

API Gateway

FAQs

Issue 02
Date 2025-01-24



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Contents

1 Common FAQs.....	1
2 Product Consulting FAQs.....	2
2.1 What Are the Relationships Between an API, Environment, and Credential?.....	2
2.2 Can I Upgrade the Shared Gateway to a Dedicated Gateway?.....	2
2.3 Why All Buttons on the APIG Console Cannot Be Clicked?.....	3
2.4 How Do I Ensure API Calling Security?.....	3
2.5 How Do I Ensure the Security of Backend Services Invoked by APIG?.....	3
3 Opening APIs.....	4
3.1 Why Can't I Create APIs?.....	4
3.2 How Do I Define Response Codes for an API?.....	4
3.3 Can I Specify a Private Network Load Balancer Address for the Backend Service?.....	4
3.4 Can I Specify the Backend Address as a Subnet IP Address?.....	4
3.5 Does APIG Support Multiple Backend Endpoints?.....	5
3.6 Can I Bind Private Domain Names for API Access?.....	5
3.7 Why Do I Fail to Call an API Across Domains?.....	5
3.8 How Do I Use APIG to Open Up Services Deployed on Huawei Cloud?.....	6
4 Publishing an API.....	8
4.1 Do I Need to Publish an API Again After Modification?.....	8
4.2 Can I Access an API Published in a Non-RELEASE Environment?.....	8
4.3 Can I Invoke Different Backend Services by Publishing an API in Different Environments?.....	8
4.4 Can I Specify an Environment for API Debugging?.....	9
5 Calling APIs.....	10
5.1 What Are the Possible Causes for an API Calling Failure?.....	10
5.2 Why Am I Seeing the Error Message "414 Request URI too large" When I Call an API?.....	11
5.3 Why Am I Seeing the Error Message "The API does not exist or has not been published in the environment" When I Call an API?.....	11
5.4 Why Am I Seeing the Error Message "No backend available" When I Call an API?.....	12
5.5 Why Am I Seeing the Error Message "Backend unavailable" or "Backend timeout" When I Invoke the Backend Service?.....	12
5.6 Why Am I Seeing the Error Message "Backend domain name resolution failed" When I Invoke the Backend Service?.....	13
5.7 Why Am I Seeing the Error Message "Incorrect IAM authentication information" When I Call an API?.....	14

5.8 Why Am I Seeing the Error Message "Incorrect app authentication information" When I Call an API?	17
5.9 Why Doesn't Modification of the backend_timeout Parameter Take Effect?.....	18
5.10 Does APIG Have a Limit on the Size of the API Request Body?.....	19
5.11 Is There a Limit on the Size of the Response to an API Request?.....	19
5.12 How Do I Call an API Using App Authentication in iOS System?.....	19
5.13 Why Can't I Create a Header Parameter Named x-auth-token for an API Called Through IAM Authentication?.....	19
5.14 Can Mobile Apps Call APIs?.....	20
5.15 Can Applications Deployed in a VPC Call APIs?.....	20
5.16 Does APIG Support WebSocket Data Transmission?.....	21
5.17 How Will the Requests for an API with Multiple Backend Policies Be Matched and Executed?.....	21
5.18 How Can I Access Backend Services over Public Networks Through APIG?.....	21
6 API Authentication.....	23
6.1 Does APIG Support Two-Way Authentication?.....	23
6.2 Will the Request Body Be Signed?.....	23
6.3 API Credential FAQs.....	23
7 API Policies.....	25
7.1 Can I Configure the Maximum Number of Concurrent Requests?.....	25
7.2 Is the Restriction of 1,000 Requests Per Day to a Debugging Domain Name Applied to Enterprise Accounts?.....	25
7.3 Does APIG Have Bandwidth Limits?.....	25
7.4 Why Doesn't a Request Throttling Policy Take Effect?.....	25
7.5 How Do I Provide an Open API to Specific Users?.....	26
7.6 Are Client IP Addresses Verified for Access Control?.....	26
8 Importing and Exporting APIs.....	27
8.1 What Are the Possible Causes of an API Import Failure?.....	27
8.2 Is There a Template for Importing APIs Using a Swagger File?.....	27

1 Common FAQs

Opening APIs

- [Can I Specify a Private Network Load Balancer Address for the Backend Service?](#)
- [Can I Specify the Backend Address as a Subnet IP Address?](#)
- [Can I Bind Private Domain Names for API Access?](#)

Calling APIs

- [What Are the Possible Causes for an API Calling Failure?](#)
- [Why Am I Seeing the Error Message "The API does not exist or has not been published in the environment" When I Call an API?](#)
- [Why Am I Seeing the Error Message "No backend available" When I Call an API?](#)
- [Why Am I Seeing the Error Message "Backend unavailable" or "Backend timeout" When I Invoke the Backend Service?](#)

API Authentication

- [Does APIG Support Two-Way Authentication?](#)

API Policies

- [Can I Configure the Maximum Number of Concurrent Requests?](#)
- [Does APIG Have Bandwidth Limits?](#)
- [How Do I Provide an Open API to Specific Users?](#)

Importing and Exporting APIs

- [What Are the Possible Causes of an API Import Failure?](#)
- [Is There a Template for Importing APIs Using a Swagger File?](#)

2 Product Consulting FAQs

2.1 What Are the Relationships Between an API, Environment, and Credential?

An API can be published in different environments, such as RELEASE (online environment) and BETA (test environment).

