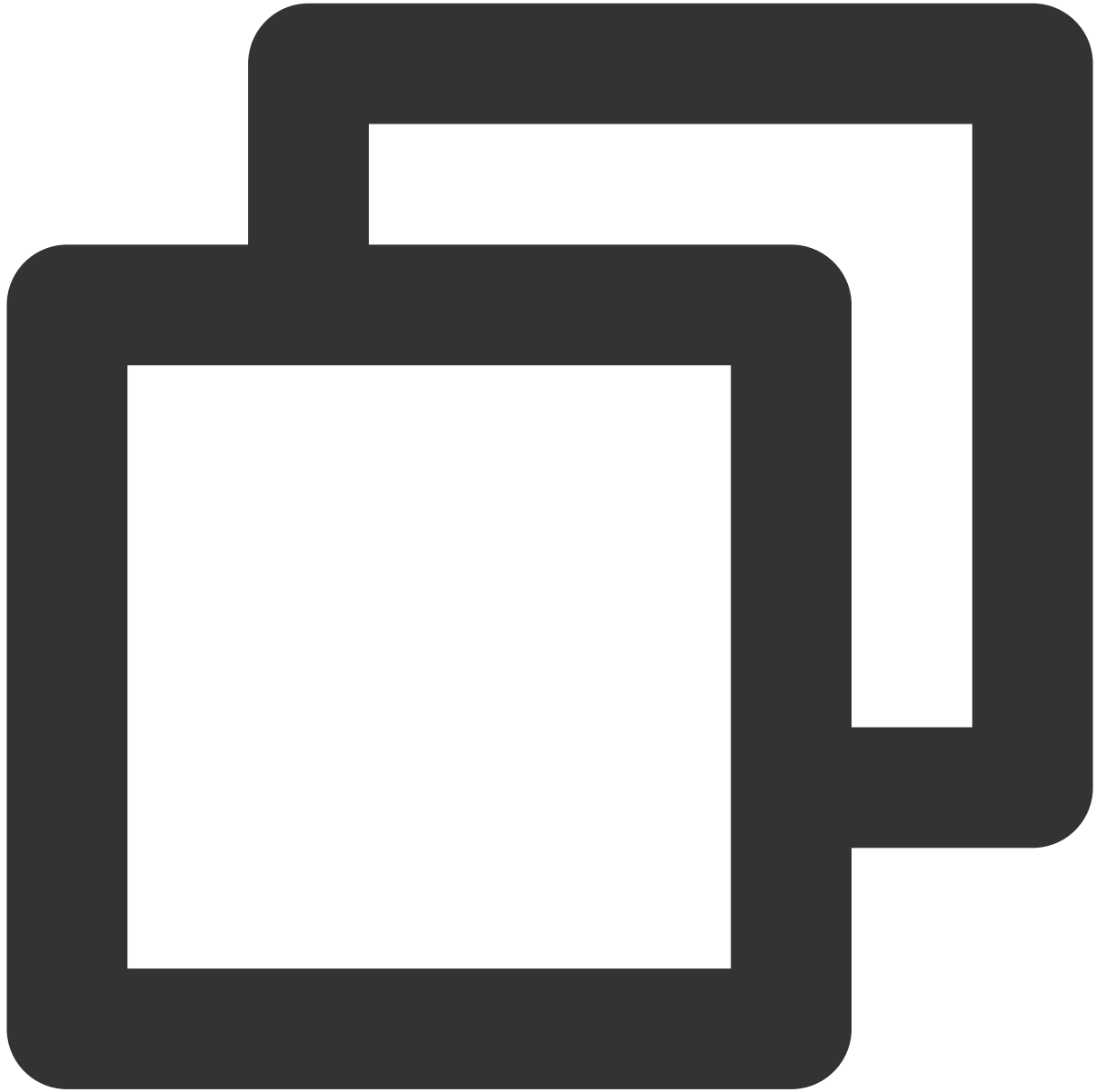
The keys of the `Query` and `Form` parameter pair are sorted in lexicographical order and then spliced in the above-mentioned method.

If `Query` and `Form` parameters are empty, use `Path` directly without adding `?` .

If the value of a parameter is empty, only the key is kept to participate in the signature calculation, and the equal sign does not need to be added in the signature.

If there are array parameters in `Query` and `Form` (i.e., parameters with the same key but different values), the values need to be sorted in lexicographical order and then spliced in the above-mentioned method.

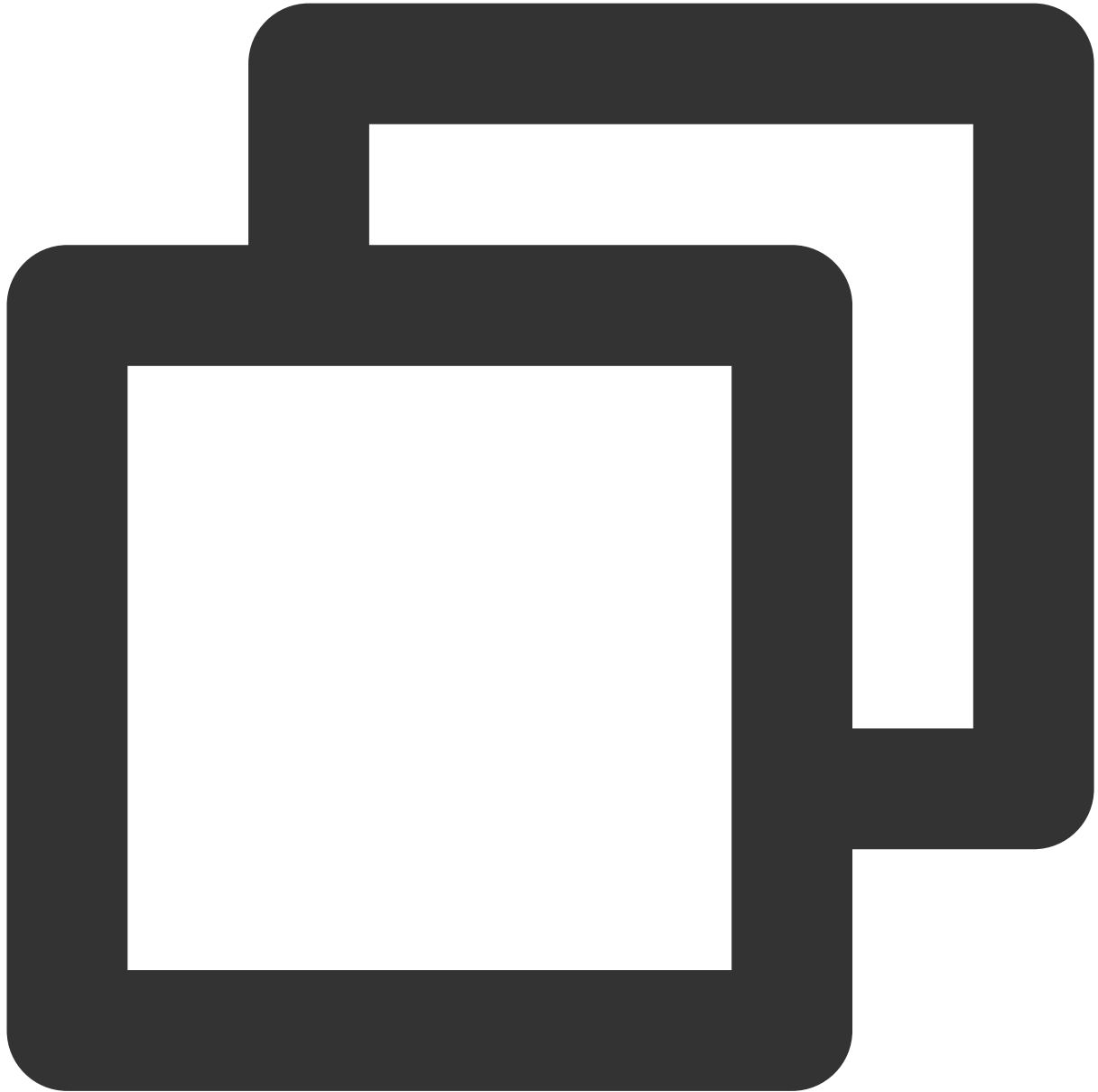
Take a general HTTP request as an example:



```
POST / HTTP/1.1
host:service-3rmwxxxx-1255968888.cq.apigw.tencentcs.com
accept:application/json
content-type:application/x-www-form-urlencoded
source:apigw test
x-date:Thu, 11 Mar 2021 08:29:58 GMT
content-length:8
```

```
p=test
```

The generated correct signature string is as follows:



```
source: apigw test
x-date: Thu, 11 Mar 2021 08:29:58 GMT
POST
application/json
application/x-www-form-urlencoded

/?p=test
```

Signature calculation

After the client extracts the critical data from the HTTP request and splices it into a signature string, it needs to encrypt and encode the signature string to form the final signature in the following steps:

1. Use UTF-8 to decode the signature string (`signing_str` signing information) to get a byte array.
2. Use the encryption algorithm to encrypt the byte array.
3. Base64-encode the encrypted byte array to form the final signature.

Signature transfer

The client needs to put the `Authorization` in the HTTP request and transfer it to API Gateway for signature verification.

The format of the `Authorization` header is as follows:



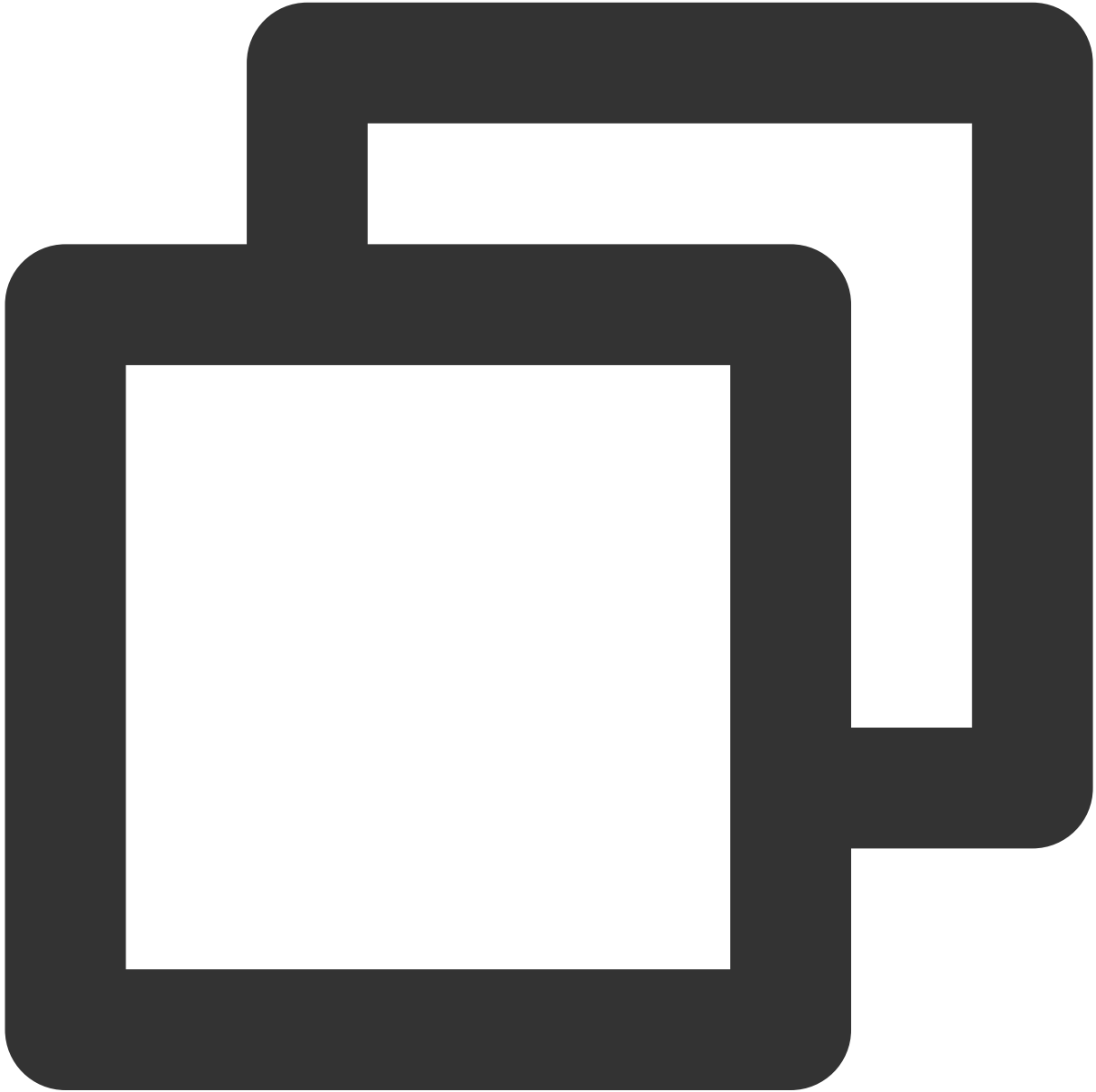
```
Authorization: hmac id="secret_id", algorithm="hmac-sha1", headers="date source", s
```

The parameters in `Authorization` are described as follows:

Parameter	Description
hmac	Fixed and used to indicate the calculation method
ID	Value of the <code>secret_id</code> in the key
algorithm	Encryption algorithm. HMAC-SHA1 and HMAC-SHA256 are supported currently

headers	Headers involved in the signature calculation
signature	Signature obtained after signature calculation is completed, with <code>signing_str</code> as its content

Below is an example of an HTTP request with a signature:



```
POST / HTTP/1.1
host:service-3rmwxxxx-1255968888.cq.apigw.tencentcs.com
accept:application/json
content-type:application/x-www-form-urlencoded
source:apigw test
```

```
x-date:Thu, 11 Mar 2021 08:29:58 GMT
Authorization:hmac id="xxxxxxx", algorithm="hmac-sha1", headers="source x-date", si
content-length:8
p=test
```

Signature Troubleshooting

Question:

If the API Gateway signature verification fails, the server's signature string (StringToSign) will be put in the HTTP response header and returned to the client with an error code of 401.

Solution:

1. Check whether the locally calculated signature string (StringToSign) is the same as that returned by the server.
2. Check whether the `ApiAppSecret` used for signature calculation is correct.

Since line breaks cannot be expressed in HTTP headers, line breaks in `StringToSign` are replaced with `#`.



```
"message": "HMAC signature does not match, Server StringToSign:source: apigw test#x-
```

This means that the server signature is:



```
source: apigw test
x-date: Thu, 11 Mar 2021 08:29:58 GMT
POST
application/json
application/x-www-form-urlencoded

/?p=test
```

Authentication-Free Mode

Last updated : 2023-12-22 09:54:18

You can select "Authentication-Free" for your API when creating it.

