# Alibaba Cloud

API Gateway Create API

Document Version: 20220513

C-J Alibaba Cloud

# Legal disclaimer

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

- You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloudauthorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
- 2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
- 3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
- 4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
- 5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud and/or its affiliates Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
- 6. Please directly contact Alibaba Cloud for any errors of this document.

# **Document conventions**

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

# Table of Contents

1.Overview	05
2.Create an API	07
3.Enable API services	14
4.Bind a domain name to an API group	16
5.Import Swagger files to create APIs	23
6.Use a resource in a VPC as the backend service of an API ope	43
7.Use Function Compute as the backend service of an API opera	47
8.Use Mock as the backend service of an API operation	59
9.Create API operations for data services	62
10.Create API operations for an online prediction service	63
11.Parameter mapping and verification rules	64
12.Version management	77
13.Configure different environments for an API operation	82
14.Enable HTTP/2 for an API group	90
15.Manage models	92
16.Implement bidirectional communication	95

# 1.0verview

# 1 Methods to create an API

You can use one of the following methods to create an API:

# • Create an API in the API Gateway console

Log on to the API Gateway console. In the top navigation bar, select a region. In the left-side navigation pane, choose Publish APIs > APIs. On the API List page, click Create API. For more information, see Create an API.

# • Create an API by using OpenAPI Explorer

You can create an API by using OpenAPI Explorer. For more information, see Create an API.

# • Create an API by importing a Swagger file

Swagger is a specification used to describe API definitions. It is widely used to define and describe APIs for backend services. You can import a Swagger 2.0 file to create an API in the API Gateway console. You can use one of the following methods to import a Swagger file:

Use OpenAPI Explorer. For more information, see Import Swagger.

Perform operations in the API Gateway console. For more information, see Import a Swagger file to create an API.

# 2 API Gateway instance and API group

# 2.1 API Gateway instance

An API Gateway instance is composed of a group of resources used to access and process your APIs. The resources include public IP addresses, private IP addresses, public egress IP addresses, and load balancing configurations. API Gateway supports shared instances and dedicated instances. These two types of instances are suitable for different scenarios.

- Dedicated instances provide high performance and strict service level agreement (SLA) compliance. These instances are suitable for production environments.
- Shared instances are suitable for development, test, and assessment environments, as well as small-scale production environments.

For more information about the differences between the two types of instances, see Instance type and guidance for selection.

# 2.2 API group

An API group is a management unit of APIs. Before you create an API, you must create an API group. When you create an API group, the system allocates a second-level domain to the API group. The second-level domain is used only for test. You can use the second-level domain to call APIs from the client. However, you can access this domain for a maximum of 1000 times per day. We recommend that you bind an independent domain that is filed in Alibaba Cloud to your API group. Then, you can use the domain to call APIs in your API group.

For more information about how to bind a domain to an API group, see Bind a wildcard domain name to an API group.

# 3 Steps to create an API

3.1 Configure basic information

Configure the basic information for the API that you want to create. The information includes Group, API Name, Security Certification, API Options, and Description.

#### 3.2 Define an API request

Define an API request. The detailed configurations include Request Type, Protocol, Custom Domain Name, Request Path, HTTP Method, Request Mode, and Input Parameter Definition.

API Gateway supports the HTTP, HTTPS, and WebSocket protocols.

#### 3.3 Define an API backend service

Define mappings between request and response parameters and configure information of the backend service of your API. The information includes Backend Service Type, Backend Service Address, Backend Request Path, HTTP Method, Backend Timeout, Backend Service Parameter Configuration, Constant Parameter, and System Parameter. Assume that parameter mapping rules are configured for the current API. After API Gateway receives a request for the API, API Gateway maps the request parameters based on the mapping rules before it routes the request to the backend service.

The backend services supported by API Gateway include HTTP (s) Service, Function Compute, VPC, and Mock.

HTTP (s) Service: Create an API with HTTP as the backend service.

Function Compute: Use Function Compute as the backend service of the API.

VPC: Use a resource in a VPC as the backend service of the API.

Mock: Use Mock as the backend service of the API.

#### 3.4 Define response information

Configure Content Type of Response, Sample of Returned Results, Sample of Returned Failure, and Error Code Definition.

# 4 Configure environments for the API

API Gateway allows you to configure the test (Test), staging (Pre), and production (Release) environments for an API group. This meets your requirement for API calls in different research and development scenarios. For example, in the test environment, the API uses appropriate resources as its backend service. In this way, testers can simulate API calls in the production environment.

For more information, see Configure different environments for an API.

# 2.Create an API

This topic describes how to create and debug an API and perform security configurations for the API. To create an API, you must configure basic information, request information, a backend service, and response information for the API. After you debug the API, you can publish the API for users to call.

# 1 Create an API

Log on to the API Gateway console. In the left-side navigation pane, choose Publish APIs > APIs. On the API List page, click Create API. Then, you can create and configure an API.

1.1	Conf	igure	basic	inf	formation	

Parameter	Description
Group	The basic management unit of APIs. Before you create an API, you must create an API group. When you select an API group, the region where the API group is created is selected for the API.
API Name	The name of the API that you want to create. Each API name in an API group must be unique.
	The authentication mode of API requests. Valid values: Alibaba Cloud APP, OpenID Connect, OpenID Connect & Alibaba Cloud APP, and No Certification.
	• Alibaba Cloud APP: This mode requires the requester to pass the application authentication to call an API.
Security Certification	• <b>OpenID Connect</b> : OpenID Connect is a simple authentication layer on top of OAuth 2.0. It is an authorization framework that implements identity interaction and authorization by using RESTful APIs. If you select this mode, the OpenID Connect authentication layer is integrated with API Gateway and the backend service of the current API to implement authentication. For more information, see Implement OpenID Connect authentication by using API Gateway.
	• <b>OpenID Connect &amp; Alibaba Cloud APP</b> : Both OpenID Connect and Alibaba Cloud APP are used for user identity authentication during API calls.
	• No Certification: This mode allows all users who understand the request definition of the API to initiate a request. API Gateway directly forwards the request to the backend service without verifying the identity of the requester. We recommend that you do not set Security Certification to No Certification.
Signature Method	<ul> <li>The algorithm that is used to sign API requests. Valid values:</li> <li>HmacSHA256</li> <li>HmacSHA1 and HmacSHA256: If you set the parameter to this value, both the algorithms are supported.</li> </ul>

Parameter	Description
Description	The description of the API.