An app (credential) refers to the identity of an API caller. After you create an app (credential), the system automatically generates a key and secret for authenticating the app (credential). After an API is published and assigned to an app (credential), the owner of the app (credential) can call the API.

After publishing an API in different environments, you can define different request throttling policies and authorize different apps (credentials) to call the API. For example, during the test process, API v2 is published in the BETA environment and authorized to test apps (credentials). API v1 is stable and can be authorized to all users or apps (credentials) in the RELEASE environment.

2.2 Can I Upgrade the Shared Gateway to a Dedicated Gateway?

Currently, you cannot upgrade the shared gateway to a dedicated gateway. However, you can do as follows to achieve the same purpose:

1. Buy a dedicated gateway.
2. Export APIs from the shared gateway.
3. Import the APIs to the dedicated gateway.
4. Bind a new domain name for the APIs, and change the DNS record to CNAME the domain name to the public access IP address of the dedicated gateway.

2.3 Why All Buttons on the APIG Console Cannot Be Clicked?

Check whether your account is in arrears, and top up your account if necessary.

For details, see [Billing](#).

2.4 How Do I Ensure API Calling Security?

- Identity authentication
Configure IAM or App authentication for APIs to prevent malicious calling.
- Access control policies
Configure a whitelist or blacklist of IP addresses/IP address ranges or accounts for APIs to secure access.
- Request throttling policies
By default, an API can be called up to 200 times per second. If your backend service does not support this access rate, decrease the quota accordingly.

2.5 How Do I Ensure the Security of Backend Services Invoked by APIG?

You can ensure the security of backend services invoked by APIG by using the following methods:

- Bind signature keys to APIs
After a signature key is bound to an API, APIG adds signature information to each request sent to the backend service. The backend service calculates the signature information in each request and checks whether the signature information is consistent with that on APIG.
- Encrypt requests using HTTPS
Ensure that the required SSL certificate exists.
- Perform backend authentication
Enable security authentication for backend services of the desired APIs to process only API requests that carry correct authentication information.

3 Opening APIs

3.1 Why Can't I Create APIs?

The creation of APIs is free of charge. If you cannot create APIs, your account must be in arrears. [Top up](#) your account in time.

For details, see [Billing](#).

3.2 How Do I Define Response Codes for an API?

There are two types of responses:

- Gateway response codes: returned by the gateway for API requests that are throttled, denied, or failed in authentication. For details, see [Customizing Error Response for APIs](#).
- Backend service responses: defined by backend API services (API providers) and transparently transmitted by APIG.

3.3 Can I Specify a Private Network Load Balancer Address for the Backend Service?

- For dedicated gateways, you can use private network load balancer addresses.
- The shared gateway on the old console only supports VPC channels.
- Alternatively, you can use the EIP bound to a public network load balancer.

3.4 Can I Specify the Backend Address as a Subnet IP Address?

If you use a dedicated gateway, you can specify either an IP address that belongs to the same subnet where the gateway is deployed, or the private address of a local data center connected to the gateway through Direct Connect.

Unsupported network segments:

- 0.0.0.0/8
- 10.0.0.0/8
- 100.125.0.0/16
- 127.0.0.0/8
- 169.254.0.0/16
- 172.16.0.0/12
- 192.0.0.0/24
- 192.0.2.0/24
- 192.88.99.0/24
- 192.168.0.0/16
- 198.18.0.0/15
- 198.51.100.0/24
- 203.0.113.0/24
- 224.0.0.0/4
- 240.0.0.0/4
- 255.255.255.255/32

If you use the shared gateway on the old console, you cannot specify the backend address as a subnet IP address. For a backend service deployed on multiple ECSs that are located in the same region but are not bound with any EIPs, **create a VPC channel** and associate the ECSs with it.

3.5 Does APIG Support Multiple Backend Endpoints?

Yes

APIG supports the configuration of multiple backend endpoints through a VPC channel (also called "load balance channel"). You can add multiple cloud servers to each VPC channel.

For details, see [Load Balance Channels](#).

3.6 Can I Bind Private Domain Names for API Access?

In the shared gateway on the old console, the domain name to be bound must have been registered, and there must be CNAME records pointing the domain name to the subdomain name of the group to which the target API belongs. You cannot bind private domain names or domain names that do not support public access to API groups.

In a dedicated gateway, you can add a private domain name (filing not required), and add an A record to point the domain name to the inbound access address of the gateway.

3.7 Why Do I Fail to Call an API Across Domains?

Possible Causes

The CORS configuration of the API is incorrect.

Solution

1. Ensure that CORS has been enabled for the API.

Go to the API details page, click **Edit**, and check whether CORS is enabled. If it is not, enable it.