If "Authentication-Free" is checked, authentication will succeed and the bound usage plan will take effect when API Gateway receives an anonymous request.

If signature authentication is performed with the key in the usage plan, the traffic limit in the usage plan will take effect. In case of access by an anonymous user, the maximum traffic limit imposed on each API by Tencent Cloud will take effect.

OAuth2.0

Last updated : 2023-12-22 09:54:29

Overview

This topic describes how to configure OAuth 2.0 authorization access for APIs in the API Gateway console to meet your personalized security setting needs.

OAuth 2.0 Overview

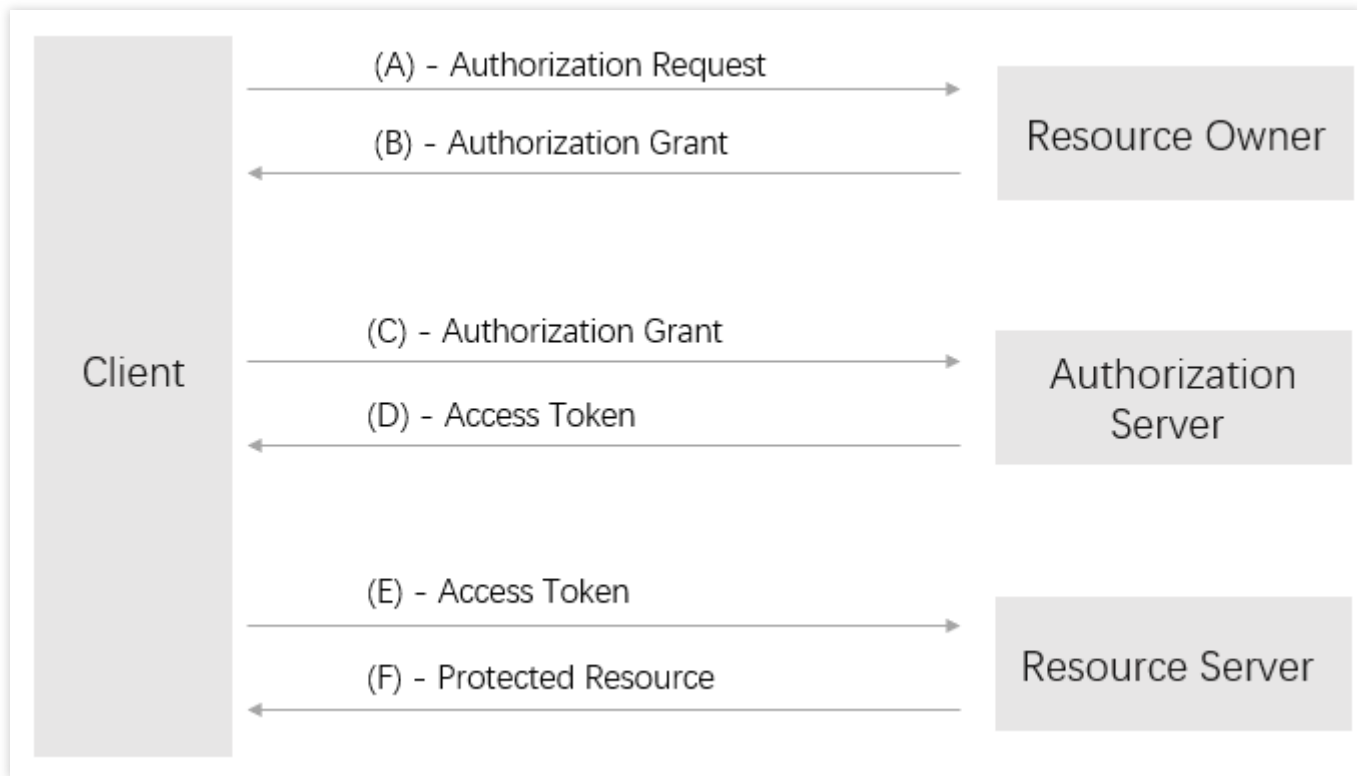
OAuth 2.0 is an open authorization standard that enables you to allow **third-party applications** to access your **specific private resources** in a service without **providing the account and password** to the applications. OAuth 2.0 is an authorization protocol rather than an authentication protocol.

OAuth 2.0 roles

OAuth 2.0 has the following 4 roles:

Role	Description
Resource owner	Owner of the resource
Resource server	Server where the resource is stored
Client	Third-party application client, which can be any third-party application that can consume the resource server
Authorization server	Intermediate layer that manages the above 3 roles

OAuth 2.0 authorization process



(A) The client initiates a request to the resource owner for authorization.

(B) The resource owner approves the authorization.

(C) The client applies to the authorization server for an authorization token after getting the resource owner's authorization.

(D) The authorization server grants the authorization token after authenticating the client.

(E) The client requests the resource server to send the user information after getting the authorization token.

(F) The resource server sends the user information to the client after verifying that the token is correct.

Prerequisites

An authorization server for distributing tokens is available. (You need to build an authorization server. The API Gateway provides the [Python3 Demo](#) and the [Golang Demo](#) for your reference.)

You have created an API Gateway service (for more information, see [Creating Services](#)).

Directions

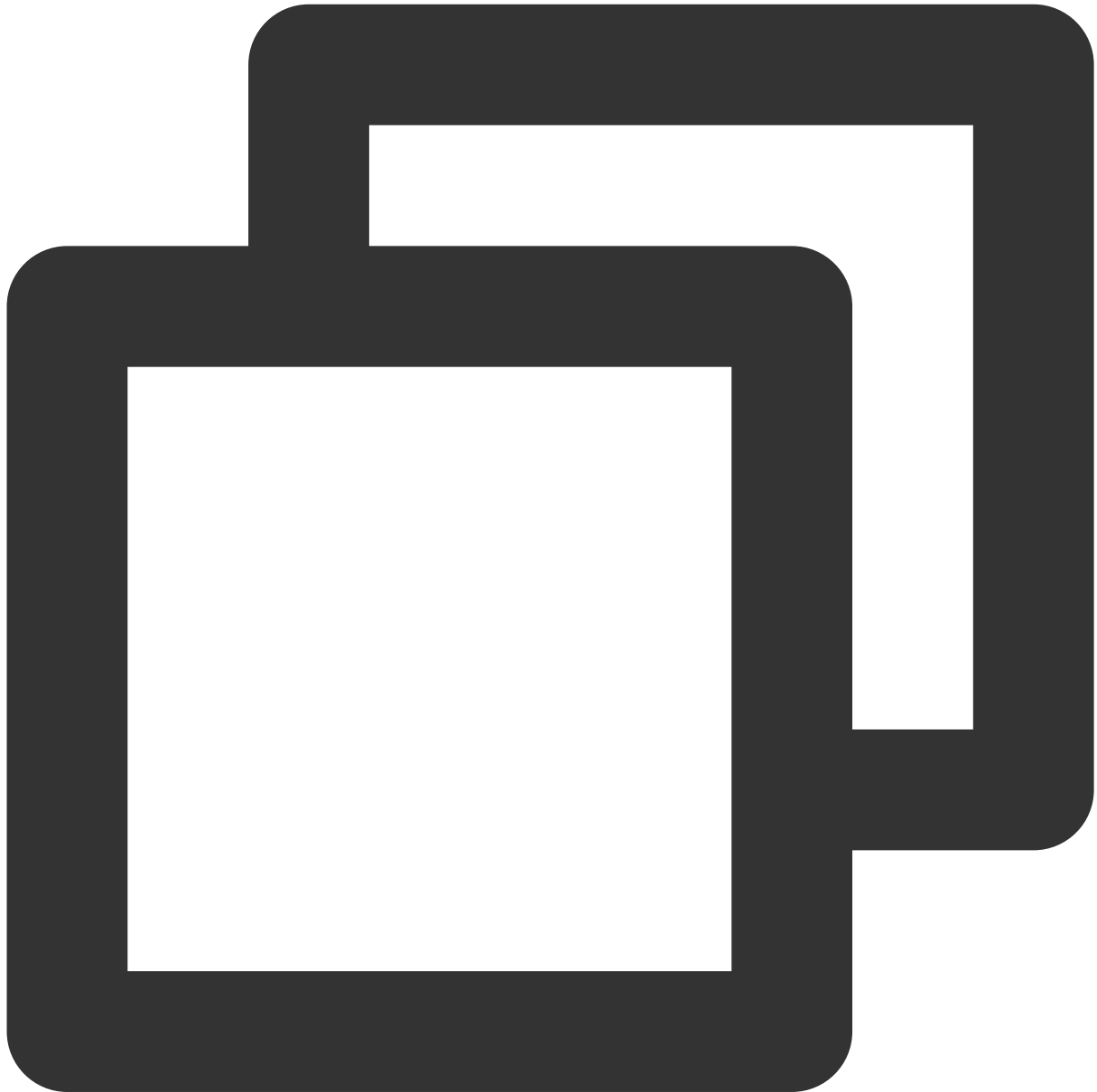
Step 1: build an authorization server (Python3 Demo is used as an example).

1. Download the [Python3 Demo](#) from the official repository of the API Gateway.
2. Generate the RSA public and private keys and run `produce_key.py` in Python 3 to generate 3 files:

public_pem: public key in PEM format.

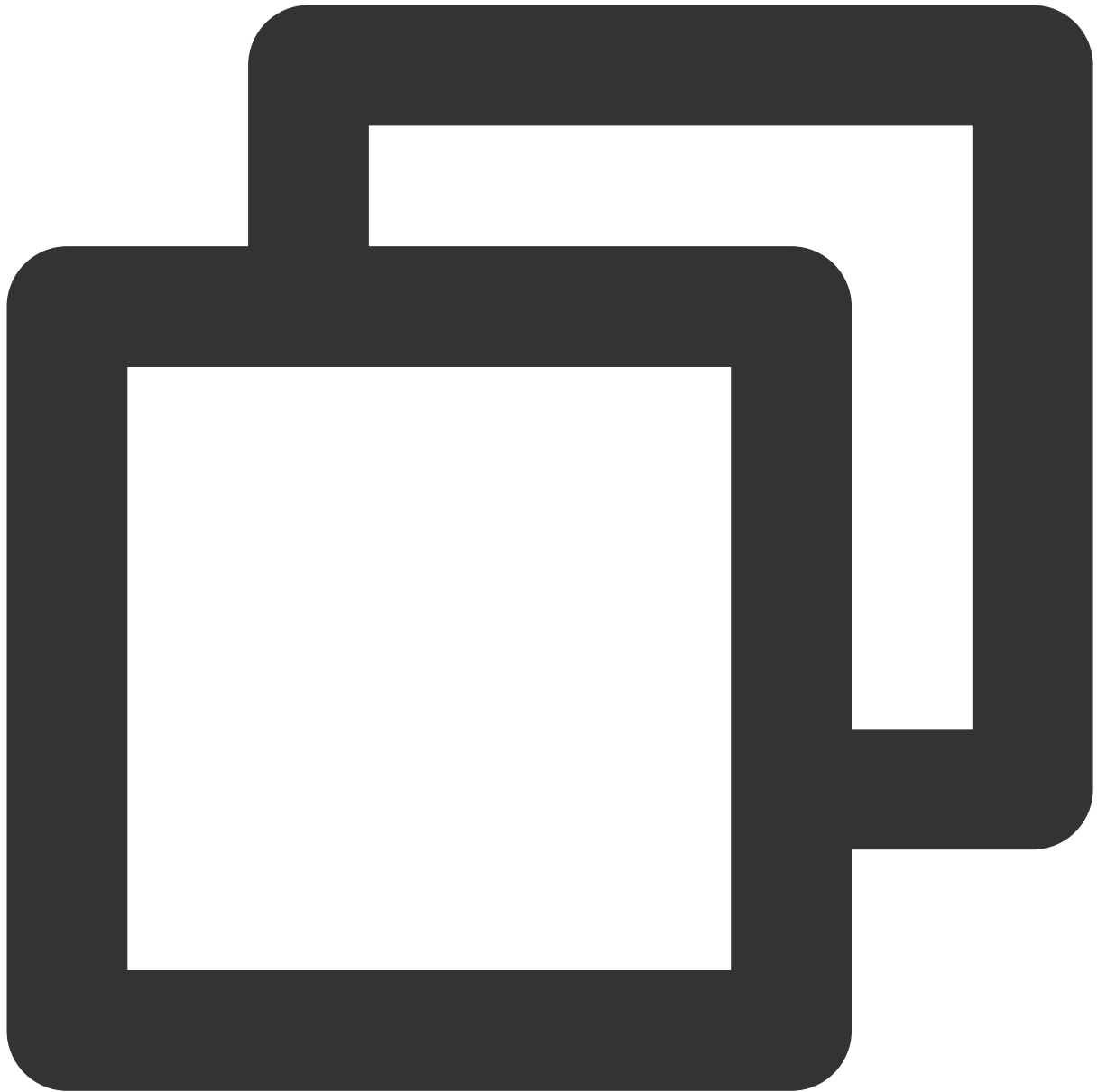
priv_pem: private key in PEM format.

public: public key in JSON format. The file content is used to configure the authorization API of API Gateway and is in the following format:



```
{"e": "AQAB", "kty": "RSA", "n": "43nSuC6lmGLogEPgFVwaaxAmPDzmZcocRB4Jed_dHc-sV7rcAcNB0i
```

3. Start the service. After installing the `bottle` library by running `pip3 install bottle`, run `server.py` in Python 3 to generate a token. Then, you can simply check whether the token is successfully generated.



```
curl localhost:8080/token
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1OTIyNzgwODksImZvbyI6ImJhciIsIm1hdC
```

Step 2: configure a Tencent Cloud API Gateway authorization API.

1. In the created service, create an authorization API. When you are configuring the frontend, select **OAuth 2.0** as the authentication type and **Authorization API** as the OAuth mode.

1 Frontend Configuration

2 Backend Configuration

3 Response Result

Service

testapigateway

API Name

outhapi

Up to 60 chars

Frontend Type

http

Path

1、 It starts with "/", "=", or "^~/", and supports uppercase and lowercase letters, digits, and \$-_.+!*()/%.
2、 The request parameter of Path type must be enclosed with {}, and should be an independent component of the path (s

Request Method

GET

Authentication Type

OAuth 2.0

OAuth Mode

Authorization API

CORS is supported

☐

Remarks

Please enter remarks

Parameter Configuration

Parameter Name	Parameter Location ⓘ	Type	Default Value ⓘ
Newly added parameter configuration (0/30)			

2. When you are configuring the backend, select your own server address as the authentication server, select **Header** as the token location, and enter the content in the `public` file generated by running `produce_key.py` as the public key. After the API is created, click **Complete**.