# 1.2 Define an API request

Define an API request. The detailed configurations include Protocol, Request Path, HTTP Method, Request Mode, and Input Parameter Definition.

Parameter	Description
Request Type	The request type. Valid values: COMMON: indicates common HTTP or HTTPS requests. REGISTER (WEBSOCKET): indicates the bidirectional control signaling used to register devices. It is sent from the client to the server. UNREGISTER (WEBSOCKET): indicates the bidirectional control signaling used to deregister devices. It is sent from the client to the server. After a device is deregistered, it no longer receives server-to-client notifications. NOT IFY (WEBSOCKET): After receiving the registration signaling sent from the client, the backend service records the device ID and sends a server-to-client notification to API Gateway. Then, API Gateway sends the notification to the device. If the client is connected, API Gateway can send the server-to-client notification to the client.
Protocol	The network protocol that is used to initiate the request. Valid values: HTTP, HTTPS, and WEBSOCKET.
Request Path	The API request path that corresponds to the service host. The request path can be different from the actual backend service path. You can specify a valid path with correct semantics as the request path. You can configure dynamic parameters in the request path. This requires users to specify path parameters in the request. At the same time, the path parameters can be mapped to the query and header parameters that are received by the backend service.
HTTP Method	The standard HTTP request method. Valid values: PUT, GET, POST, DELETE, PATCH, HEAD, OPTIONS, and ANY.

Parameter	Description
	Valid values: Request Parameter Mapping (Filter Unknown Parameters), Request Parameter Mapping (Passthrough Unknown Parameters), and Request Parameter Passthrough.
	• Request Parameter Mapping (Filter Unknown Parameters): indicates that you must configure request and response data mappings for query, path, and body form parameters. API Gateway transparently passes only the configured parameters to the backend service. Other parameters are filtered out.
Request Mode	• Request Parameter Mapping (Passthrough Unknown Parameters): indicates that you must configure request and response data mappings for query, path, and body form parameters. API Gateway maps and verifies only the configured parameters and transparently passes unknown parameters to the backend service.
	• <b>Request Parameter Passthrough</b> : indicates that you do not need to configure query and body form parameters. However, you still need to configure path parameters. API Gateway transparently passes all parameters sent from the client to the backend service.

**Input Parameter Definition**: The request parameters of your API. You can specify different request parameters for different parameter paths. You can select Head, Query, Body, or Parameter Path from the Param Location drop-down list. When you configure a dynamic path parameter, you must provide a description of this dynamic parameter in the Input Parameter Definition section. The following data types are supported: String, Int, and Boolean.

- Note that the names of all parameters must be unique.
- You can use the upwards and downwards arrows in the Order column to adjust the parameter order.
- To delete a parameter that is no longer required, you can click **Remove** in the Operation column.

# ? Note

If you have configured a dynamic parameter in the request path, you must define a parameter with the same name as the dynamic parameter and set Param Location to Parameter Path.

**Configure parameters verification rules**: Find the parameter for which you want to configure verification rules and click **More** in the Operation column. For example, you can specify Max Length and Enumeration. API Gateway pre-verifies requests based on the verification rules. Requests with invalid parameters are not sent to your backend service. This reduces the workload on your backend service.

# 1.3 Define an API backend service

Define mappings between the request and response parameters and configure the information of the backend service of your API. The information includes Backend Service Address, Backend Request Path, Backend Timeout, and Backend Service Parameter Configuration, Constant Parameter, and System Parameter. After receiving a request, API Gateway converts the format of the request into the format that is required by your backend service based on the backend service configuration. Then, API Gateway forwards the request to your backend service.

# Define a backend service

Parameter	Description
Backend Service Type	<ul> <li>Valid values: HTTP(s) Service, Function Compute, VPC, and Mock.</li> <li>HTTP(s) Service: This option is selected by default. It indicates that API Gateway accesses the backend service over HTTP or HTTPS. If API Gateway can directly communicate with the backend service, select this option. To configure an HTTPS service as the backend service, you must first obtain a Secure Sockets Layer (SSL) certificate.</li> <li>Function Compute: If you use Function Compute as the backend service, you must create a function in the Function Compute console and obtain Function Compute Role Arn on the Functions tab. Then, specify Service Name and Function Name.</li> <li>VPC: If the backend service is deployed in a VPC, select this option.</li> <li>Mock: If you want to simulate expected return results, select this option.</li> </ul>
VPC Access Name	If your backend service is deployed in a VPC, you must specify a VPC access name.
Backend Service Address	The host name of the backend service. It can be a domain name or a value in the format of http(s)://host:port. The address must start with http:// or https://.
Backend Request Path	The actual request path of your API on your backend server. If you want to receive dynamic parameters in the backend path, you must declare parameter mappings by specifying the locations and names of the corresponding request parameters.
HTTP Method	The standard HTTP request method. Valid values: PUT, GET, POST, DELETE, PATCH, HEAD, OPTIONS, and ANY.
Backend Timeout	The response time for API Gateway to access the backend service after API Gateway receives an API request. The response time is from the time when API Gateway sends an API request to the backend service to the time when API Gateway receives a response returned by the backend service. The response time cannot exceed 30 seconds. If API Gateway does not receive a response from the backend service within 30 seconds, API Gateway stops accessing the backend service and returns an error message.