2. Check whether an API with the OPTIONS method has been created. Only one such API is required for each API group.

Parameters are as follows:

- **API Group:** The same group to which the API with CORS enabled belongs.
- **Method:** Select **OPTIONS**.
- **Protocol:** The same protocol used by the API with CORS enabled.
- **Path:** Same as or prefixally matching the request path set for the API with CORS enabled.
- **Matching:** Select **Prefix match**.
- **Authentication Mode:** **None** means all users will be granted access. It is not recommended.
- **CORS:** Enable this option.

3.8 How Do I Use APIG to Open Up Services Deployed on Huawei Cloud?

- For a service deployed on Huawei Cloud with a **public network IP address**, specify the IP address as the backend service address when creating an API in APIG. If the service has been bound with a domain name, use the domain name as the backend service address. For details about how to create an API, see [Creating an API](#).

The screenshot shows the 'Backend Configuration' interface in APIG. The 'Backend Type' is set to 'HTTP&HTTPS'. The 'Basic Information' tab is active, showing fields for 'URL', 'Method', 'Protocol', 'Backend Address', and 'Path'. The 'Backend Address' field is highlighted with a red box and contains the IP address '192.168.20.10:8448'. Other fields include 'Timeout (ms)' set to 5000, 'Retries' set to -1, and checkboxes for 'Two-Way Authentication' and 'Backend Authentication'.

- For a service deployed on Huawei Cloud without a public network IP address, specify a VPC channel for access to the backend service when creating an API in APIG. For details about how to create a VPC channel and API, see [Creating a Load Balance Channel](#) and [Creating an API](#).

Backend Configuration

Backend Type: **HTTP&HTTPS** | FunctionGraph | Mock

Default Backend

Backend Policies ⓘ

Basic Information

Load Balance Channel: **Configure** | Skip

Method: GET | Protocol: HTTPS | Load Balance Channel: **VPC_4z6** | Create Load Balance Channel | Path: / ⓘ

Host Header ⓘ:

Timeout (ms):

Retries ⓘ:

Two-Way Authentication Use the certificate configured in backend_client_certificate for client authentication. [Configure backend_client_certificate](#)

Backend Authentication ⓘ Use custom authorizer for authentication

Parameter Orchestration

Max. backend, constant, and system parameters: 50. Available for creation: 50

Backend Parameters ⓘ ▾

Constant Parameters ⓘ ▾

System Parameters ▾

4 Publishing an API

4.1 Do I Need to Publish an API Again After Modification?

Yes.

After you modify the parameters of a published API, you must publish the API again to synchronize the modifications to the environment.

For details, see [Publishing an API](#).

4.2 Can I Access an API Published in a Non-RELEASE Environment?

Yes. To access an API published in a non-RELEASE environment, add the **x-stage** header to the API request.

Example:

```
r.Header.Add("x-stage", "RELEASE")
```

You can also refer to the examples in [Quickly Opening and Calling APIs](#).

4.3 Can I Invoke Different Backend Services by Publishing an API in Different Environments?

Yes, you can invoke different backend services by publishing an API in different environments while specifying environment variables and backend parameters.

For details, see [\(Optional\) Configuring the Environment and Environment Variables](#).

4.4 Can I Specify an Environment for API Debugging?

During API debugging, the specific debugging environment of APIG is used by default. Therefore, you cannot specify other environments.

After debugging is completed, you need to publish your API in an environment, and use code or Postman to add the X-Stage header to specify the environment where you want to call the API.

5 Calling APIs

5.1 What Are the Possible Causes for an API Calling Failure?

Network

API calling failures may occur in three scenarios: within a VPC, between VPCs, and on a public network.

- Within a VPC: Check whether the domain name is the same as that automatically allocated for the API.
- Between VPCs: Check whether the two VPCs are connected. If they are not connected, create a VPC peering connection to connect the two VPCs.
For details about how to create and use VPC peering connections, see [VPC Peering Connection Overview](#) or [Exposing Backend Services Across VPCs](#).
- On a public network:
 - The API is not bound with an EIP and does not have a valid address for public network access.
Bind an EIP to the API and try again. For details, see [Network Environment](#).
 - The inbound rules are incorrectly configured.
For details about how to configure inbound rules, see [Network Environment](#).
 - The request header "host:*Group domain name*" is not added when you call the API. Add the request header and try again.

Domain Name

- Check whether the domain name bound to the API group to which the API belongs has been successfully licensed and can be resolved.
- Check whether the domain name has been bound to the correct API group.
- The subdomain name (debugging domain name) automatically allocated to the API group is accessed too many times. The subdomain name can be

accessed only 1000 times a day. It is unique and cannot be modified. Add independent domain names for the group to make the APIs in the group accessible.

API Publishing

Check whether the API has been published. If the API has been modified, publish it again. If the API has been published to a non-RELEASE environment, specify the **X-Stage** header as the environment name.

API Authentication

If the API uses app authentication, check whether the AppKey and AppSecret used to call the API are correct.