✓ Frontend Configuration

>

2 Backend Configuration

>

3 Response Result

Backend Type

OAuth 2.0

VPC Info

Disable VPC

Backend Domain Name ⓘ

http://

8000

Backend Path

Request Method

GET

Token Location

Header

Redirect Address

Backend timeout ⓘ

15

Public Key

It starts with http or https and contains domain content, and "/" is not required at the end

1、 It starts with "/", and supports uppercase and lowercase letters, digits, and \$-.,+!*()/%.

2、 "=" and "^~" in the frontend parameters are used for exact match to the frontend path, which are not available to the backend.

3、 The request parameter of Path type must be enclosed with {}, and should be an independent component of the path (such as {id}).

Optional. You may enter the address to redirect to when a bound service API is called without authorization. But the redirect is in the form of http://ip:port/path.

Time range: 1-1,800s

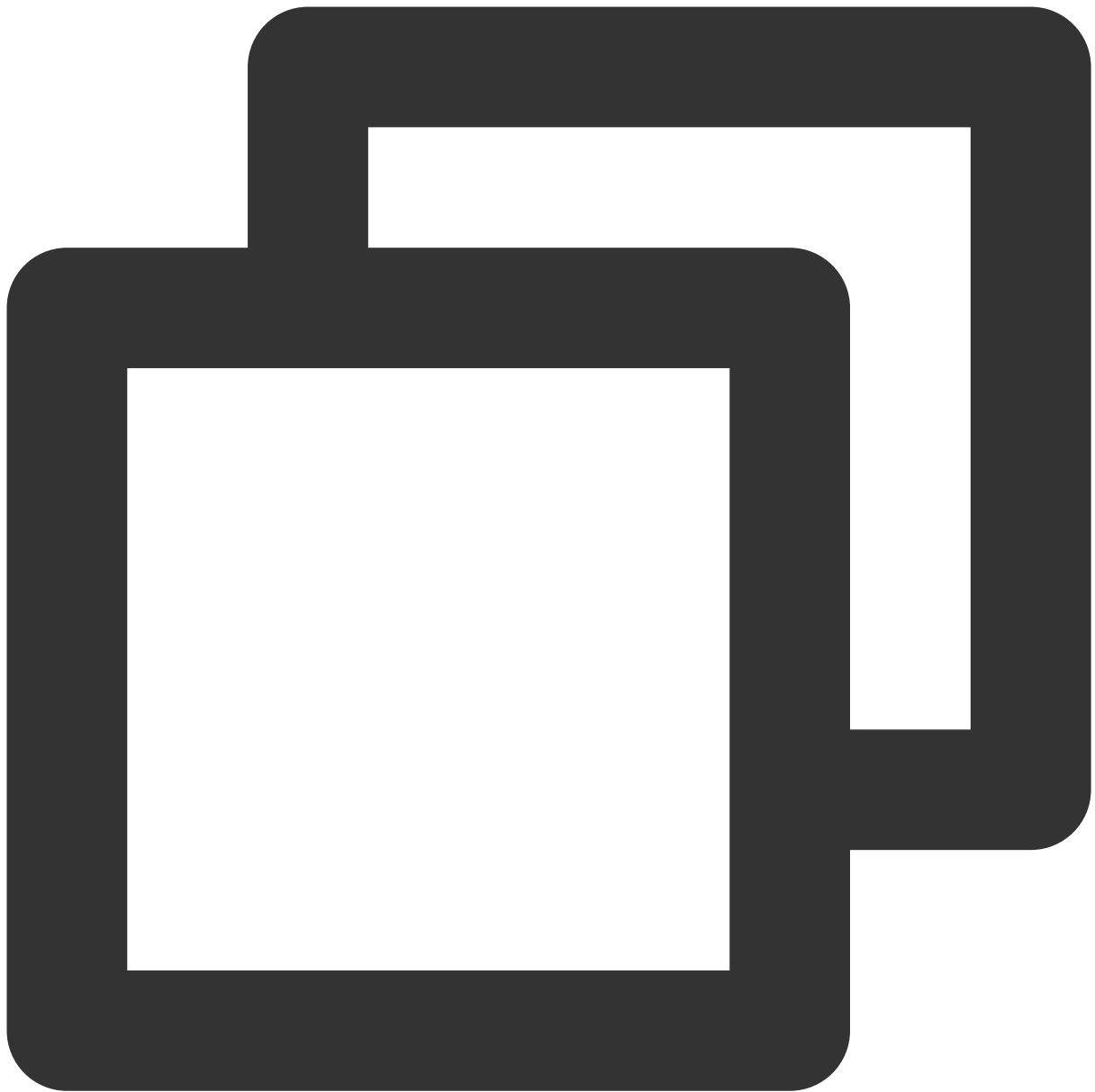
The public key required for verifying the Token carried on the client

Step 3: configure a Tencent Cloud API Gateway business API.

1. In the authorization API service, create a business API. When you are configuring the frontend, select **OAuth 2.0** as the authentication type, **Business API** as the OAuth mode, and the created authorization API as the associated authorization API.
2. When you are configuring the backend, select **mock** as the backend type and enter `hello world` as the returned data.

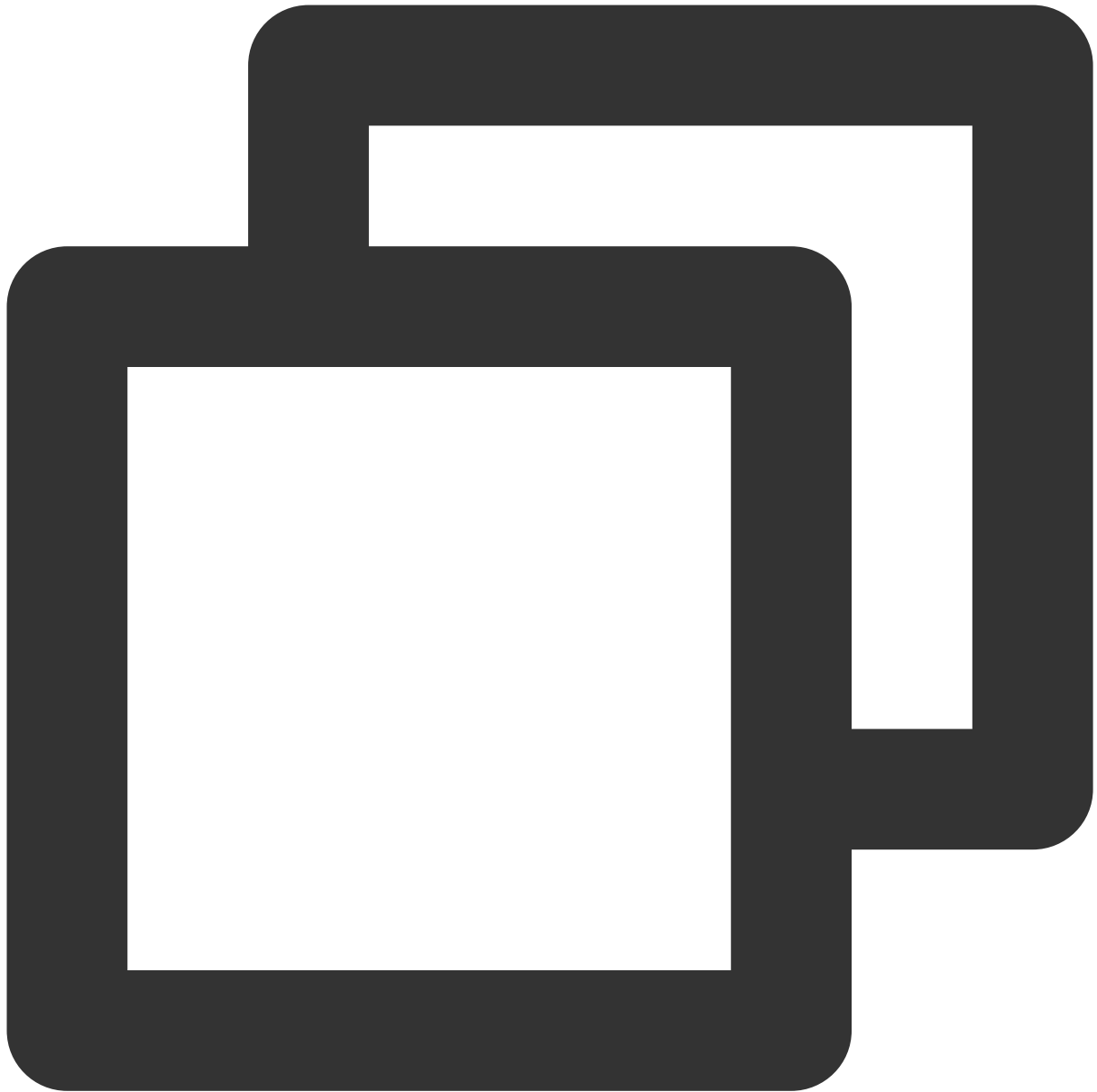
Step 4: perform verification.

1. Request the authorization API to get the token:



```
curl http://service-cmrrdq86-1251890925.gz.apigw.tencentcs.com:80/token
```

Returned result:

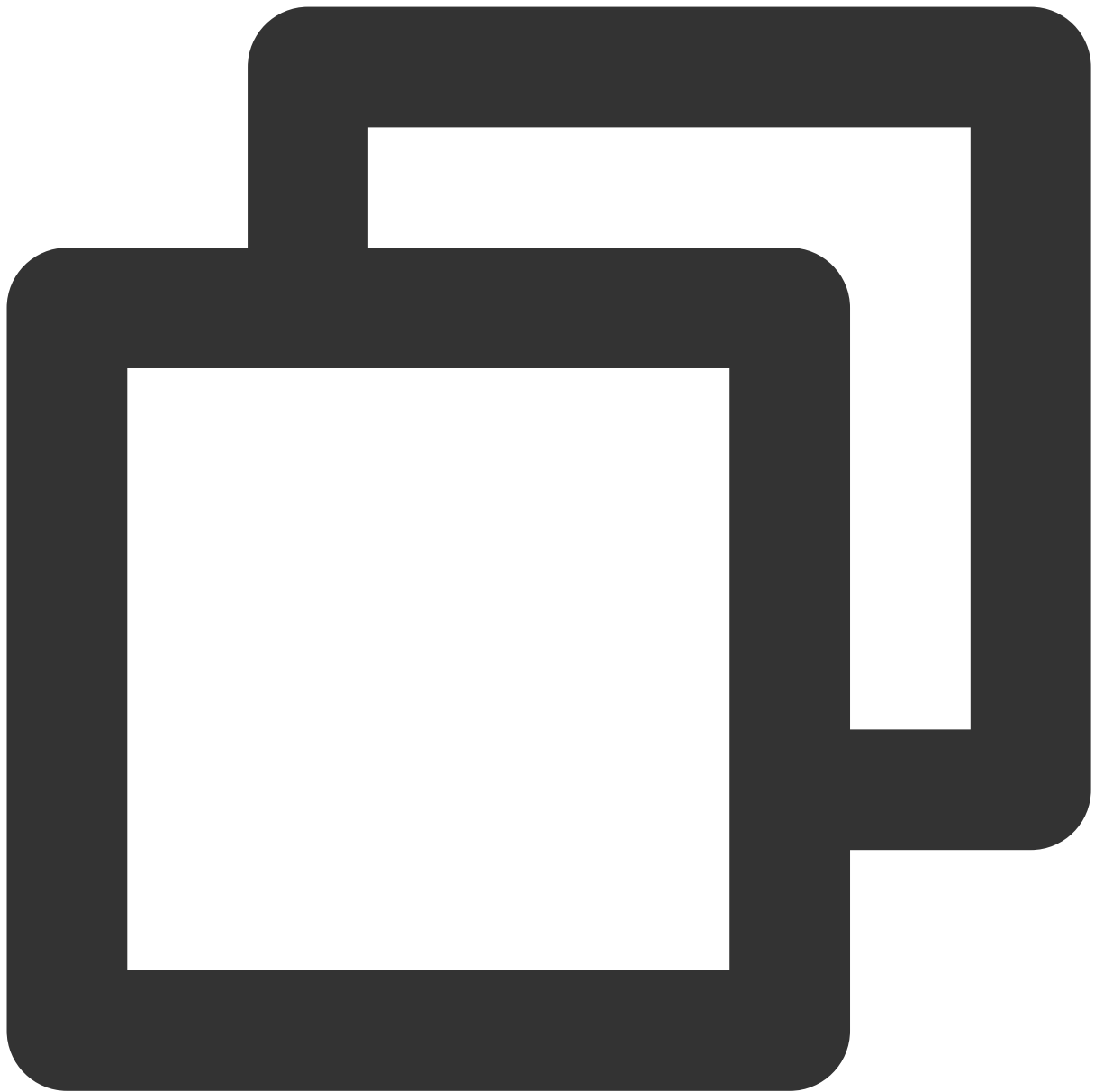


eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1OTIyNzk3MTAsImZvbyI6ImJhcnVzImVudC

Note:

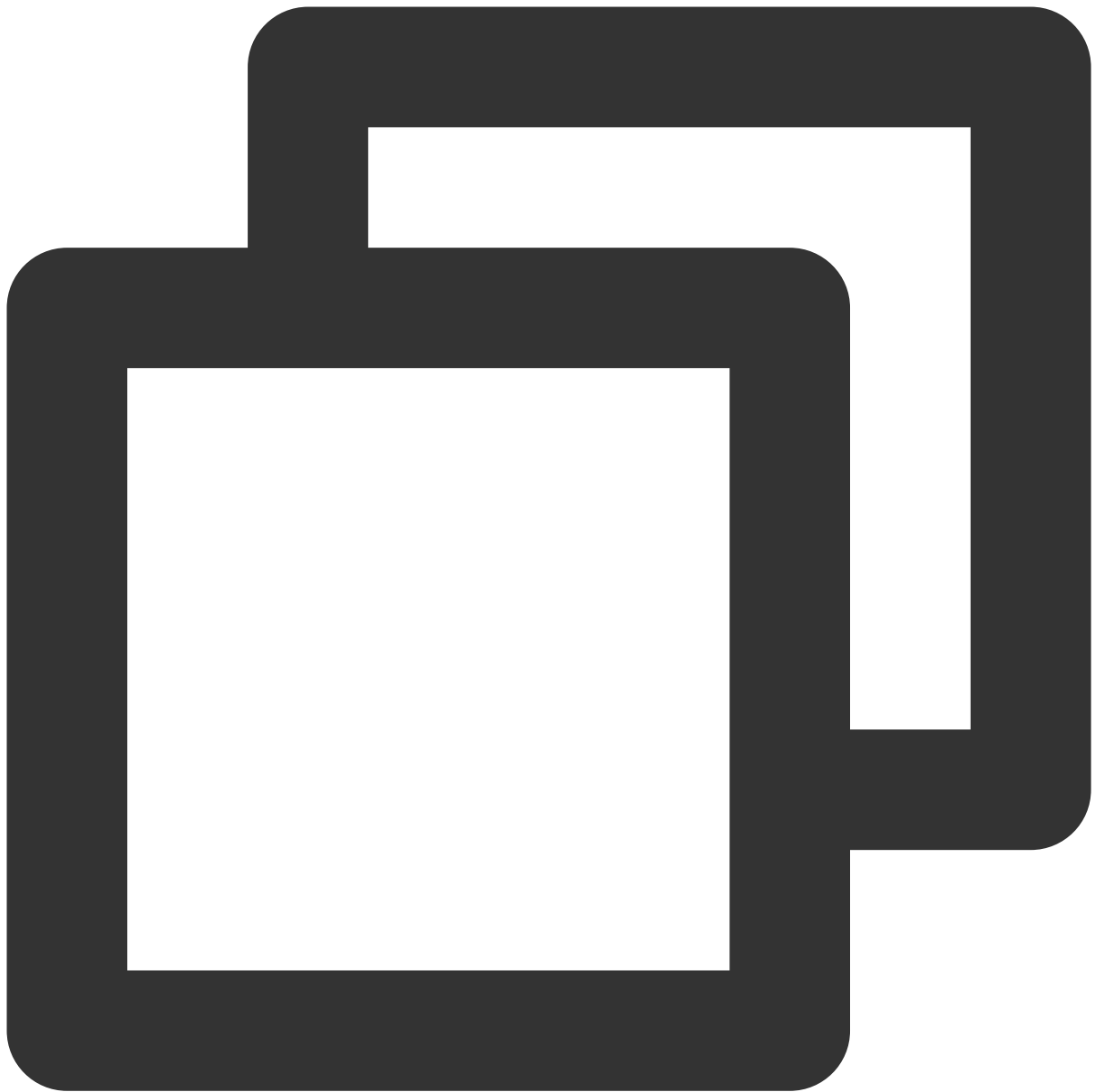
You can get the token using either of the following methods: 1. send a request to the API Gateway authorization API address to get the token; 2. quickly get the token directly from the authorization server. The first method is used in this document to protect the authorization server.

2. Use the token to request the business API. As you can see, the business API can be requested successfully.



```
curl http://service-cmrrdq86-1251890925.gz.apigw.tencentcs.com:80/work -H'Authoriza
```

Returned result:



```
hello world
```

Using the authorization code to get the token

In the sample above, no authorization code is used to get the token. To ensure that only specified users can get the token, the authorization code needs to be obtained from the resource owner according to the authorization process. As can be seen in the `server.py` file, you can first request the authorization code path to get the code and then register the distributed code to verify its validity when getting the token.

CAM Policy

Last updated : 2023-12-22 09:54:39

CAM Introduction

Basic concepts

The root account authorizes sub-accounts by binding policies. The policy setting can be accurate to **"API, Resource, User/User Group, Allow/Deny, Condition"**.

Account

Root account: As the fundamental owner of Tencent Cloud resources, root account acts as the basis for resource usage fee calculation and billing, and can be used to log in to Tencent Cloud services.

Sub-account: An account created by the root account, which has a specific ID and identity credential that can be used to log in to Tencent Cloud console. A root account can create multiple sub-accounts (users). **A sub-account does not own any resources by default, and must be authorized by its root**

account.Identity credential: Includes login credential and access certificate. **Login credential** refers to a user's login name and password, while **access certificate** refers to the Cloud API key (SecretID and SecretKey).

Resources and permission

Resource: Resources are objects that the cloud services operate on, such as the CVM instance, COS bucket and VPC instance.

Permission : Permission is an authorization to allow or forbid users to perform certain operations. By default, **root account has full access to all resources under the account**, while **sub-accounts do not have access to any resources under its root account**.

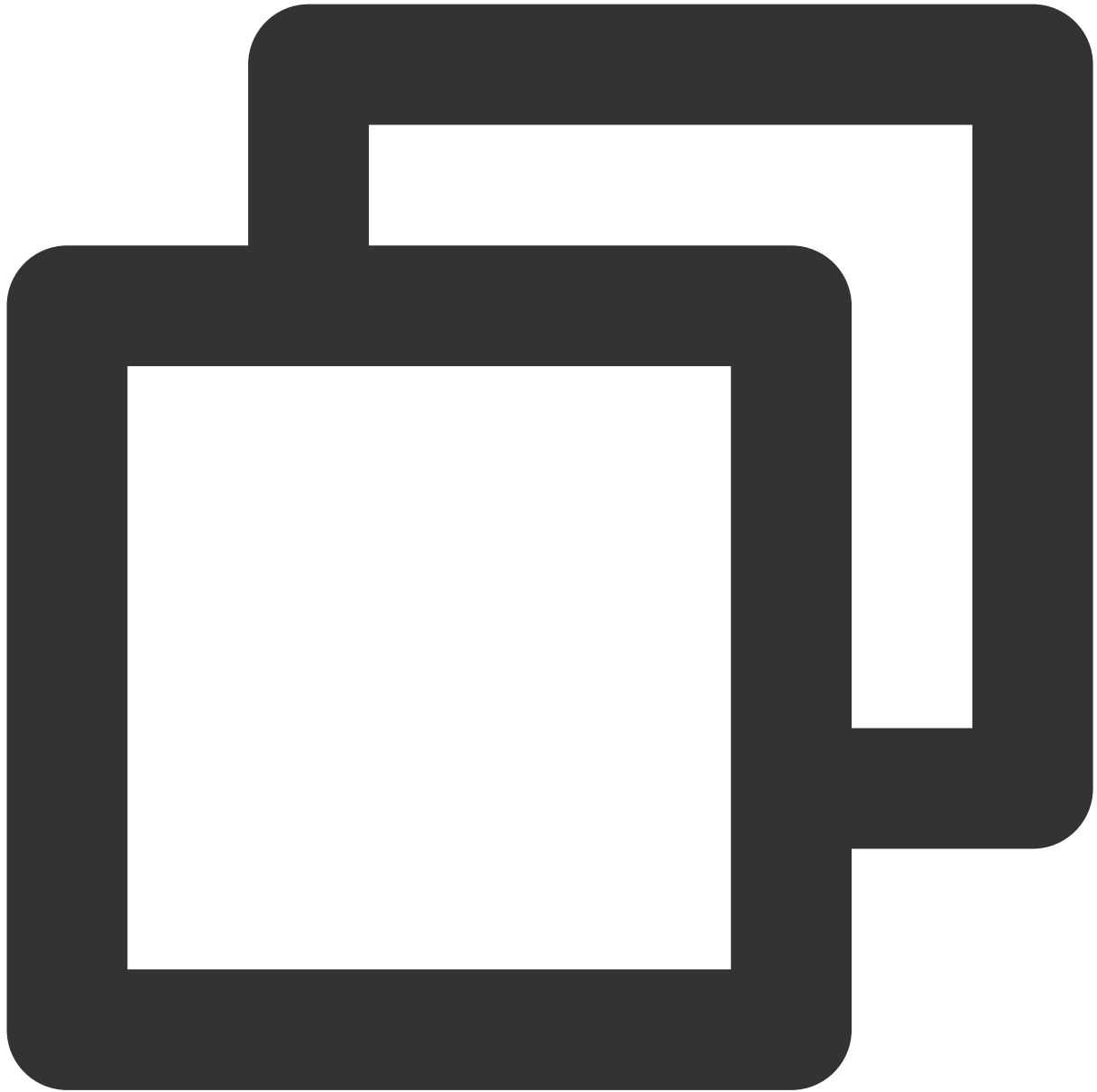
Policy : Policy is the syntax rule used to define and describe one or more permissions. **Root account** performs authorization by **associating policies** with users/user groups.

Related Documents

Content	Link
Relationship between policy and user	Policy Management
Basic structure of policy	Policy Syntax
More products that support CAM	CAM-Enabled Products

[Click to learn more about CAM](#)

API Gateway Resources



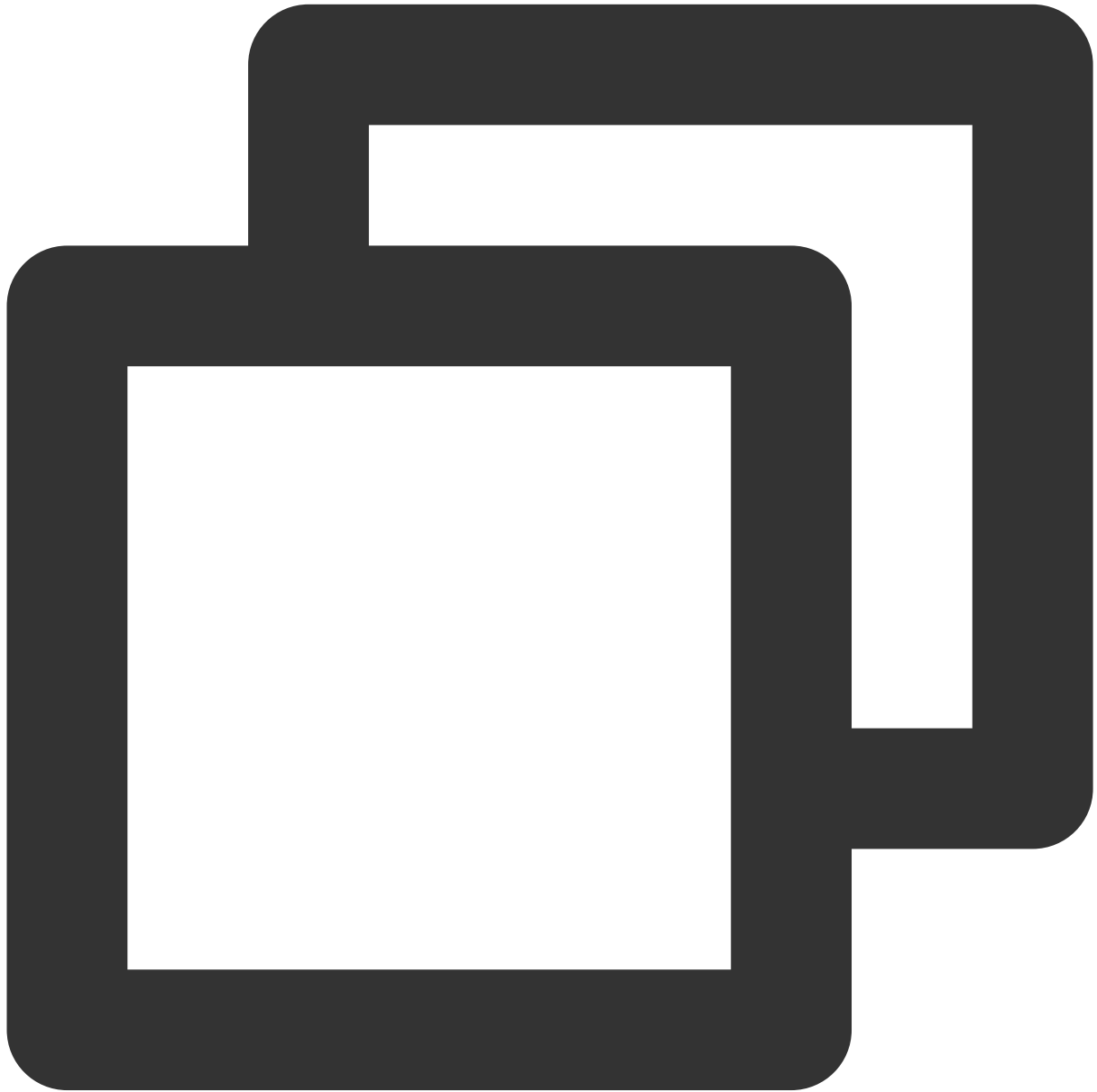
```
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:service/_`serviceid`_  
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:service/_`serviceid`_/API/_`apiid`_  
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:usagePlan/_`usagePlanid`_  
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:secret/_`secretid`_  
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:IPStrategy/_`IPStrategyId`_  
- qcs::APIGateway:_`region`_:uin/_`uin-id`_:logRule/_`logRuleId`_
```

All creation APIs are at account level, while other APIs are at resource level.

CAM Policy Examples

Full read-write policy for any API Gateway resources

The following policy statement gives the sub-user permission to fully manage (creating, managing, etc.) any API services.



```
{  
  "version": "2.0",  
  "statement": [  
    {
```

```
"action": [
    "apigw:*"
],
"resource": "*",
"effect": "allow"
}
]
```

You can also configure the system's [full read-write policy](#) to support this permission.