**Backend Service Parameter Configuration**: API Gateway can set up mappings between the request and response parameters, including a name mapping and a location mapping. API Gateway can map a path, header, query, or body request parameter to a response parameter at a different location. This way, you can package your backend service into a standardized and professional API form. This part declares the mappings between the request and response parameters.

**Constant Parameter**: You can configure constant parameters for API requests. The constant parameters include Backend Param Name, Param Value, Param Location, Description, and Operation. Constant parameters are invisible to your users. After API Gateway receives requests, API Gateway adds these parameters to the specified locations in each request and forwards the requests to your backend service. If you want API Gateway to attach the abc parameter as a constant parameter. After API Gateway receives a request, API Gateway automatically adds the parameter to the specified location and forwards the request to your backend service.

**System Parameter**: the system parameters that apply globally in API Gateway. By default, API Gateway does not add system parameters to API requests. If you want to obtain the system parameters, you can configure their locations and names. The following table describes the system parameters.

Parameter	Description
CaClientIp	The IP address of the client that sends an API request. If you have configured Web Application Firewall (WAF) or Content Delivery Network (CDN), the system records their origin IP address. You must check the IP address of the client in X-Forwarded- For.
CaDomain	The domain name that is used to send an API request.
CaRequestHandleTime	The request time. It must be in GMT.
CaAppld	The ID of the application from which an API request is sent.
CaRequestId	RequestId
CaApiName	The name of the API.
CaHttpSchema	The protocol that is used to call the API. Valid values: HTTP and HTTPS.
CaProxy	The proxy that is used to handle an API request. The value must be set to AliCloudApiGateway.

Parameter	Description
CaClientUa	The user agent of the client that sends an API request.
CaCloudMarketInstanceld	The ID of the instance of the service that is requested.
СаАррКеу	The key of the application from which an API request is sent.
CaStage	The environment in which you call the API. Valid values: RELEASE, TEST, and PRE.

You can configure **Passthrough HOST Head (Head Domain)** for a shared instance on the Group Details page of an API group. If you want to perform the same configuration for a dedicated instance, submit a ticket. After you enable Passthrough HOST Head (Head Domain), API Gateway transparently passes the domain name of an API request to your backend service. Otherwise, API Gateway passes the host name that you specified in API Gateway to your backend service.

For example, if you attach a host name xuemeng.XXXX.com to an API group, the host name of the backend service of APIs in the API group is apigatewayXXXXXalicloudapi.com:8080. The following figures show different host names received by your backend service before and after Passthrough HOST Head (Head Domain) is enabled.

#### Passthrough HOST Head (Head Domain) is enabled:



Passthrough HOST Head (Head Domain) is disabled:



# ? Note

When you configure parameters for the backend service of the API, make sure that the name of each parameter is globally unique in API Gateway. This applies to dynamic parameters in request paths, header parameters, query parameters, body parameters, constant parameters, and system parameters. Note that values of body parameters must be non-binary. For example, if you specify a header parameter and a query parameter with the same **name**, an error will occur after you send a request.

# 1.4 Define response information

Configure Content Type of Response, Sample of Returned Results, Sample of Returned Failure, and Error Code Definition.

# 2 Debug the API

After you create and configure the API, you can debug the API on the API debugging page. This ensures the usability of the API.

After the API is created and configured, you are navigated to the **API List** page. To debug the API, perform the following steps:

- 1. Find the API that you created and click its name or click **Manage** in the **Operation** column to go to the **API Definition** page.
- 2. In the left-side navigation pane of the API Definition page, click Debug API.
- 3. On the page that appears, specify request parameters in the Request Parameters section and click **Send Request**.

A result appears on the right side of the page.

If a success response is returned, the API can be called.

If an HTTP status code 4XX or 5XX is returned, an error occurred during the API call. For more information, see How to obtain the error message and Error code table.

# 3 What to do next

After you configure and debug the API, the creation of the API is complete. You can publish the API to the test, staging, or production environment for further debugging or for users to call. You can also perform security configurations for the API. For example, you can attach the Request signature documentation for the API.

# **3.Enable API services**

# **Enable API services**

This section provides information you must understand for the API group and domain name before you enable API services.

# API group

An API group is the management unit of APIs. You must create a group before creating an API. The group consists of four attributes: name, description, region, and domain name. Note that:

- The group region is fixed once selected.
- Each account can have up to 50 API groups and each API group can have up to 200 APIs.
- When you create a group, the system assigns the group a second-level domain name to test your API. To enable the API service, you must bind the group to an independent domain name filed on Alibaba Cloud and resolve the CNAME of the independent domain name to the second-level domain name of the group. Up to five independent domain names can be bound to a group.

# Domain name and certificate

API Gateway locates the unique API group through the domain name, and the unique API through the Path+HTTPMethod. Before enabling API services, you must know the second-level domain name and independent domain name as follows:

- The unique and fixed second-level domain name is assigned by the system during group creation. By default, a second-level domain name is used to call the API only in the test environment under a small amount of traffic.
- An independent domain name is used for enabling API services. You can bind up to five independent domain names to a group. When configuring independent domain names, pay attention to the following points:
  - Resolve the CNAME of an independent domain name to the API second-level domain name of the group before binding the API group and domain name.
  - Verify the domain name within one day. Otherwise, the unprocessed binding request is automatically withdrawn by the system.
  - If a domain name is already bound to another group, resolve the domain name to the second-level domain name of the to-be-bound group before binding. Otherwise, the binding fails.

If your API supports the HTTPS protocol, you must upload the SSL certificate of the domain name by entering the parameters on the **Group Details** page, including the name, content, and private key.