API Control Policies

- Check whether the access control policy bound to the API is correct.
- Check whether the request throttling limit of the API has been reached. If no request throttling policy is created for an API, the API can be accessed 200 times per second by default. To change this limit of dedicated gateways, go to the **Gateway Information** page, click the **Parameters** tab, and modify the **ratelimit_api_limits** parameter.

5.2 Why Am I Seeing the Error Message "414 Request URI too large" When I Call an API?

The request URL (including request parameters) is too long. Place the request parameters in the request body and try again.

For details about API calling errors, see [Error Codes](#).

5.3 Why Am I Seeing the Error Message "The API does not exist or has not been published in the environment" When I Call an API?

If an open API in APIG failed to be called, troubleshoot the failure by performing the following operations:

1. The domain name, request method, or path used for calling the API is incorrect.
 - For example, an API created using the POST method is called with GET.
 - Missing a slash (/) in the access URL will lead to a failure in matching the URL in the API details. For example, URLs **http://7383ea59c0cd49a2b61d0fd1d351a619.apigw.region.cloud.com/test/** and **http://7383ea59c0cd49a2b61d0fd1d351a619.apigw.region.cloud.com/test** represent two different APIs.

2. The API has not been published. APIs can be called only after they have been published in an environment. For details, see [Publishing an API](#). If the API has been published in a non-production environment, check whether the **X-Stage** header in the request is the name of the environment.
3. The domain name is resolved incorrectly. If the domain name, request method, and path for calling the API are correct and the API has been published in an environment, the API may not be correctly resolved to the group to which the API belongs. For example, if you have multiple API groups and each group has an independent domain name, the API may be called using the independent domain name of another group. Ensure that the API is being called using the correct domain name.
4. Check whether the API allows OPTIONS cross-region requests. If yes, enable cross-origin resource sharing (CORS) for the API, and create an API that uses the OPTIONS method. For details, see [CORS](#).

5.4 Why Am I Seeing the Error Message "No backend available" When I Call an API?

- Check whether the backend service is accessible, and modify the backend service if it is inaccessible.
- Check the ECS security group configurations of the backend service and verify that the required port has been enabled.
- Check whether the backend service address is a public IP address. If yes, enable outbound access on the **Gateways > Access Console > Gateway Information** page.
- Check whether ACL configurations of the VPC restrict the communication between the API gateway and the subnet where the backend service is located.
- If you use a VPC channel, check whether the service port, health check port, and backend servers of the VPC channel have been correctly configured. If the VPC channel type is microservice, check whether a route has been added to the gateway. For details, see [Load Balance Channel](#). **Backends of the shared gateway do not support private network load balancers.**

5.5 Why Am I Seeing the Error Message "Backend unavailable" or "Backend timeout" When I Invoke the Backend Service?

The following table lists the possible causes if a backend service fails to be invoked or the invocation times out.

Possible Cause	Solution
The backend service address is incorrect.	Change the backend service address in the API definition. If the domain name is used, ensure that the domain name can be correctly resolved to the IP address of the backend service.
The timeout duration is incorrect. If a backend service fails to return a response within the configured timeout duration, APIG displays a message indicating that the backend service fails to be invoked.	Increase the timeout duration of the backend service or shorten the processing time in the API definition.
If the backend address is an ECS address, the security group to which the ECS belongs may block the request in the inbound or outbound direction.	Check the security group to which the ECS belongs and ensure that the inbound and outbound port rules and protocols of this security group are correct.
The request protocol is incorrect. For example, the backend service uses HTTP, but HTTPS is selected on APIG.	Ensure that the protocol of the created API is the same as that of the backend service.
The backend service URL is unreachable.	Check the URL.

5.6 Why Am I Seeing the Error Message "Backend domain name resolution failed" When I Invoke the Backend Service?

An error message indicating a domain name resolution failure is displayed when the backend service is called, although private domain name resolution is completed for the VPC where the API gateway is located.

Possible Cause

The VPC of the API gateway is isolated from that of the backend service. Private domain names can be resolved only for the VPC of the backend service.

Solution

- Method 1: When creating an API, set **Backend Address** to a public network domain name.
- Method 2: When creating an API, do not use a VPC channel (load balance channel). Instead, set **Backend Address** to the backend service IP address, and add a constant parameter to specify the **Host** field in the header.
- Method 3: When creating an API, specify a VPC channel (load balance channel).

- a. Create a VPC channel (load balance channel).
- b. Add the backend service address.
- c. When creating an API, select the VPC channel (load balance channel) and configure a custom header.