Create by Policy Syntax

✓ Select policy template > 2 Edit Policy

Template Type: System APIGW

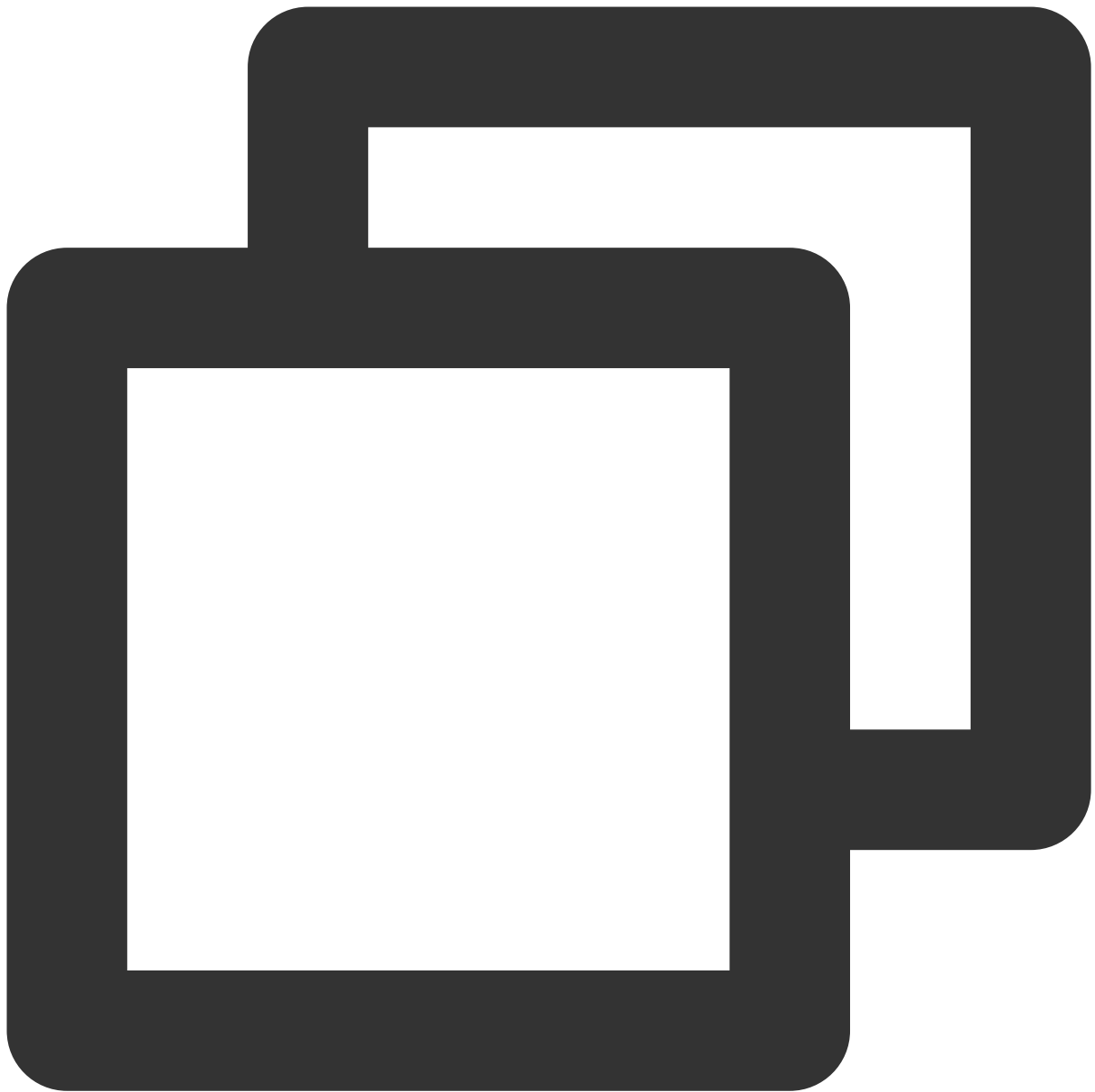
Select template type

System Template Search "APIGW", 2 result(s) found. [Back to the original list](#)

<input type="radio"/> QcloudAPIGWFullAccess System Full read-write access to API Gateway	<input type="radio"/> QcloudAPIGWReadOnlyAccess System Read-only access to API Gateway
--	--

Full management policy for single API Gateway service

The following policy statement gives the sub-user permission to fully manage (creating, managing, etc.) a specified API service:

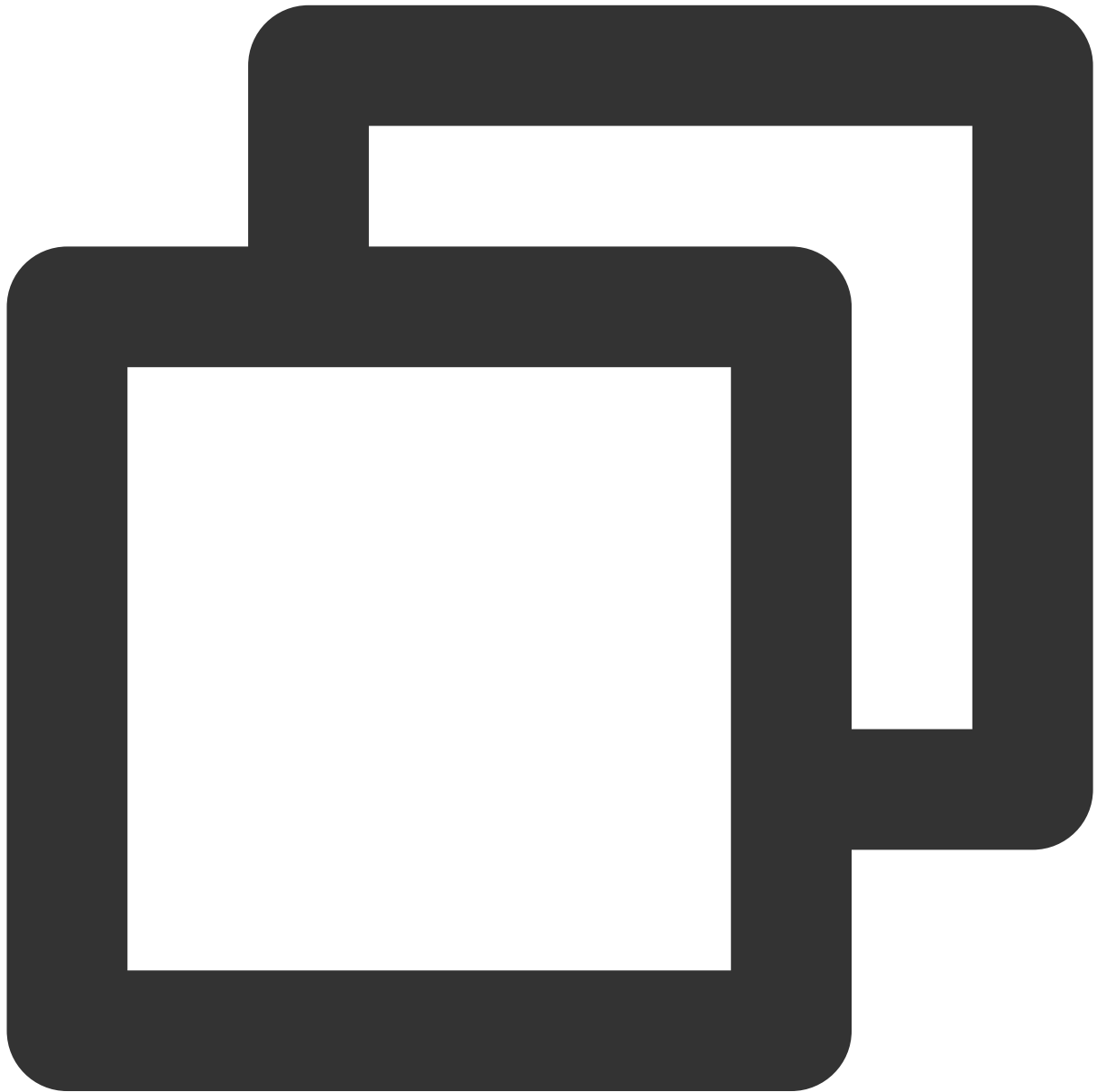


```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "apigw:*"
      ],
      "resource": "qcs::apigw:ap-guangzhou:uin/{ownerUin}:service/service-id/A",
      "effect": "allow"
    }
  ]
}
```

```
}
```

Read-only policy for single API Gateway service

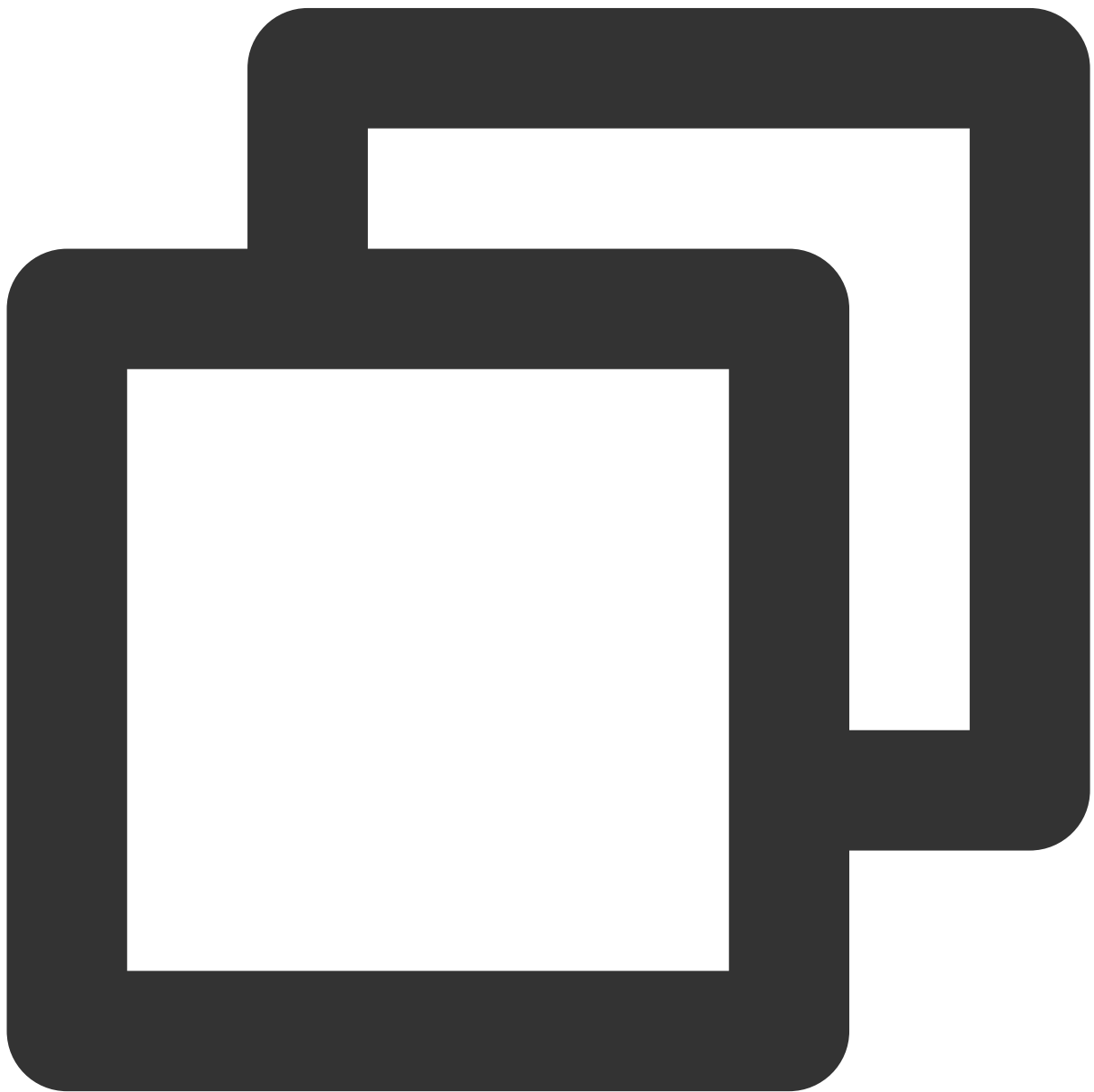
1. Create a policy with policy generator, and grant the permissions to list information for all resources and product monitoring. The following policy statement will grant read-only permission to all resources of the account.



```
{  
  "version": "2.0",  
  "statement": [  
    {
```

```
    "action": [  
        "apigw:Describe*",  
        "apigw:GenerateApiDocument"  
    ],  
    "resource": "*",  
    "effect": "allow"  
}  
]  
}
```

2. Grant read-only permission to a single API.



```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "ckafka:Get*",
        "ckafka:List*"
      ],
      "resource": "qcs::apigw:ap-guangzhou:uin/{ownerUin}:service/service-id/API",
      "effect": "allow"
    }
  ]
}
```

Log Statistics

View Log Analysis

Last updated : 2023-12-22 09:55:11

Operation Scenarios

This document describes how to view logs of requests received by API Gateway in the Tencent Cloud API Gateway Console.

Directions

1. Log in to the [API Gateway Console](#) and click **Service** on the left sidebar to enter the service list page.
2. On the service list page, click a service name to enter the service details page.
3. At the top of the service details page, click **Service Log** to open the log page of the service.
4. You can find service logs as needed.

Time Range

The real-time service log feature allows you to view real-time logs and logs in the last 3 hours and to customize the time range for log query.

With the real-time service log feature, you can view logs in the last 30 days.

The time range for log query cannot exceed one day.

Query by Criteria

Currently, the Tencent Cloud API Gateway Console allows you to query real-time service logs by `RequestID` , API ID, or keyword.

RequestID: if you enter the complete `RequestID` of a request, API Gateway will return the logs that correspond to it for the selected time range. You cannot enter a partial `RequestID` for fuzzy match.

API ID: if you enter the complete API ID of an API under the current service, API Gateway will return all logs for the API for the selected time range. You cannot enter a partial API ID for fuzzy match.

Keyword: if you enter a keyword, API Gateway will return all logs containing a field that exactly matches the keyword for the selected time range.

N :

In one single query, you can query by only one of the three query criteria.

If you don't select a query criterion, the query will be performed by keyword by default.

Exporting Service Logs

Last updated : 2023-12-22 09:55:22

Overview

This document describes how to export service logs through the API Gateway console for data analysis and problem location.

Directions

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to access the service list page.
2. On the service list page, click a service name to access the service details page.
3. Click the **Service Log** tab.
4. Click the **Export** icon in the red box as shown in the following figure to export service logs.