# Test, production, and authorization

To test or enable the API, authorization is indispensable. Authorization means granting an app the permission to call an API. Note that:

- You can authorize the created app and access the second-level domain name to call the API.
- You can authorize the apps of customers to access the independent domain name to call your API service.
- Only an authorized app can call the API.

Now you have successfully enabled your API service. From creating the API to enabling it, you can create, modify, delete, view, test, release, remove, authorize, and revoke the authorization of an API. You can also view the release history and switch the version.

# 4.Bind a domain name to an API group

This topic describes how to bind your domain name to an API group hosted on API Gateway. This way, you can use your domain name to call APIs in the API group and provide external services.

# 1. Overview

# 1.1 Relationships between domain names and API groups and between domain names and APIs

You must bind your domain name to an API group hosted on API Gateway to establish a mapping. When API Gateway receives an HTTP request from a client, API Gateway identifies the API group to which the request belongs based on the domain name in the request, and determines a unique API in the API group based on HTTP Method and Request Path. API Gateway provides a default Internet second-level domain for each API group. A client can call the Internet second-level domain for a maximum of 1,000 times per day. By default, a response that is generated when an Internet second-level domain is called contains the "Content-Disposition: attachment; filename=ApiResponseForInnerDomain" information in the header. When you publish APIs in a production environment, you must bind an independent domain to the destination API group. The number of API calls is not limited for independent domains.

# 1.2 ICP filing

If you want to bind an independent domain to an API group in a region inside mainland China, you must apply for an ICP filing or access the independent domain at Alibaba Cloud ICP filing. If you want to bind an independent domain to an API group in a region outside mainland China, ICP filing is not required.

# 1.3 Ownership verification of domain names

If your domain name has been bound to an API group by other users, or conflicts with a wildcard domain name that has been bound, the domain name must pass ownership verification. Otherwise, the domain name cannot be bound. You can use one of the following methods to verify the ownership of a domain name:

- 1. Add a CNAME record for your domain name to the Internet second-level domain provided by API Gateway.
- 2. Add a TXT record that contains the following information for your domain name: record name in the format of API Group ID.Domain name and record value in the format of apigateway-domain-verification=Internet second-level domain. Example:

The ID of a specific API group is b7eb2f79e64f4431b08bbb948ed2567e. The Internet second-level domain that is provided for the API group is b7eb2f79e64f4431b08bbb948ed2567e-cn-hangzhou.alicloudapi.com. The domain name that is bound to the API group is a single domain name, such as youdomain.com, or a wildcard domain name, such as \*.yourdomain. You can add a record whose record type is TXT, record name is b7eb2f79e64f4431b08bbb948ed2567e.yourdomain.com, and record value is apigateway-domain-verification=b7eb2f79e64f4431b08bbb948ed2567e-cn-hangzhou.alicloudapi.com for the domain name.

# 2. Bind a single domain name

Perform the following steps to bind your single domain name to an API group:

- 1. Optional. Add a CNAME record for your single domain name to the Internet second-level domain provided for a specific API group.
- 2. Bind your single domain name to the API group on the Group Details page of the API Gateway console.

If the operation in Step 1 fails, you can also perform Step 2 to bind your single domain name to the API group. During the binding process, if no domain name conflict occurs, API Gateway does not check whether a CNAME record is added for your single domain name. If you bind your single domain name to an API group with no CNAME record added, a domain name request from a client cannot be routed to API Gateway.

# 2.1 Add a CNAME record for your single domain name

Step 1 On the Group Details page of the API Gateway console, find the Internet second-level domain provided for the destination API group.

Group Details				Refresh
Basic Information			Turn on cloud monitoring Api List	Modify Group Message
Region: China North 2 (Beijing)	Group Name: testFunctionGroup	Group ID:		
Subdomain Name	Internet Subdomain  The subdomain is only for API test, when the client directly calls it, there will be 1000 access restrictions per day. It is recommended to use the independent domain name for group binding, and it will not be subject to this restriction. For details, see <u>configuration process</u> ) API gateway set: Not activate the Ast advated <b>Desse</b> advates on the instance first VPC Intranet Subdomain: Not activated Please set Visit to VPC' in 'Instance'		Disable Internet Subdomain	
Instance Type: Dedicated VPC Instance ID: a Instance Name: testdtrac	Group Traffic Limit (QPS): 2500 (Consistent with the dedicated instance)	Modify API Group's Instance	Instance Type And Selection Guide	
Network Access Policy	HTTPS Security Policy: HTTPS2_TLS1_0 HTTPS Security Policy Documentation (Be consistent with the dedicated instance HttpsPolicy)			
Legal Status: NOFMAL				
Description:				

Step 2 Log on to your DNS management platform. If you use Alibaba Cloud DNS, visit https://dns.console.aliyun.com. On the Manage DNS page of the Alibaba Cloud DNS console, click your single domain name to go to the DNS Settings page.

Step 3 Click Add Record. In the Add Record panel, set Type to CNAME, Host to test, and Value to the Internet second-level domain that you obtained in Step 1, and then click Confirm.

Add Record		)	×
Type :			
CNAME- Canonical name	/		
Host:			
yichao		?	
ISP Line :			
Default - Return to the default value when the query is not matched to any view.	/	?	
* Value :			
-cn-beijing.alicloudapi.com			
* TTL:			
1 second(s)	/		



# 2.2 Bind your single domain name

Step 1 Log on to the API Gateway console. In the left-side navigation pane, choose Publish APIs > API Groups. On the Group List page, click the API group to which you want to bind your single domain name. The Group Details page appears. Step 2 In the lower-right corner of the Group Details page, click Bind Domain.

			Bind Domain
WebSocket Channel Status	Domain Legal Status	SSL Certificate	Operation
You have	not bound a domain name		
	WebSocket Channel Status You have	WebSocket Channel Status         Domain Legal Status           You have not bound a domain name	WebSocket Channel Status     Domain Legal Status     SSL Certificate       You have not bound a domain name     SSL Certificate

Step 3 In the Bind Domain Name dialog box, set Domain Name to test.yourdomain.com and click OK.