5.7 Why Am I Seeing the Error Message "Incorrect IAM authentication information" When I Call an API?

You may encounter the following errors related to IAM authentication information:

- [Incorrect IAM authentication information: verify aksk signature fail](#)
- [Incorrect IAM authentication information: AK access failed to reach the limit,forbidden](#)
- [Incorrect IAM authentication information: decrypt token fail](#)
- [Incorrect IAM authentication information: Get secretKey failed](#)

Incorrect IAM authentication information: verify aksk signature fail

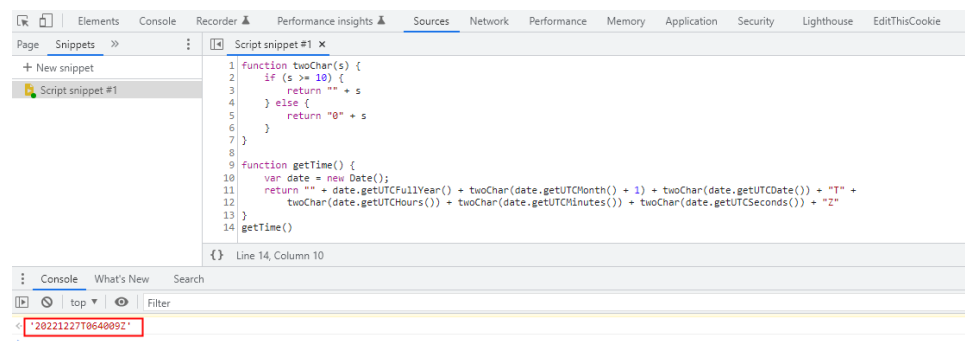
```
{
  "error_msg": "Incorrect IAM authentication information: verify aksk signature fail, .....",
  "error_code": "APIG.0301",
  "request_id": "*****"
}
```

Possible Cause

The signature algorithm is incorrect, and the signature calculated by the client is different from that calculated by APIG.

Solution

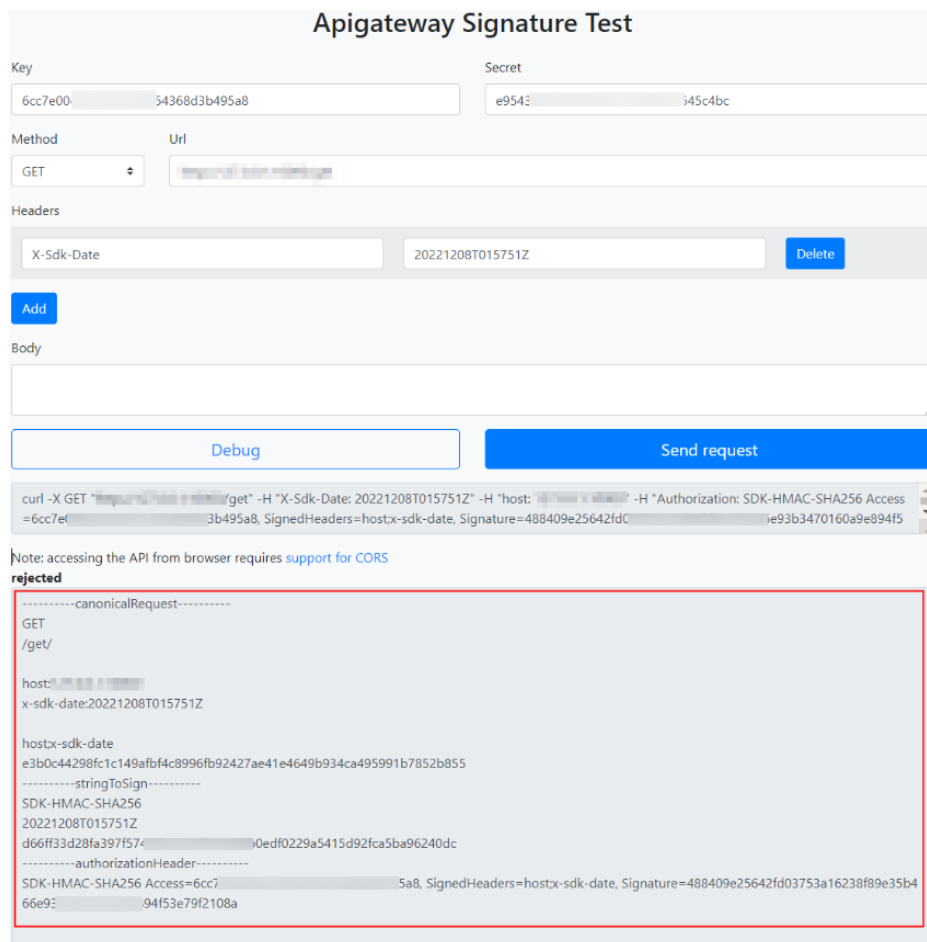
- Step 1** Download the [JavaScript SDK](#), view the visualized signing SDK, and obtain the signature.
- Step 2** Decompress the package and open the **demo.html** file using a browser.
- Step 3** Obtain the value of **x-sdk-date**, and check whether the difference between this value and the current time is within 15 minutes.
 1. Press **F12** on the keyboard, and choose **Sources** > **Snippets** > **New snippet**.
 2. Copy the following code to the script snippet on the right, right-click the snippet name on the left, and select **Run** from the shortcut menu. The value displayed on the **Console** tab is the value of **x-sdk-date**.



```
function twoChar(s) {
  if (s >= 10) {
    return "" + s
  } else {
    return "0" + s
  }
}

function getTime() {
  var date = new Date();
  return "" + date.getUTCFullYear() + twoChar(date.getUTCMonth() + 1) + twoChar(date.getUTCDate()) +
  "T" +
  twoChar(date.getUTCHours()) + twoChar(date.getUTCMinutes()) + twoChar(date.getUTCSeconds()) +
  "Z"
}
getTime()
```

Step 4 Add **x-sdk-date** to **Headers**, set other parameters, and click **Debug** to obtain the signature.