Real time	Last 3 hours	2020-09-04 13:46:15 ~ 2020-09-04 14:46:15	Descend by time	Add filters	?
Request ID	Time	API ID/Path	Environment	Protocol	
▶ 2fe5143efaa179091c483b0d7a9b15d7	2020-09-04 14:45:44	api-fy193s5m /favicon.ico	release	http	2
▶ 7808f8c1f97dd272179612feebde4266	2020-09-04 14:45:44	api-fy193s5m /release	release	http	2
▶ 967d74f2715242dfe1ac6cacf8d993b8	2020-09-04 14:45:43	api-fy193s5m /favicon.ico	release	http	2
▶ e97b2455780821f4d44c525da899a77e	2020-09-04 14:45:43	api-fy193s5m /release	release	http	2
▶ de515b599c733717c20449025fab3f5f	2020-09-04 14:45:43	api-fy193s5m /favicon.ico	release	http	2

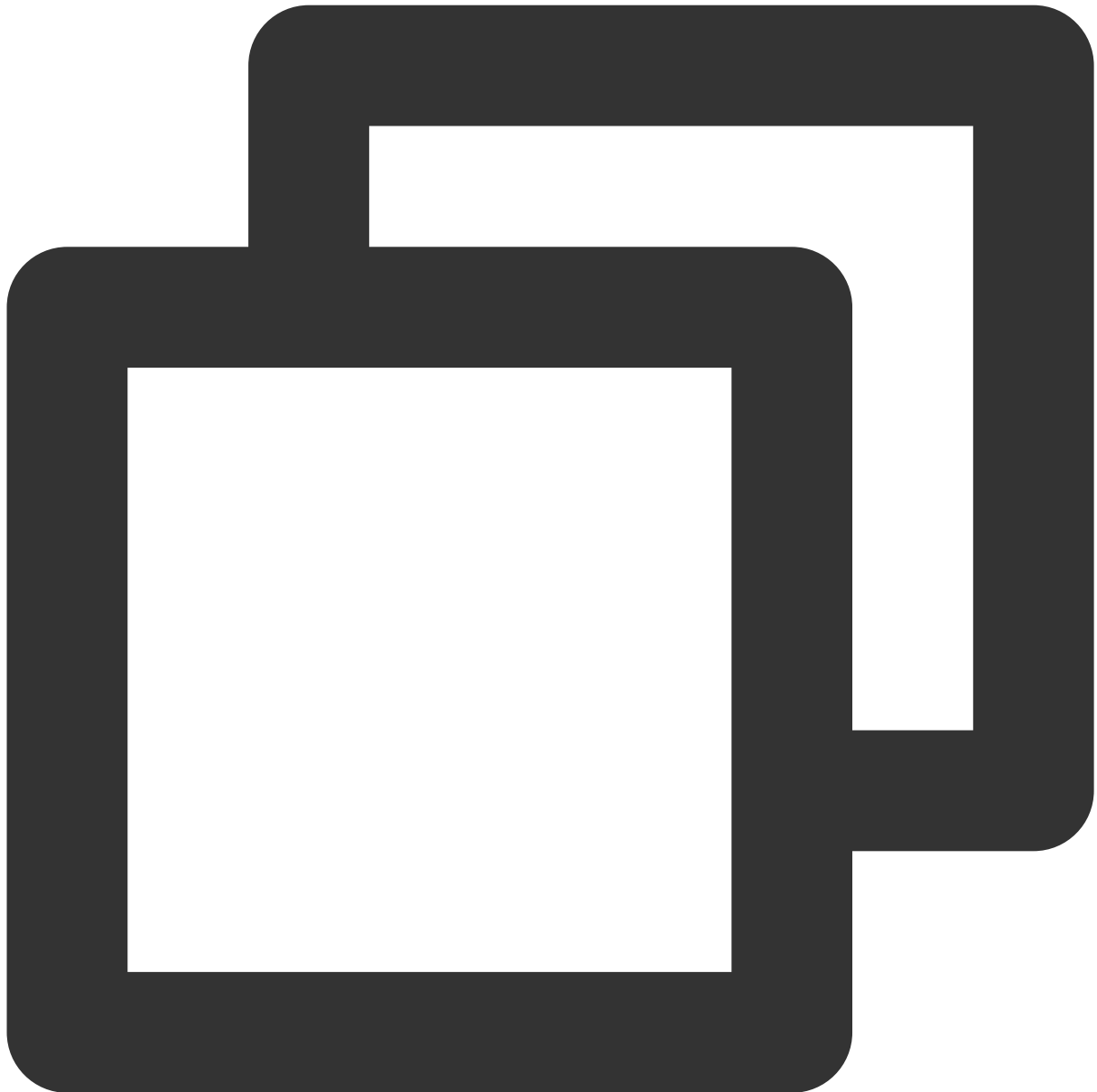
Note:

The time range of the logs to be exported is consistent with the time range of the queried logs. You can modify the time range of the logs to be exported by modifying the time range of the queried logs.

The exported log file is a CSV file. The file name is in the following format: Service Log-service ID (start time - end time).

Log Format

The format of exported service logs is as follows:



```
log_format
```

```
'[$app_id] [$env_name] [$service_id] [$http_host] [$api_id] [$uri] [$scheme] [rsp_st:$stat  
'[cip:$remote_addr] [uip:$upstream_addr] [vip:$server_addr] [rsp_len:$bytes_sent] [req_  
'[req_t:$request_time] [ups_rsp_t:$upstream_response_time] [ups_conn_t:$upstream_conn  
'[err_msg:$err_msg] [tcp_rtt:$tcpinfo_rtt] [$pid] [$time_local] [req_id:$request_id]';
```

The following table describes the parameters.

Parameter	Description
app_id	User ID.
env_name	Environment name.
service_id	Service ID.
http_host	Domain name.
api_id	API ID.
uri	Request path.
scheme	HTTP/HTTPS protocol.
rsp_st	Request response status code.
ups_st	Backend business server's response status code (if the request is passed through to the backend, this variable will not be empty. If the request is blocked by the API Gateway, this variable will be displayed as <code>-</code>).
cip	Client IP.
uip	Backend business service (upstream) IP.
vip	VIP requested to be accessed.
rsp_len	Response length.
req_len	Request length.
req_t	Total request and response time.
ups_rsp_t	Total backend response time (time between the connection establishment by the API Gateway and the backend response receipt).
ups_conn_t	Time when the backend business server is successfully connected to.
ups_head_t	Time when the backend response header arrives.
err_msg	Error message.
tcp_rtt	Client TCP connection information. Round Trip Time (RTT) consists of 3 parts: link propagation delay, end system processing delay, and queuing delay in the router cache.
pid	Process ID.
time_local	Time when the request is initiated.

req_id	Request ID.
--------	-------------

Viewing Operation Logs

Last updated : 2023-12-22 09:55:33

Operation Scenarios

This document describes how to query and download API Gateway operation history in the [CloudAudit Console](#). [CloudAudit](#) enables you to perform supervision, compliance check, operational review, and risk review for your Tencent Cloud account. It provides event history of your Tencent Cloud account activities, including operations performed through Tencent Cloud Console, APIs, command line tools, and other Tencent Cloud services, which simplifies security analysis, resource change tracking, and troubleshooting.

Directions

1. Log in to the [CloudAudit Console](#).
2. On the left sidebar, click **Event History** to enter the event history page.
3. On the event history page, you can query the operations by username, resource type, resource name, event source, event ID, etc. By default, only partial data will be displayed, and you can click **View More** at the bottom of the page to get more results.

ResourceType		apigw	Nearby 7 days		2020-04-02 00:00:00 ~ 2020-04-08 23:59:59
Event time	User name	Event name	Resource type		
▶ 2020-04-08 15:41:58	100011677482	CreateService	apigw		
▶ 2020-04-06 14:40:35	100011677482	ReleaseService	apigw		
▶ 2020-04-06 14:10:58	100011677482	ReleaseService	apigw		

4. You can click



on the left of an operation to view its details such as access key, error code, and event ID. You can also click **View Event** to view the details of an event.

ResourceType		apigw	✕ 🔍		Nearly 7 days	2020-04-02 00:00:00 ~ 2020-04-08 23:59:59		📅
Event time		User name		Event name		Resource type		
▼ 2020-04-08 15:41:58		100						

Shipping to CLS

Last updated : 2023-12-22 09:55:45

Overview

API Gateway records the client access logs, which can help you better understand client requests, troubleshoot issues, and analyze user behaviors.

API Gateway provides a basic log dashboard in the console, where you can directly view and search for logs. It also allows you to ship logs to [CLS](#) for multidimensional statistical analysis.

Directions

Step 1. Create a logset and log topic

To configure access logs in CLS, you need to first create a logset and log topic.

You can directly proceed to [step 2](#) if you have already created a logset and log topic.

1. Log in to the [API Gateway console](#) and click **Tool > Log delivery** on the left sidebar.
2. On the **Access Logs** page, select a region for the logset, and then click **Create Logset** in the **Logset Information** section.
3. In the pop-up **Create Logset** dialog box, set the retention period and click **Save**.

Note:

You can only create a single logset named "apigw_logset" in each region.

4. Click **Create Log Topic** in the **Log Topic** section of the **Access Logs** page.
5. In the pop-up window, select a dedicated API Gateway instance to add to the list on the right, and then click **Save**.

Note:

API Gateway logs are shipped at the instance level. Only dedicated instances can ship logs to CLS, while shared instances can't.

In the **Operation** column on the right of the log topic list, you can click **Manage** to edit the added dedicated API Gateway instance.

Each API Gateway instance can be added to only one log topic.

Multiple log topics can be created in a logset. You can store logs of different dedicated API Gateway instances in different log topics.

6. (Optional) To disable logging, just click **Disable**.

Step 2. View access logs

Without any manual configurations, API Gateway has been automatically configured with index search by access log variable. You can directly query access logs through search and analysis.

1. Log in to the [API Gateway console](#) and click **Tool > Log delivery** on the left sidebar.
2. Select a log topic, and click **Search** in the operation column to redirect to the **Search Analysis** page in the [CLS console](#).
3. On the **Search Analysis** page, enter the search syntax in the input box, select a time range, and then click **Search Analysis** to search for access logs reported by API Gateway to CLS.

Note:

See [Syntax and Rules](#) for more information on search syntax.

Log Format Description

A shipped service log is in the following format:



log_format

```
'[$app_id][$env_name][$service_id][$http_host][$api_id][$uri][$scheme][rsp_st:$stat  
'[cip:$remote_addr][uip:$upstream_addr][vip:$server_addr][rsp_len:$bytes_sent][req_  
'[req_t:$request_time][ups_rsp_t:$upstream_response_time][ups_conn_t:$upstream_conn  
'[err_msg:$err_msg][tcp_rtt:$tcpinfo_rtt][$pid][$time_local][req_id:$request_id]';
```

The parameters are as detailed below:

Parameter	Description
app_id	User ID.

env_name	Environment name.
service_id	Service ID.
http_host	Domain name.
api_id	API ID.
uri	Request path.
scheme	HTTP/HTTPS protocol.
rsp_st	Request response status code.
ups_st	Backend business server response status code (if the request is passed through to the backend, this variable will not be empty. If the request is blocked in API Gateway, this variable will be displayed as -).
cip	Client IP.
uip	Backend business service (upstream) IP.
vip	VIP requested to be accessed.
rsp_len	Response length.
req_len	Request length.
req_t	Total request response time.
ups_rsp_t	Total backend response time (time between connection establishment by API Gateway and backend response receipt).
ups_conn_t	Time when the backend business server is successfully connected to.
ups_head_t	Time when the backend response header arrives.
err_msg	Error message.
tcp_rtt	Client TCP connection information. RTT (Round Trip Time) consists of three parts: link propagation delay, end system processing delay, and queuing delay in router cache.
pid	Process ID.
time_local	Request time.
req_id	Request ID.

Notes

API Gateway logs are shipped at the instance level. Only dedicated instances can ship logs to CLS, while shared instances can't.

Shipping API Gateway access logs to CLS is now free of charge. You only need to pay for the CLS service.

This feature is only supported in CLS available regions. See [Available Regions](#).

Access Monitoring

Viewing Monitoring Charts

Last updated : 2023-12-22 09:55:57

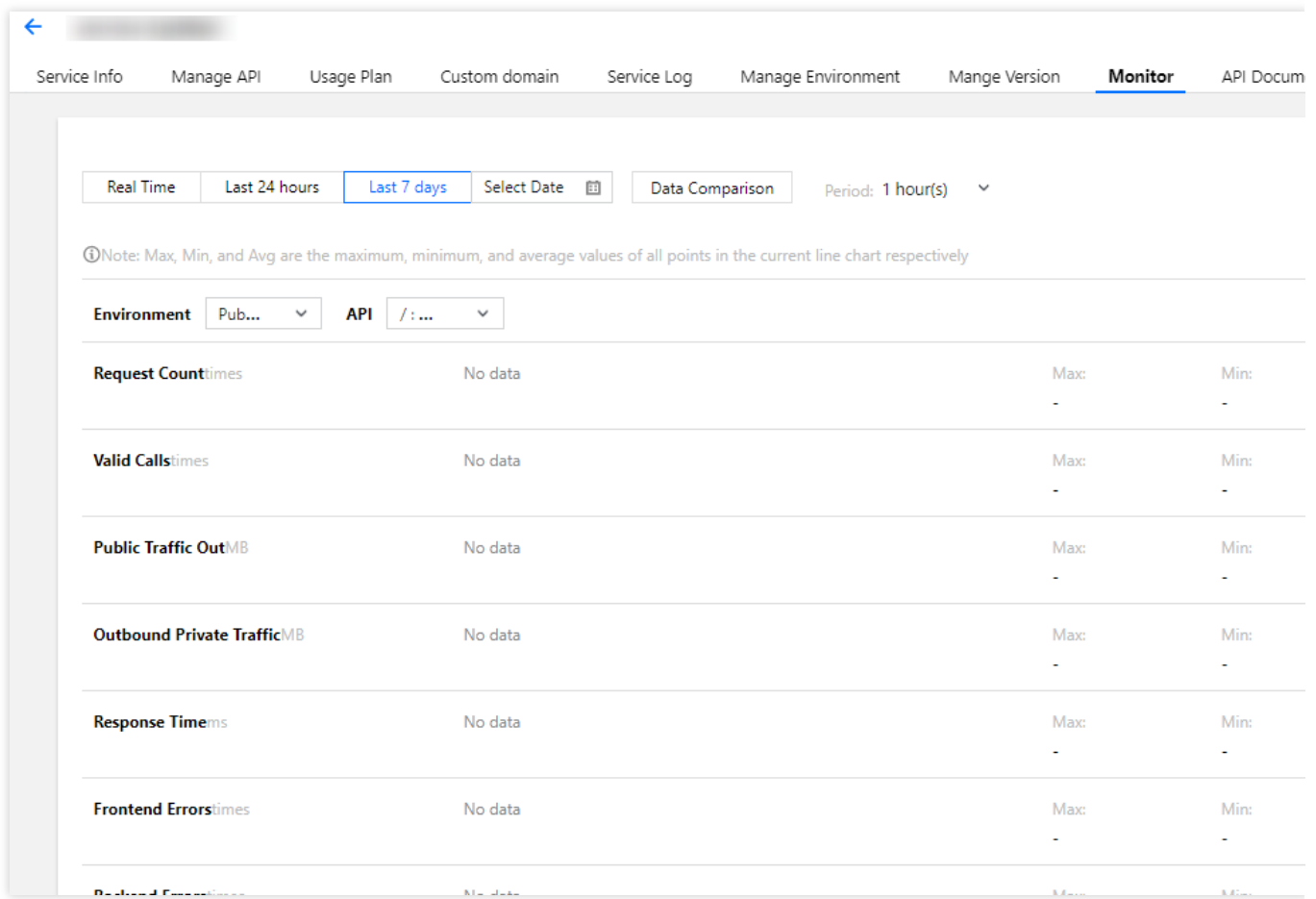
Overview

This document describes how to view monitoring charts by environment, API, and key, respectively, in the API Gateway console.