Bind Domain Name		>
Make sure the custom do group. Otherwise it will no Up to 5 domain names ca	omain name to bind has already been resolved to the subdomain name of this o <mark>t be invoked after binding. View subdomain name</mark> an be bound to a group.	
Group Name:	testFunctionGroup	
*Domain Name:		
	The VPC instance already supports the function of extensive domain name. Bind the extensive domain name in the format of `*.api.foo.com`. Click to open the help document	
Stage:	Default(Use X-Ca-Stage to determine the st	
	For more information about environmental management, Click to open the help document	
	OK Cance	1

# 2.3 Troubleshooting

Causes of domain name binding failures and solutions:

- The domain name that you want to bind has already been bound to another API group, or is in range conflict with another domain name that you have bound. The range conflict indicates that a wildcard domain name overlaps a single domain name. In this case, you must unbind the domain name, and then bind the domain name to the destination API group.
- The domain name that you want to bind has already been bound to an API group created by a different user, or is in range conflict with another domain name that you have bound. In this case, you must follow the instructions described in the "Ownership verification of domain names" section to verify the ownership of the domain name.

# 2.4 Make API calls

After the binding is complete, you can use this domain name to call an API in the API group. The following example shows how to call an API by using cURL.

```
curl http://yourdomain.com/apipath -i
HTTP/1.1 200 OK
Date: Mon, 23 Mar 2020 08:40:01 GMT
Connection: keep-alive
Keep-Alive: timeout=25
Server: Jetty(7.2.2.v20101205)
X-Ca-Request-Id: E2B8CBAB-D6EF-4576-838F-44DDC1A6B20D
```

# 3. Wildcard domain names

# 3.1 Support for wildcard domain names

API Gateway allows you to bind wildcard domain names to API groups. You can resolve a wildcard domain name to API Gateway and bind the wildcard domain name to your API group in the API Gateway console. After the binding is complete, you can use the wildcard domain name to call APIs in the API group hosted on API Gateway. Assume that you are the owner of the domain name abc.com. If you want to resolve all subdomains, such as 1.abc.com and 2.abc.com of abc.com, to API Gateway to provide external services, perform the following steps:

- 1. On your DNS management platform, use a CNAME record to resolve \*.abc.com to the Internet second-level domain of the destination API group.
- 2. On the Group List page of the API Gateway console, bind \*.abc.com to the destination API group.

After the binding is complete, the client can access APIs in the API group by using one of the subdomains of abc.com. For example, if an API in the API group can be anonymously called by using the GET method, the API can also be called by using the subdomains of \*.abc.com.



Only the instances that are deployed in a VPC support wildcard domain names.

# 3.2 Bind a wildcard domain name

The process of binding a wildcard domain name is similar to that of binding a single domain name. The differences between these two processes lie in the following aspects:

- 1. When you bind a wildcard domain name, you must verify the ownership of the wildcard domain name. For more information, see the "Ownership verification of domain names" section.
- 2. After a wildcard domain name is bound, you must configure the wildcard domain name template on the Group Details page. Then, you can use the wildcard domain name to call APIs.

	Instance Type: Shared Instance Instance ID: api-shared-vpc-002	(VPC) 2	Maximum C (To increase	PPS: 500 the quota, <u>purchase a dedicated instance</u> ,)	Modify Inst	ance for API Group Deployment	Instance types
	Network Access Policy	Set	Wildcard Domain Na	groupId].api.foo.com	×	TPS Security Policy HTTPS Security Policy	y Description
	Validity: NORMAL		Name Pattern	{userId}.user.apl.foo.com			
10	Description:			Wildcard Domain Name Documentation	11		
	Independent Domains			(Use carriage returns, commas (,), or semicolons (;) to separa	ate domain		Bind Domain Name
	Independent Domain Netwo	vork Type		name patterns.)			
	*.fredhuang.com Inter	nal Network				Vildcard Domain Name Pattern Change Env	vironment
	Custom Log Tracing			Confirm	Cancel		Modify
	*Tracing Field Position		\$				e
	Tracing Field Name						
	After you customize a tracing field na	ld name, API Gate me.	way passes it to the l	packend service and records it in the CustomTraceId field of the	ne API caller. If you	do not customize a tracing field name, API Ga	ateway uses the generated

The wildcard domain name template is used to configure domain name parameters. Variable fields in the template can be transmitted to the backend service as parameters.

# 4. Configure a default domain name

API Gateway allows you to upload an HTTPS certificate for your domain name so that you can use the domain name to call APIs over HTTPS. If multiple domain names are bound to an API group, and all these domain names support HTTPS-based API calls, you must configure a default domain name. This way, API Gateway can return the certificate for the default domain name when it receives an SSL handshake request from a client that does not support SNI. If no default domain name is configured, API Gateway returns the certificate for a domain name randomly. The configuration of a default domain name applies only to dedicated instances. By default, shared instances do not support the certificate for a default domain name. If a client of an earlier version that does not support SNI makes API calls over HTTPS, a certificate confusion problem may occur.

You can configure a default domain name in the Set HTTPS default domain name (dedicated instance only) section on the Group Details page, as shown in the following figure.