For all requests except get, delete, and head, add a body in the **Body** area by using the same format as a real request body.

Step 5 Copy the **curl** command in the figure of **Step 4**, run it in a command line interface, and then go to the next step.

```
curl -X GET "http://192.168.0.1:10000/get" -H "X-Sdk-Date: 20221208T015751Z" -H "host: 192.168.0.1:10000" -H "Authorization: SDK-HMAC-SHA256 Access=6cc7***95a8, SignedHeaders=host;x-sdk-date, Signature=4884***108a" -d "$"
```

If a custom authorizer is used, replace **Authorization** in the **curl** command with the authorizer name.

Step 6 Compare the signature in the local code with the visualized signature of JavaScript.

For example, check whether the values of **canonicalRequest**, **stringToSign**, and **authorizationHeader** in the Java signing code are the same as those in the visualized signature of JavaScript.

```
public void sign(Request request) throws UnsupportedEncodingException {
    String singerDate = getHeader(request, X_SDK_DATE);
    SimpleDateFormat sdf = new SimpleDateFormat( pattern: "yyyyMMdd'T'HHmmss'Z'");
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));

    if (singerDate == null) {
        singerDate = sdf.format(new Date());
        request.addHeader(X_SDK_DATE, singerDate);
    }
    addHostHeader(request);

    String messageDigestContent = calculateContentHash(request);

    String[] signedHeaders = getSignedHeaders(request);

    final String canonicalRequest = createCanonicalRequest(request, signedHeaders, messageDigestContent);

    final byte[] signingKey = deriveSigningKey(request.getSecret());

    String stringToSign = createStringToSign(canonicalRequest, singerDate);
    byte[] signature = computeSignature(stringToSign, signingKey);
    String signatureResult = buildAuthorizationHeader(signedHeaders, signature, request.getKey());

    request.addHeader(AUTHORIZATION, signatureResult);
}
```

----End

Incorrect IAM authentication information: AK access failed to reach the limit,forbidden

```
{
  "error_msg": "Incorrect IAM authentication information: AK access failed to reach the limit,forbidden." .....,
  "error_code": "APIG.0301",
  "request_id": "*****"
}
```

Possible Causes

- The AK/SK signature calculation is incorrect.
- The AK and SK do not match. Check whether the SK is correct.
- AK/SK authentication fails for more than five consecutive times, and the AK/SK pair is locked for five minutes. (Authentication requests are rejected within this period). Try again 5 minutes later.
- An expired token is used for token authentication. Obtain a new token.

Solution

- Resolve the problem by referring to [Incorrect IAM authentication information: verify aksk signature fail](#).
- Check whether the SK is correct.
- Try again 5 minutes later.
- Obtain a new token.

Incorrect IAM authentication information: decrypt token fail

```
{
  "error_msg": "Incorrect IAM authentication information: decrypt token fail",
  "error_code": "APIG.0301",
  "request_id": "*****"
}
```

Possible Cause

The token cannot be parsed for IAM authentication of the API.

Solution

- Check whether the obtained token is the token of the corresponding IAM account.
- Check whether the token is correct.
- Check whether the token has been obtained in the environment where the API is called.

Incorrect IAM authentication information: Get secretKey failed

```
{
  "error_msg": "Incorrect IAM authentication information: Get secretKey failed,ak:*****,err:ak not exist",
  "error_code": "APIG.0301",
  "request_id": "*****"
}
```

Possible Cause

The AK used for IAM authentication of the API does not exist.

Solution

Check whether the AK is correct.

5.8 Why Am I Seeing the Error Message "Incorrect app authentication information" When I Call an API?

You may encounter the following errors related to app authentication information:

- [Incorrect app authentication information: app not found, appkey xxx](#)
- [Incorrect app authentication information: verify signature fail, canonicalRequest](#)
- [Incorrect app authentication information: signature expired](#)

Incorrect app authentication information: app not found, appkey xxx

```
{
  "error_msg": "Incorrect app authentication information: app not found, appkey 0117***e5e1",
  "error_code": "APIG.0303",
  "request_id": "a532***5aca"
}
```

Possible Causes

The AppKey is incorrect.

Solution

Step 1 In the navigation pane of the APIG console, choose **API Management > Credentials**.

Step 2 Click the corresponding credential name to go to the details page.

Step 3 Check the **Key** and reconfigure the AppKey.

----End

Incorrect app authentication information: verify signature fail, canonicalRequest

```
{
  "error_msg": "Incorrect app authentication information: verify signature fail, canonicalRequest:GET|/test/|host:d7da***3df7.example.com|x-sdk-date:20230527T015431Z|host;x-sdk-date|e3b0c***52b855",
  "error_code": "APIG.0303",
  "request_id": "cb14***62dc"
}
```

Possible Causes

The signature algorithm is incorrect, and the signature calculated by the client is different from that calculated by APIG.