Directions

Viewing monitoring charts by environment

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to access the service list page.
2. In the service list, click a service name to access the service details page.
3. At the top of the service details page, click the **Monitoring** tab to access the page for viewing monitoring charts.
4. Select an environment from the **Environment** drop-down list box and select **All** from the **API** drop-down list box. Then you can view the statistical monitoring information of all APIs in the corresponding environment, including the number of calls, traffic, response durations, and errors.



Viewing monitoring charts by API

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to access the service list page.
2. In the service list, click a service name to access the service details page.
3. At the top of the service details page, click the **Monitoring** tab to access the page for viewing monitoring charts.
4. Select an environment from the **Environment** drop-down list box and select an API from the **API** drop-down list box. Then you can view the statistical monitoring information of the API in the environment, including the number of calls, traffic, response durations, and errors.

Viewing monitoring charts by key

Prerequisites

A key pair authentication API has been configured by referring to [Key Pair Authentication](#), and the API or the service where the API resides has been bound to a usage plan.

Directions

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to access the service list page.
2. In the service list, click a service name to access the service details page.

3. At the top of the service details page, click the **Manage Environment** tab to access the environment management page.
4. On the environment management page, select an environment and click **Enable Key Monitoring**.
5. At the top of the service details page, click the **Monitoring** tab to access the page for viewing monitoring charts.
6. Select the environment for which key monitoring is enabled, select the key pair authentication API, and select a key pair. Then you can view the key monitoring information of the API in the environment to obtain the API call information.

Viewing API Statistics

Last updated : 2023-12-22 09:56:12

Overview

This task guides you on how to use API Gateway to view the statistical data of all APIs in the service environment.

Directions

1. Log in to the [API Gateway console](#) and click **Service** in the left sidebar to access the service list page.
2. On the service list page, click a service name to access its service details page.
3. At the top of the service details page, click **Monitoring** to go to the monitoring page of the service.
4. On the **Monitoring** page, choose the **Statistics** tab and then select the **Environment** and time to view the monitoring data of all APIs in the service environment within the specified time period.

The screenshot shows the 'Monitoring' page for a service named 'service-b6mulwqw'. The 'Monitoring' tab is selected in the top navigation bar. Below it, the 'Statistics' tab is selected in the 'Charts' section. The 'Environment' is set to 'Publish', and the time range is 'Real time'. A table displays the following data:

API ID/Name	Requests	Public Traffic Out	Outbound Private Traffic	Frontend Errors	Backend Errors	Backend
api-17ood9c2 index	0 times	0MB	0MB	0 times	0 times	0 times

Total items: 1

Note:

Currently, monitoring data in real time, in the last 3 hours, and in the last 24 hours are supported.

Currently, the following 7 metrics are supported: **Requests** (times), **Public Outbound Traffic**, **Outbound Private Traffic**, **Frontend Errors** (times), **Backend Errors** (times), **Backend 404 Errors** (times), **Backend 502 Errors** (times). For more information about the monitoring metrics, please see [Monitoring Metrics](#).

Monitoring Metrics

Last updated : 2023-12-22 09:56:23

Tencent Cloud Monitor provides the following monitoring metrics for API Gateway:

Metric Name	Description	Statistical Method	Unit
Requests	Number of requests passing API Gateway	Sum in the selected time period	-
Valid calls	Number of valid call requests passing API Gateway	Sum in the selected time period	-
Public network outbound traffic	Traffic generated by public network data packets sent by API Gateway	Sum in the selected time period	MB
Private network outbound traffic	Traffic generated by private network data packets sent by API Gateway	Sum in the selected time period	MB
Response time	Amount of time it takes API Gateway to respond to a request	Average in the selected time period	ms
Frontend errors	Number of invalid requests sent to API Gateway by the client, such as authentication failures or exceeding the upper limit	Sum in the selected time period	-
Backend errors	Number of status codes greater than or equal to 400 returned by the backend service after API Gateway forwards messages to the backend service	Sum in the selected time period	-
Concurrent connections	Number of current persistent connections to API Gateway	Average in the selected time period	-
Backend 404 errors	Number of errors where the requested resource is not found on the real server	Sum in the selected time period	-

Backend 502 errors	Number of errors where an invalid response is received by the real server when API Gateway attempts to execute a backend request	Sum in the selected time period	-
--------------------	--	---------------------------------	---

API Doc Generator

Last updated : 2023-12-22 09:58:26

Overview

You can use the API Doc Generator to generate exquisite API documents for APIs hosted in API Gateway and provide them to third-party callers of your APIs.

Note:

The API Doc Generator is completely free of charge with technical support provided by CODING DevOps. Click [here](#) to learn more about the capabilities of CODING.

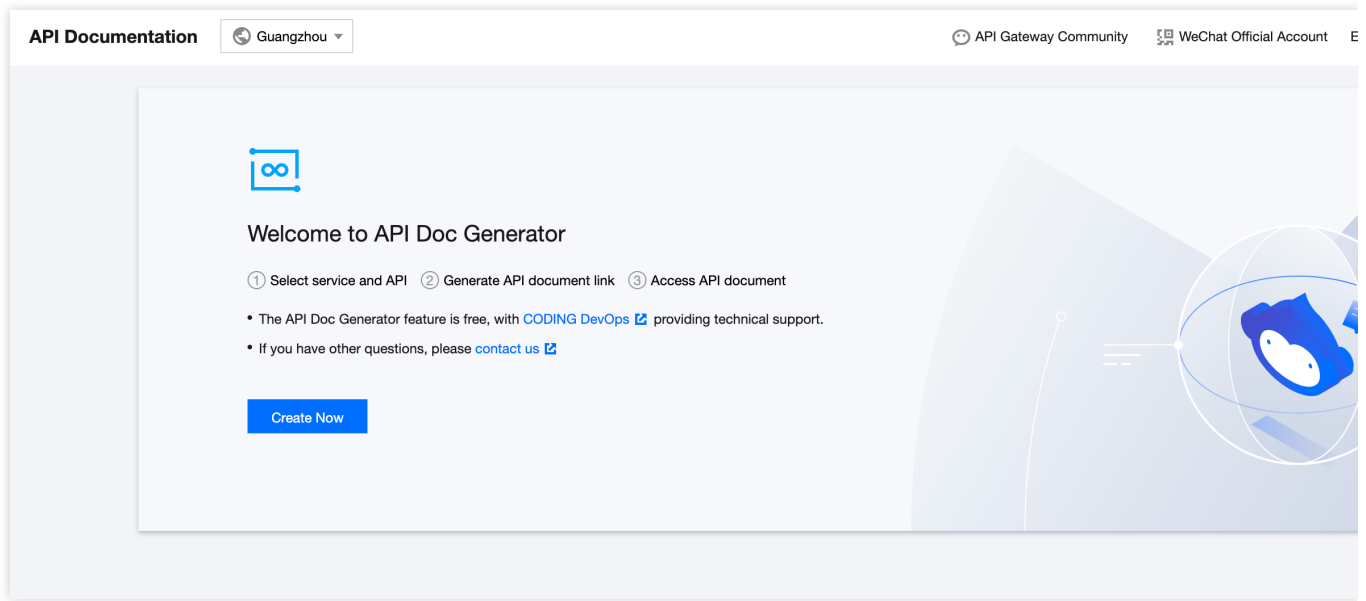
Prerequisites

You have created a service and an API in API Gateway (as instructed in [Creating Services](#) and [API Creation Overview](#)) and released the service to any environment (as instructed in [Service Release and Deactivation](#)).

Directions

Creating documents

1. Log in to the [API Gateway console](#) and click **Tool > API Doc Generator** in the left sidebar.



2. On the **API Doc Generator** page, click **Create now**, enter the document name in the pop-up window, select the environment, service, and API, and click **Submit**.

New API Document

Name

Document name

Environment

Publish

Select a service

service-a47z4tim(SCF_API_SERVICE)

Please select API

已选择(0)

Please enter a keyword

Q

☐

ID/Name

Path

Method

☐

api-cjo3653i
APIGWHtmlDemo-...

/APIGWHtmlDemo...

ANY

ID/Name

Path

Metho

No data yet

Support for holding shift key down for multiple selection

Submit

Disable

3. Wait patiently for the API document creation to complete.

Viewing document details

The following is the details page of an API document:

API Documentation

Guangzhou

API Gateway Community

WeChat Official Account

You can generate an API document for the APIs managed on API Gateway and provide it to a third party for calling your APIs. See [API Doc Generator Instructions](#) to learn more.

New Document

apidoc-cb15qt9m

test

Running

apidoc-cb15qt9m

APIs	Access Count	Builds
1	0 times	1 times

Basic Info

API Document Link

<https://apidoc-cb15qt9m-gz-apigw.doc.coding.io> Copy

API Document Password

HiSzcRhk [Reset](#) [Copy](#)

Share

API document link: <https://apidoc-cb15qt9m-gz-apigw.doc.coding.io>; access password: HiSzcRhk [Copy](#)

Last Updated

2022-04-01 21:14:51

Build Information

Service

[service-a47z4tim \(SCF_API_SERVICE\)](#)

Environment

Publish

API

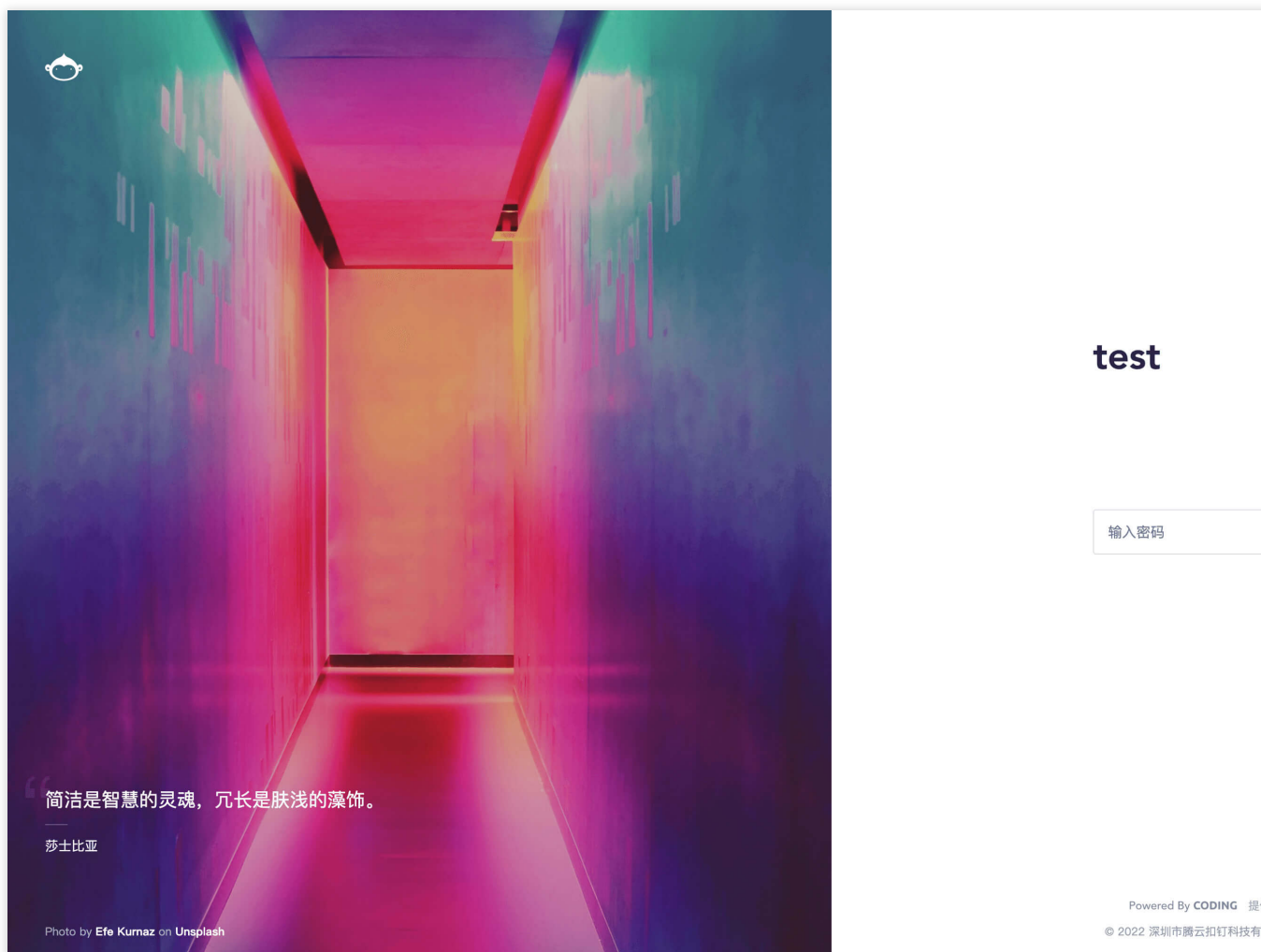
[api-cjo3653i \(APIGWHtmlDemo-1641376852\)](#)

Parameters are described in the following table.

Parameter	Description
API document address	The access address of the current API document.
API document password	The password of the current API document. API documents are encrypted by default and can be viewed only with the correct password.
Shared link	You can copy the link and share it with a third party.
Update time	Last time when the API document was updated.
Service	The service of the API for which the document is generated.
Environment	The environment where the service is released, which can be used to generate the API call address.
API	The API contained in the current API document.

Accessing documents

1. Copy the API document address and paste it in a browser to open the document login page.



2. Enter the API document password on the document login page to view the content of the document.

API Documentation Guangzhou

API Gateway Community WeChat Official Account

You can generate an API document for the APIs managed on API Gateway and provide it to a third party for calling your APIs. See [API Doc Generator Instructions](#) to learn more.

Enter a keyword

New Document

apidoc-cb15qt9m Running
test

apidoc-cb15qt9m

APIs	Access Count	Builds
1	0 times	1 times

Basic Info
API Document Link [https://...](#) [Copy](#)
API Document Password [Reset](#) [Copy](#)
Share [API doc...](#) [Copy](#)
Last Updated 2022-04-01 21:14:51

Build Information
Service [...](#) (SERVICE)
Environment Publish
API [ap-...](#) (io-1641376852)

Updating documents

After editing the API for which an API document is generated, the document will not be updated synchronously. Using the "document update" feature can ensure that the API document is consistent with the API information. The steps are as follows:

1. Click **Update** in the top-right corner of the document details page.
2. Click **Confirm** in the pop-up window and wait for the document construction to complete.