# Create API Bind a domain name to a n API group

API Groups		
APIs	Record Request Headers      Separate multiple header names with commutivation      Indicate all header names.	ias (,). You can use an aste
Backend Services	Record Response Headers      Indicate all header names.	nas (,). You can use an aste
Plug-ins	Record JWT Claims Separate multiple JWT claim names with con	mmas (,). You can use an a
VPC Access Aut	indicate all JWT claim names. If you configure User Log Settings, the following fields will be recorded in user logs based on the settings: requestBody, responseBody, requestHeaders, responseHeaders, queryString, a	and jwtClaims. Each field (
SDKs	exceed 4,096 bytes in size, and extra-long fields are truncated.	
Call API	Default Domain Name to Support Access over HTTPS (For dedicated instances only)	
Documentation	Default Domain Name *.fredhuang.com	
	<ol> <li>It multiple domain names are configured in an API group, you must specify the default domain name. It is way, when the dedicated instance receives an SSL handshake request from a Name Indication (SNI), the dedicated instance returns the certificate of the default domain name. If you do not specify the default domain name, a certificate is randomly returned.</li> <li>It multiple API groups are bound to a dedicated instance are bound to a dedicated instance are involved.</li> </ol>	a client that does not supp
	2. In multiple Aringtoups are bound to a destruction instance, only the default domain name of the lifst Aringtoup can be loaded. The settings to other Aringtoups are invalid.	

In a dedicated instance, if multiple API groups are all configured with default domain names, only the default domain name configured for the first API group can be loaded.

# 5.Import Swagger files to create APIs

Swagger is a specification used to describe API definitions. It is widely used to define and describe APIs for backend services. You can call the Create APIs By Importing Swagger operation to import Swagger 2.0 files for the creation of APIs. You can also import Swagger 2.0 files in the API Gateway console, as shown in the following figure.

ApiGateway	API List							
	API Name \$ Enter the	API name		Search	🕽 Tags	Create API	Import Swagger	Create Data Service APIs
Overview								
Instances	API Name	Tag	Visibility	Group	Description	Last Modified	Stage (All) -	Operation
Publish APIs		۲	Public	181		Oct 10,2020 15:02:54	Release (Running) Pre Test (Running)	Deploy   Debug   More -
API Groups							Balance (Burning)	
APIs		۲	Public	in imprediant.		Jun 18,2020 17:19:13	Pre	Deploy   Debug   More -
Plugin							lest (Hunning)	
VPC Access		۲	Private			Jul 16,2020 17:30:44	Release (Running) Pre Test	Deploy   Debug   More -
Log Manage Owned APIs SDK	0	۲	Private			Sep 24,2020 10:44:44	Release Pre Test	Deploy   Debug   More -
Consume APIs     Documentation		۲	Private			May 09,2020 14:34:21	Release Pre Test (Running)	Deploy   Debug   More -

API Gateway Swagger extensions are based on Swagger 2.0. You can create a Swagger definition for API entities and import Swagger files into the API Gateway console to create or update API entities. API Gateway supports Swagger 2.0 by default, which is compatible with most Swagger specifications. However, there are some differences in specification compliance for API creation. For more information, see Compatibility.

This topic describes API Gateway Swagger extensions and provides examples of their implementations.

# ? Note

All the parameters and their valid values in Swagger are case-sensitive.

# 1. Swagger extensions

Swagger extensions are used to extend the native Swagger Operation Object. They provide features such as authentication, parameter mapping, and backend services. The x-aliyun-apigateway-any-method extension is added. It is used to specify the HTTP method, by using which you can call APIs. All

extensions start with x-aligun-apigateway-, which are described as follows:

# 1.1 x-aliyun-apigateway-auth-type: authorization type

Applies to Operation Object and specifies an API authorization type.

# Valid values:

- APP: app authorization for Alibaba Cloud API Gateway. This is the default value.
- ANONYMOUS: anonymous authorization.

# Example:

```
...
paths:
    'path/':
    get:
        x-aliyun-apigateway-auth-type: ANONYMOUS
...
```

# 1.2 x-aliyun-apigateway-api-market-enable: publishing to the Alibaba Cloud Marketplace

Applies to Operation Object and specifies whether an API is published to the Alibaba Cloud Market place.

### Valid values:

- true
- false: This is the default value.

#### Example:

```
...
paths:
    'path/':
    get:
        x-aliyun-apigateway-api-market-enable: true
...
```

# 1.3 x-aliyun-apigateway-api-force-nonce-check: forcible nonce verification

Applies to Operation Object and specifies whether to perform forcible nonce verification on an API.

# Valid values:

- true
- false: This is the default value.

#### Example:

```
...
paths:
    'path/':
    get:
        x-aliyun-apigateway-api-force-nonce-check: true
...
```

# 1.4 x-aliyun-apigateway-parameter-handling: API mapping relationship

Applies to Operation Object and specifies the mappings between request parameters and backend service parameters. If you select PASSTHROUGH as the mapping relationship, the Parameter Object does

 $not \ support \ \ \text{x-aliyun-apigateway-backend-location} \quad and \ \ \text{x-aliyun-apigateway-backend-name} \ .$ 

Valid values:

- PASSTHROUGH: request parameter passthrough. This is the default value.
- MAPPING: request parameter mapping.

#### Example:

```
...
paths:
    'path/':
    get:
        x-aliyun-apigateway-parameter-handling: MAPPING
...
```

#### 1.5 x-aliyun-apigateway-backend: backend service type

Applies to Operation Object and specifies backend service information. Specific properties vary based on the backend service type. For details, see the following tables.

#### 1.5.1 Backend service type: HTTP

This type is used to configure backend service addresses. If backend service addresses can be directly accessed, this type is used.

#### Property description

Property name	Туре	Description
type	string	Required. The value is HTTP.
address	string	Required. The address of the backend service.
path	string	Optional. The path of the backend service. The value can be a variable. By default, the value of this property is the same as the root path.
method	string	Required. The backend request method.
timeout	int	Optional. The default value is 10000. Value range: [500,30000].

#### Example:

```
x-aliyun-apigateway-backend:
  type: HTTP
  address: 'http://www.aliyun.com'
  path: '/builtin/echo'
  method: get
  timeout: 10000
....
```

#### 1.5.2 Backend service type: HTTP-VPC

If a backend service is running on a VPC, this type is used. To configure the backend service, you need to Create an API operation with a resource in a VPC as the backend service and use the VPC authorization name.