Solution

For details, see [Incorrect IAM authentication information: verify aksk signature fail](#).

Incorrect app authentication information: signature expired

```
{
  "error_msg": "Incorrect app authentication information: signature expired, signature time:20230527T000431Z,server time:20230527T020608Z",
  "error_code": "APIG.0303",
  "request_id": "fd65***b8ad"
}
```

Possible Causes

The difference between the client's signature timestamp **x-sdk-date** and the APIG server's time exceeds 15 minutes.

Solution

Check whether the time on the client is correct.

5.9 Why Doesn't Modification of the backend_timeout Parameter Take Effect?

Problem Description

Modification of the **backend_timeout** parameter in gateways does not take effect.

Possible Causes

The **Timeout (ms)** parameter on the **Define Backend Request** page is not modified.

Solution

Log in to the APIG console, go to the API details page, click **Edit**, and modify the **Timeout (ms)** parameter on the **Define Backend Request** page.

5.10 Does APIG Have a Limit on the Size of the API Request Body?

Shared gateway on the old console: APIG forwards only API requests whose body is no larger than 12 MB and rejects requests with a larger body. In this case, upload the request body to Object Storage Service (OBS).

Dedicated gateway: APIG forwards only API requests whose body is no larger than 12 MB. If your gateway will receive requests with a body larger than 12 MB, modify the **request_body_size** parameter on the gateway details page. This parameter indicates the maximum request body size allowed. The value ranges from 1 MB to 9536 MB.

5.11 Is There a Limit on the Size of the Response to an API Request?

No.

But there is a limit on the size of the request body. For details, see [request_body_size](#).

5.12 How Do I Call an API Using App Authentication in iOS System?

APIG provides SDKs and demos in multiple languages, such as Java, Python, C, PHP, and Go, for app authentication.

To use Objective-C (for iOS) or other languages, see [App Authentication](#).

5.13 Why Can't I Create a Header Parameter Named x-auth-token for an API Called Through IAM Authentication?

The header parameter **x-auth-token** has already been defined in APIG.

To use this parameter to call an API, add the parameter and its value to the request header.

5.14 Can Mobile Apps Call APIs?

Yes, mobile apps can call APIs.

In app authentication mode, the AppKey and AppSecret of a mobile app are replaced with those in the relevant SDK to sign the app.

5.15 Can Applications Deployed in a VPC Call APIs?

Yes, applications deployed in a VPC can call APIs by default. If domain name resolution fails, configure a DNS server on the current endpoint by following the instructions in [Configuring an Intranet DNS Server](#). After the configuration, applications deployed in the VPC can call APIs.

Configuring an Intranet DNS Server

To configure a DNS server, specify its IP address in the `/etc/resolv.conf` file.

The IP address of the intranet DNS server depends on which region you are located in. Find the IP address of the intranet DNS server in your region from [private DNS server addresses](#).



Add an intranet DNS server with either of the following two methods:

- Method 1: Modify the subnet information of the VPC.
- Method 2: Edit the `/etc/resolv.conf` file.

The intranet DNS server configurations become invalid after the ECS restarts, and the intranet DNS server must be configured again. Therefore, method 1 is recommended.

Method 1: Modify the subnet information of the VPC.

Perform the following procedure to add a DNS server IP address to the subnet configurations of the ECS in the VPC.

- Step 1** Click  in the upper left corner to select a region.
- Step 2** In the service list, choose **Compute > Elastic Cloud Server**.
- Step 3** Click the name of the ECS you want to use.
- Step 4** On the **Network Interfaces** tab page, click  to view the subnet name.
- Step 5** On the **Summary** tab page, view the VPC name.
- Step 6** Click the VPC name to visit the VPC console.
- Step 7** Choose **Subnets** in the left navigation pane.
- Step 8** Locate the subnet mentioned in [Step 4](#) and click the subnet name.
- Step 9** Change the DNS server address of the subnet and click **OK**.

For example, change the address to **100.125.1.250**.

- Step 10** Restart the ECS. Check that the `/etc/resolv.conf` file contains the IP address of the DNS server to be configured, and the IP address is less than those of all other DNS servers.

The following figure shows the IP address **100.125.1.250** of the DNS server to be configured.

```
# Generated by NetworkManager
search openstacklocal
nameserver 100.125.1.250
nameserver 114.114.115.115
```

Modifying the subnet information of a VPC will affect all ECSs created using the subnet.

----End

Method 2: Edit the `/etc/resolv.conf` file.

Add the IP address of the intranet DNS server to the `/etc/resolv.conf` file.

For example, if you are located in **CN-Hong Kong**, add an intranet DNS server of IP address **100.125.1.250** to the `/etc/resolv.conf` file.

- The IP address of the new DNS server must be less than those of all other DNS servers.
- The DNS configurations take effect immediately after the `/etc/resolv.conf` file is saved.

5.16 Does APIG Support WebSocket Data Transmission?

Yes.