API Documentation Guangzhou ▾ API Gateway Community WeChat Official Account

apidoc-cb15qt9m

APIs	Access Count	Builds
1	0 times	1 times

Basic Info

API Document Link [http://...](#) [Copy](#)

API Document Password [Reset](#) [Copy](#)

Share [API doc...](#) [Copy](#)

Last Updated 2022-04-01 21:14:51

Build Information

Service [...](#) [SERVICE](#)

Environment [Publish](#)

API [ap...](#) [o-1641376852](#)

Resetting the password

After you reset the API document password, a new password will be generated. Users can only use the new password to access the document, while the old password will not work. The steps are as follows:

1. Click **Reset** after the API document password.
2. Click **Confirm** in the pop-up window to generate a new API document password.

API Documentation Guangzhou API Gateway Community WeChat Official Account

You can generate an API document for the APIs managed on API Gateway and provide it to a third party for calling your APIs. See [API Doc Generator Instructions](#) to learn more.

Q
New Document

apidoc-cb15qt9m Running
test

apidoc-cb15qt9m

APIs	Access Count	Builds
1	0 times	1 times

Basic Info
API Document Link [https://...](#) Copy
API Document Password Reset Copy
Share API Copy
Last Updated 2022-04-01 21:14:51
Build Information
Service [s...](#) (RVICE)
Environment P
API [a...](#) 1641376852

Deleting documents

1. Click **Delete** in the top-right corner of the document details page.
2. Click **Confirm** in the pop-up window to delete the API document.

Application Management

Last updated : 2023-12-22 09:58:37

Overview

An application is an identity that calls an API. It needs to be authorized by the API before it can call the API. Each application has a key pair of `ApiAppKey` and `ApiAppSecret`. `ApiAppKey` needs to be passed in as a parameter in the header of a request, and `ApiAppSecret` needs to be used to calculate the request signature. For more information on the signature calculation method, please see [Application Authentication Method](#).

Prerequisites

The API authentication method is "application authentication".

Directions

Application creation

1. Log in to the [API Gateway console](#) and click **Application** on the left sidebar to enter the application management page.
2. On the application management page, click **Create Application** in the top-left corner, fill in the form, and submit it to create an application.

API authorization

Authorization refers to granting an application the permission to call an API. The application needs to be authorized by the API first before it can call the API. There are two ways of authorization:

Authorization Method	Applicable Scenario	Description
Direct authorization	Application created by yourself	-
Partner authorization	Use the APIs provided by partners (other accounts)	Create your own application and provide its ID to the API provider. The API provider can search for the application ID to perform authorization

You can click the following tabs to view the directions of the corresponding authorization method.

Direct authorization

Partner authorization

1. Log in to the [API Gateway console](#) and click **Service** on the left sidebar.
 2. In the service list, click the name of the target service to view it.
 3. In the service information, click the **Manage API** tab and click **Authorization** behind the API list to start authorization.
 4. Select the environment and application to be authorized. Your applications are on the left. Click **Search** directly, and the applications under the current account will be automatically loaded.
1. Log in to the API Gateway console and click **Service** on the left sidebar.
 2. In the service list, click the name of the target service to view it.
 3. In the service information, click the **Manage API** tab and click **Authorization** behind the API list to start authorization.
 4. If you want to authorize applications under other accounts, you need to select the application ID, enter it in the text box, and then click **Search** to query.

Notes

The key pair of `AppKey` and `AppSecret` has all the permissions of the application and should be kept private. If it is disclosed, you can reset it in the API Gateway console.

You can create multiple applications and authorize them to different APIs according to your business needs.

You can create, modify, and delete applications, view application details, manage keys, and view authorized APIs in the API Gateway console.

Permission Management

Last updated : 2023-12-22 09:58:49

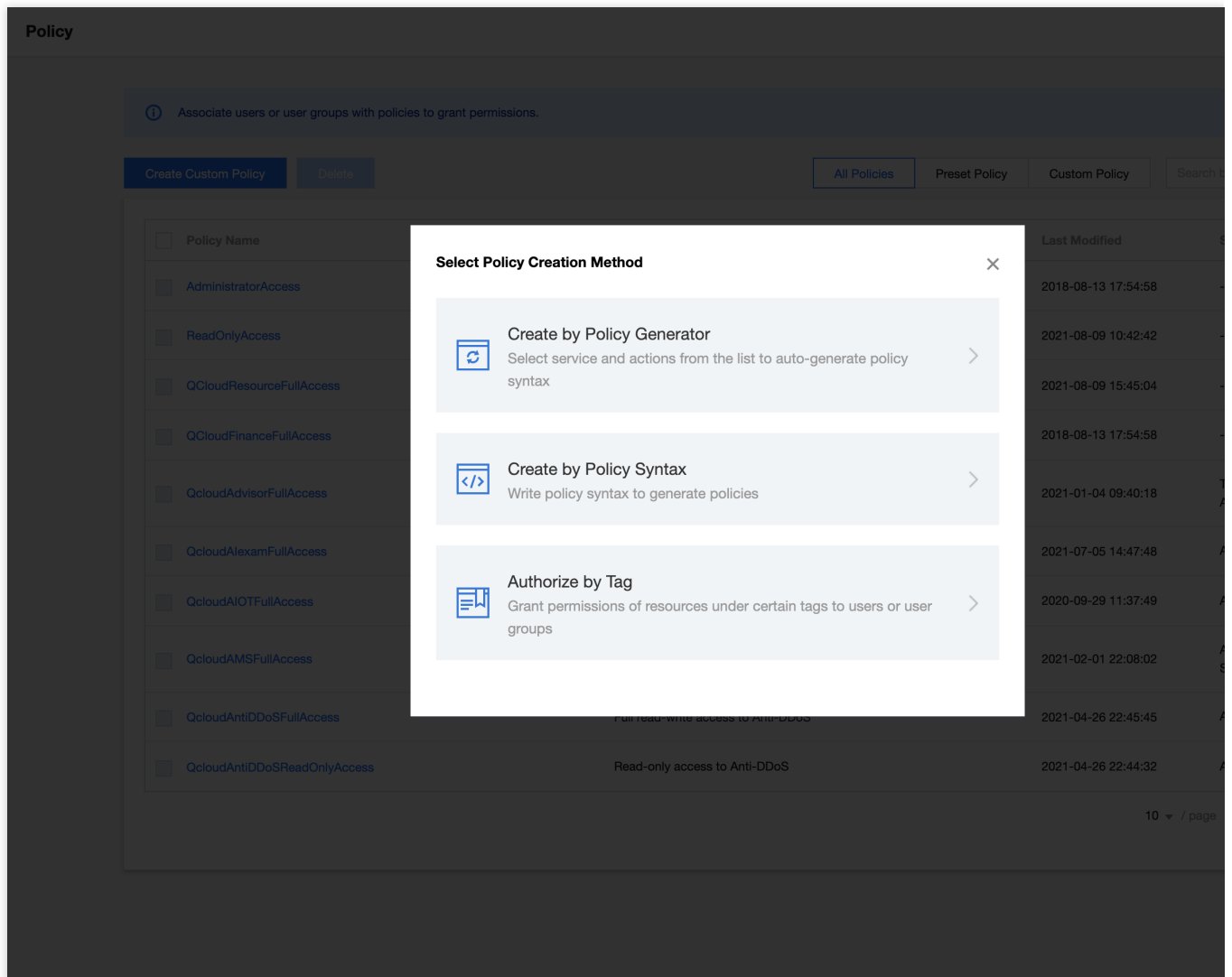
Operation Scenarios

In the CAM Console, the root account can configure permissions of common operations, services that can be manipulated, and API resources on API Gateway for sub-accounts and collaborators.

Directions

Sub-account or collaborator permission management (non-tagged resource)

1. Log in to the [CAM Console](#) > [Policy](#) with the root account.
2. In the policy list, click **Create Custom Policy** in the top-left corner.
3. In the policy creation pop-up window, select **Create by Policy Builder**:



4. Enter the custom policy information. After the settings are completed, click **Add Statement** and **Next** to enter the policy editing page.

← Create by Policy Generator

1 Edit Policy

>

2 Associate Users/User Groups

Visual Policy Generator

JSON

▼ API Gateway(All actions)

Effect *

☒ Allow
 ☐ Deny

Service *

API Gateway (apigw)

Action *

Collapse

Select actions

☒ All actions (apigw:*) [Show More](#)

Action Type

☒ Read (40 selected) [Show More](#)

☒ Write (64 selected) [Show More](#)

☒ List (7 selected) [Show More](#)

Resource *

Select resources.

Condition

☐ Source IP ⓘ
 [Add other conditions.](#)

+ Add Permissions

Next

Effect: allowed.

Service: API Gateway.

Action: select involved operations.

Resource: you can enter `*` to indicate all resources. If only part of resources are involved, you can set this field as instructed in [Resource Description Method](#).

5. Enter the policy name and edit the policy content (for detailed directions, please see [Policy Syntax](#)).

6. Click **Create Policy** to create the policy.

7. Bind the created policy to users or user groups (for detailed directions, please see [Associating Policy with User/User Group](#)).

Sub-account or collaborator permission management (tagged resource)

Tag is a unified capability in Tencent Cloud. You can set the same tag for different resources in different Tencent Cloud products so that they can have the same operation permissions.

Note:

The same entry is used for configuring tagged management permissions and non-tagged management permissions.

You can create an **Authorize by Tag** policy in **CAM > Policy**. For detailed directions, please see [Authorizing by Tag](#).

1. Log in to the [API Gateway Console](#) > [Service](#).

©2013-2022 Tencent Cloud. All rights reserved.

Page 168 of 172

2. In the service list, click a service name to enter the service details page and click **Manage API**.
3. On the API management page, click an API ID to enter the API details page.
4. On the API details page, click the



icon under "Tags" at the bottom of the page to modify the tags.

Pre-publishFollow service bandwidth throttling value (Max: 5000 QPS)

TestFollow service bandwidth throttling value (Max: 5000 QPS)

API traffic throttling is only effective for current API. It is limited by service traffic throttling. You can go to service configuration page to change service traffic throttling setting.
Please use basic traffic throttling plugins for throttling by API/application/ClientIP with a time granularity at the second/minute/hour/day level. For details, see [User Guide](#).

Cross-Origin Resource Sharing (CORS) [Edit](#)

Current Status **Disable**

After enabling, the API Gateway will add Access-Control-Allow-Origin: * in the response header by default to solve the cross-domain problem of resources.
To customize CORS configuration, please create a CORS plugin and bind it with the API. See [CORS Plugin Usage Guide](#).

Response compression

Minimum Compression Threshold **1KB**

Compression Algorithm **gzip**

Content-Type **text/xml;text/plain;text/css;application/javascript;application/x-javascript;application/rss+xml;application/xml;application/json;application/octet-stream**

It is enabled by default. When the client request carries the Accept-Encoding: gzip request header, and the Content-Type meets the conditions and the response body is larger than 1KB, the response is compressed.
For more information, please see [Response Compression Instructions](#).

Base64 Encoding [Edit](#)

Current Status **Disable**

After you enable this feature, API Gateway will Base64-encode your request content before sending it to SCF to support binary file upload.
You can configure Base64 encoding to be triggered for all requests or based on specific Content-Type and Accept headers. For more information, please see [Base64 Encoding Instructions](#).

Tag Management (Total 0)

[Tag Management Guide](#)

Precautions

Last updated : 2023-12-22 09:59:00

You can use API Gateway to design your businesses as APIs for service provision. You are recommended to design APIs in compliance with the following rules for easier future use:

Do not add a slash (/) at the end of the URI. A resource whose URL has an ending slash may be mistaken as a directory, causing errors in calls.

Use a hyphen (-) to improve the readability of the URI. Connect words with hyphens so that the URI can be easily understood.

It is not allowed to use the underscore (_) in the URL, as it may be covered by the underline effect in the text viewer and thus less readable.

Avoid using uppercase letters which are not aesthetically pleasing and error-prone.

Do not include an extension in the URI. REST API clients are recommended to use the format selection mechanism "Accept request header" provided by HTTP.

Private IP Ranges and Public VIPs of API Gateway Regions

Last updated : 2023-12-26 14:35:54

Warning:

Tencent Cloud platform is planning to upgrade shared services in different regions in the following time periods:

1. Before the official upgrade, the platform will uniformly send various notifications such as SMS and Message Center.
2. During the upgrade process, your service access will not be affected.
3. After the completion of the upgrade, the public network VIPs of the shared services will change.

For services created under shared instances in the API Gateway, you may refer to the following information to check the private network gateway and the public network IP.

The private network ranges can be used for interconnection, while the public network VIPs can be used to differentiate the outbound traffic for ultimately generating the billing statement.

Note:

Shared instances are multi-user, and the public network VIPs **often change**.

If a user creates a service using a shared instance and sets a **security group** due to business needs, they should promptly check this page for synchronous updates in the security group.

If a user wants a fixed VIP while using the service, it is recommended to purchase a dedicated instance.

The private network IP ranges and public network VIPs in different regions are as follows:

Regions	Private IP Network Ranges (consistent across all regions)	Public Network VIPs
Guangzhou	9.0.0.0/8,10.0.0.0/8,100.64.0.0/10,11.0.0.0/8,30.0.0.0/8	125.94.61.15,125.94.61.16,125.94.61.17
Beijing	9.0.0.0/8,10.0.0.0/8,100.64.0.0/10,11.0.0.0/8,30.0.0.0/8	123.206.35.235,123.206.35.236,123.206.35.237
Shanghai	9.0.0.0/8,10.0.0.0/8,100.64.0.0/10,11.0.0.0/8,30.0.0.0/8	211.159.135.21,211.159.135.22,211.159.135.23
Hong Kong (China)	9.0.0.0/8,10.0.0.0/8,100.64.0.0/10,11.0.0.0/8,30.0.0.0/8	123.206.35.245,123.206.35.246,123.206.35.247

Singapore	9.0.0.0/8,10.0.0.0/8,100.64.0.0/10,11.0.0.0/8,30.0.0.0/8	119.28.123.219,119.28.123.220,119.28.123.221,119.28.123.222,119.28.123.223,119.28.123.224,119.28.123.225,119.28.123.226,119.28.123.227,119.28.123.228,119.28.123.229,119.28.123.230,119.28.123.231,119.28.123.232,119.28.123.233,119.28.123.234,119.28.123.235,119.28.123.236,119.28.123.237,119.28.123.238,119.28.123.239,119.28.123.240,119.28.123.241,119.28.123.242,119.28.123.243,119.28.123.244,119.28.123.245,119.28.123.246,119.28.123.247,119.28.123.248,119.28.123.249,119.28.123.250,119.28.123.251,119.28.123.252,119.28.123.253,119.28.123.254,119.28.123.255