#### Property description

Property name	Туре	Description
type	string	Required. The value is HTTP-VPC.
vpcAccessName	string	Required. The name of the VPC on which the backend service is running.
path	string	Optional. The path of the backend service. The value can be a variable. By default, the value of this property is the same as the root path.
method	string	Required. The backend request method.
timeout	int	Optional. The default value is 10000. Value range: [500,30000].

# Example:

...
x-aliyun-apigateway-backend:
 type: HTTP\_VPC
 vpcAccessName: vpcAccess1
 path: '/users/{userId}'
 method: GET
 timeout: 10000
....

# 1.5.3 Backend service type: FC

The backend service type is Function Compute.

# Property description

Property name	Туре	Description
type	string	Required. The value is FC.
fcRegion	string	Required. The region where Function Compute is deployed.
serviceName	string	Required. The service name of Function Compute.
functionName	string	Required. The function name of Function Compute.
arn	string	Optional. RAM authorization for Function Compute.

#### Example:

```
...
x-aliyun-apigateway-backend:
  type: FC
  fcRegion: cn-shanghai
  serviceName: fcService
  functionName: fcFunction
  arn: acs:ram::11111111:role/aliyunapigatewayaccessingfcrole
...
```

1.5.4 Backend service type: MOCK

This type is used to simulate expected return results.

#### Property description

Property name	Туре	Description
type	string	Required. The value is MOCK.
mockResult	string	Required. Return results simulated by Mock.
mockStatusCode	Integer	Optional.
mockHeaders	Header	Optional.

#### Header description

Property name	Туре	Description
name	string	Required.
value	string	Required.

#### Example:

```
...
x-aliyun-apigateway-backend:
    type: MOCK
    mockResult: mock resul sample
    mockStatusCode: 200
    mockHeaders:
        - name: server
        value: mock
        - name: proxy
        value: GW
...
```

#### 1.6 x-aliyun-apigateway-constant-parameters: constant parameters

Applies to Operation Object and defines the constant parameters that are always received by backend services. However, you do not need to include these parameters in the requests that are sent to the backend services.

# Property description

Property name	Туре	Description
backendName	string	Required. The name of the constant parameter that is always received by the backend service.
value	string	Required. The value of the constant parameter.
location	String	Required. The location in which the constant parameter is carried. Valid values: query and header.
description	string	Optional. The description of the constant parameter.

# Example:

```
...
x-aliyun-apigateway-constant-parameters:
    backendName: swaggerConstant
    value: swaggerConstant
    location: header
    description: description of swagger
...
```

# 1.7 x-aliyun-apigateway-system-parameters: system parameters of backend services

Applies to Operation Object and specifies the system parameters of backend services.

# Property description

Property name	Туре	Description
systemName	string	Required. The name of the system parameter.
backendName	string	Required. The name of the backend parameter.

Property name	Туре	Description
location	String	Required. The location in which the system parameter is carried. Valid values: query and header.

#### Example:

```
...
x-aliyun-apigateway-system-parameters:
    - systemName: CaAppId
    backendName: appId
    location: header
...
```

# 1.8 x-aliyun-apigateway-backend-location: backend parameter location

Applies to Parameter Object and specifies the location in which a parameter is carried in a request sent to a backend service. This extension takes effects only when

x-aliyun-apigateway-parameter-handling is set to MAPPING.

Valid values:

- path
- header
- query
- formData

Example:

```
parameters:
  - name: swaggerHeader
    in: header
    required: false
    type: number
    format: double
    minimum: 0.1
    maximum: 0.5
    x-aliyun-apigateway-backend-location: query
    x-aliyun-apigateway-backend-name: backendQuery
....
```

1.9 x-aliyun-apigateway-backend-name: backend parameter name

Applies to Parameter Object and specifies the name of a parameter in a request sent to a backend

service. This extension takes effects only when x-aliyun-apigateway-parameter-handling is set to MAPPING.

#### Example:

```
parameters:
  - name: swaggerHeader
    in: header
    required: false
    type: number
    format: double
    minimum: 0.1
    maximum: 0.5
    x-aliyun-apigateway-backend-location: query
    x-aliyun-apigateway-backend-name: backendQuery
...
```

#### 1.10 x-aliyun-apigateway-query-schema: model definition for query parameters

Applies to Parameter Object and defines a model for query parameters. This extension can be used when a parameter is of the STRING type and is defined as a query parameter.

#### Example:

```
...
parameters:
    - name: event_info
    in: query
    required: true
    type: string
    x-aliyun-apigateway-query-schema:
        $ref: "#/definitions/EvnetInfo"
...
```

# 1.11 x-aliyun-apigateway-any-method: ANY method

Applies to Path Item Object and specifies the HTTP method, by using which you can call APIs.

#### Example:

```
...
paths:
    'path/':
        x-aliyun-apigateway-any-method:
        ...
...
```

1.12 x-aliyun-apigateway-app-code-type: AppCode-based simple authentication

Applies to Operation Object and specifies whether an API supports AppCode-based simple authentication.

Valid values:

- DEFAULT: AppCode-based simple authentication is supported by default.
- DISABLE: AppCode-based simple authentication is disabled.
- HEADER: AppCode is carried in the header of a request.
- HEADER\_QUERY: AppCode is carried in the header of a request or in a query parameter.

Example:

```
...
paths:
    'path/':
    get:
        x-aliyun-apigateway-app-code-type: HEADER
...
```

# 2. Compatibility

API Gateway and Swagger specifications define APIs in different ways.