When creating an API, you can select HTTP, HTTPS, or HTTP&HTTPS. HTTP is equivalent to WebSocket (ws), and HTTPS is equivalent to WebSocket Secure (wss).

5.17 How Will the Requests for an API with Multiple Backend Policies Be Matched and Executed?

If multiple backend policies are configured for an API, APIG will match the backend policies in sequence. If an API request matches one of the backend policies, APIG immediately forwards the request to the corresponding backend and stops matching.

If no backend policy is matched, the API request is forwarded to the default backend server.

5.18 How Can I Access Backend Services over Public Networks Through APIG?

Enable [public access](#) to allow external services to call APIs.

If you encounter a network problem when calling APIs, see [What Are the Possible Causes for an API Calling Failure?](#)

6 API Authentication

6.1 Does APIG Support Two-Way Authentication?

Dedicated gateway: Yes.

- Frontend two-way authentication: When binding an independent domain name, select an [SSL certificate](#) that contains a CA certificate. Client authentication, that is, two-way authentication, can be enabled.
- Backend two-way authentication: When creating an API, enable two-way authentication for the backend service. For details, see the two-way authentication parameter description in [Creating an API](#).

Shared gateway on the old console: No. Only HTTPS one-way authentication is supported.

6.2 Will the Request Body Be Signed?

Yes. The request body is another element that needs to be signed in addition to the mandatory request header parameters. For example, when an API used to upload a file using the POST method is called, the hash value of the file to upload is calculated to generate a signature.

For details about signatures, see [the signature authentication algorithm description](#).

6.3 API Credential FAQs

How many apps (credentials) can I create?

You can create a maximum of 50 apps (credentials).

How do I isolate the calling information among the third parties that call the same API through app authentication?

Create multiple apps (credentials) for different third parties and bind the apps (credentials) to the same API.

Are there any restrictions on the maximum number of third parties that can call the same app through app authentication?

No restrictions.

Do I need to create an app (credential) for an API so that it can be called through app authentication?

Yes, you need to create an app (credential) and bind it to the API. After the app (credential) is created, an AppKey and AppSecret are automatically created. Provide the AppKey and AppSecret for third parties to call the API.

How can an API be called by third parties through app authentication?

Provide third parties with the AppKey and AppSecret of the app you have created for accessing the API. The third parties then can use the AppKey and AppSecret to call the API through an SDK. For details about how to use an SDK, see [Calling APIs Through App Authentication](#).

7 API Policies

7.1 Can I Configure the Maximum Number of Concurrent Requests?

No,

but you can limit the maximum number of API calls allowed within a specific period of time.

7.2 Is the Restriction of 1,000 Requests Per Day to a Debugging Domain Name Applied to Enterprise Accounts?

Yes.

For details about the subdomain name (debugging domain name), see [Configuring the Domain Name for Calling APIs](#).

7.3 Does APIG Have Bandwidth Limits?

The shared gateway does not have limits on the bandwidth. It throttles requests based on request throttling policies and limits the maximum body size to 12 MB.

Dedicated gateways have bandwidth limits. When you create a dedicated gateway, you can set the bandwidth for public inbound and outbound access.

7.4 Why Doesn't a Request Throttling Policy Take Effect?

- If the request throttling for API or source IP address does not take effect, check whether the request throttling policy is bound to the API.

- If the user request throttling does not take effect, check whether the authentication mode of the API is App or IAM.
- If the credential request throttling does not take effect, check whether the API uses App authentication.

7.5 How Do I Provide an Open API to Specific Users?

You can provide an open API to specific users in either of the following ways:

- Select app authentication when you create the API, and share the AppKey and AppSecret with the target users.
- Configure an access control policy to allow access from specific IP addresses or account names, and bind the access control policy to the API.

7.6 Are Client IP Addresses Verified for Access Control?

Not necessarily.

In APIG, access control is based on the value of **\$remote_addr**. **\$remote_addr** indicates a client IP address and is determined by the access mode. If a client accesses APIG without using any proxy, **remote_addr** is the client's IP address. If a client accesses APIG using a proxy, the client first accesses the proxy, and the proxy then forwards the request to APIG. In this case, **remote_addr** is the proxy's IP address.

8 Importing and Exporting APIs

8.1 What Are the Possible Causes of an API Import Failure?

- Possible cause 1: The number of APIs exceeds the maximum allowed limit for a single import. For more APIs (300), import them in batches or submit a service ticket to increase the limit.
- Possible cause 2: Parameters are incorrect. Check and rectify the parameters. You are advised to create an API on the APIG console, export it, and then use it as a template for importing APIs.
- Possible cause 3: The YAML file is in incorrect format. Check and modify the file.
- Possible cause 4: The local proxy network has restrictions. Change the network environment.
- Possible cause 5: The header of the API request contains **X-Auth-Token**. Remove **X-Auth-Token** from the header.

8.2 Is There a Template for Importing APIs Using a Swagger File?

The template is being developed.

Currently, you can configure one or two APIs in APIG, and then export them to use as templates.