2.1 Comparison of the parameter types in Swagger and API Gateway

Parameter type in Swagger	Parameter type in API Gateway	Supported verification parameter and rule
<ul><li>type:integer</li><li>format:int32</li></ul>	Int	<ul><li>mininum</li><li>maxnum</li></ul>
<ul><li>type:integer</li><li>format:int64</li></ul>	Long	<ul><li>mininum</li><li>maxnum</li></ul>
<ul><li>type:number</li><li>format:float</li></ul>	Float	<ul><li>mininum</li><li>maxnum</li></ul>
<ul><li>type:number</li><li>format:double</li></ul>	Doulbe	<ul><li>mininum</li><li>maxnum</li></ul>
type:string	String	<ul><li>maxLength</li><li>enumValues</li><li>pattern</li></ul>

Parameter type in Swagger	Parameter type in API Gateway	Supported verification parameter and rule
<ul><li>type:boolean</li><li>format:Boolean</li></ul>	Boolean	-

# 2.2 Support for the consumes field

If a Swagger configuration file contains FormData parameters, the consumes field must be configured. In API Gateway, this field can only be set to application/x-www-form-urlencoded .

```
consumes:
    - application/x-www-form-urlencoded
```

# 3. Swagger extension examples

This section provides four examples of Swagger extensions for API Gateway. The examples cover practically all aspects of the Swagger extensions. You can refer to these examples when you define API entities based on the Swagger extensions.

? Note

The examples are for reference only.

#### 3.1 Example with HTTP as the backend service type

```
swagger: '2.0'
basePath: /
info:
  version: '0.9'
 title: Aliyun Api Gateway Swagger Sample
schemes:
 - http
  - https
x-aliyun-apigateway-paramater-handling: MAPPING
x-aliyun-apigateway-api-market-enable: true
x-aliyun-apigateway-api-force-nonce-check: true
x-aliyun-apigateway-backend:
 type: HTTP
 address: 'http://www.aliyun.com'
 method: get
 timeout: 10000
paths:
  '/http/get/mapping/{userId}':
   get:
      operationId: case1
      schemes:
        - http
```

```
- https
   x-aliyun-apigateway-parameter-handling: MAPPING
   x-aliyun-apigateway-api-market-enable: true
   x-aliyun-apigateway-auth-type: ANONYMOUS
   parameters:
     - name: userId
       in: path
       required: true
       type: string
     - name: swaggerQuery
       in: query
       required: false
       default: '123465'
       type: integer
       format: int32
       minimum: 0
       maximum: 100
     - name: swaggerHeader
       in: header
       required: false
       type: number
       format: double
       minimum: 0.1
       maximum: 0.5
       x-aliyun-apigateway-backend-location: query
       x-aliyun-apigateway-backend-name: backendQuery
   x-aliyun-apigateway-constant-parameters:
     - backendName: swaggerConstant
       value: swaggerConstant
       location: header
       description: description of swagger
   x-aliyun-apigateway-system-parameters:
     - systemName: CaAppId
       backendName: appId
       location: header
   responses:
      '200':
       description: 200 description
      '400':
       description: 400 description
'/echo/test/post/{userId}':
 post:
   operationId: testpost
   schemes:
     - http
     - https
   x-aliyun-apigateway-parameter-handling: MAPPING
   x-aliyun-apigateway-backend:
     type: HTTP
     address: 'http://www.aliyun.com'
     method: post
     timeout: 10000
   consumes:
     - application/x-www-form-urlencoded
```

```
parameters:
   - name: userId
      required: true
     in: path
     type: string
   - name: swaggerQuery1
     in: query
      required: false
     default: '123465'
     type: integer
      format: int32
     minimum: 0
     maximum: 100
     x-aliyun-apigateway-enum: 1,2,3
   - name: swaggerQuery2
     in: query
      required: false
      type: string
      x-aliyun-apigateway-backend-location: header
     x-aliyun-apigateway-backend-name: backendHeader
      x-aliyun-apigateway-query-schema:
       $ref: '#/definitions/AiGeneratePicQueryVO'
   - name: swaggerHeader
     in: header
     required: false
     type: number
      format: double
     minimum: 0.1
     maximum: 0.5
     x-aliyun-apigateway-backend-location: query
      x-aliyun-apigateway-backend-name: backendQuery
   - name: swaggerFormdata
     in: formData
      required: true
      type: string
 responses:
    '200':
     description: 200 description
      schema:
       $ref: '#/definitions/ResultOfGeneratePicturesVO'
    '400':
      description: 400 description
x-aliyun-apigateway-any-method:
 operationId: case2
 schemes:
   - http
   - https
 x-aliyun-apigateway-parameter-handling: MAPPING
 x-aliyun-apigateway-backend:
   type: HTTP
   address: 'http://www.aliyun.com'
   path: '/builtin/echo/{abc}'
   method: post
   timeout: 10000
  narameters.
```

parameters. - name: userId in: path required: false default: '123465' type: integer format: int32 minimum: 0 maximum: 100 x-aliyun-apigateway-backend-name: abc x-aliyun-apigateway-backend-location: path responses: '200': description: 200 description '400': description: 400 description definitions: AiGeneratePicQueryVO: type: object properties: transactionId: type: string description: asynchronous task ID GeneratePictureVO: type: object properties: id: type: integer format: int64 description: image ID name: type: string description: image name GeneratePicturesVO: type: object properties: failSize: type: integer format: int64 description: number of failures list: type: array description: image list items: \$ref: '#/definitions/GeneratePictureVO' title: GeneratePictureVO successSize: type: integer format: int32 description: number of successes totalSize: type: number format: float description: total number of requests

ResultOfGeneratePicturesVO:
```
type: object
properties:
model:
    description: returned content
    $ref: '#/definitions/GeneratePicturesVO'
    title: GeneratePicturesVO
    requestId:
    type: string
    description: request ID
```

3.2 Example with HTTP-VPC as the backend service type

```
swagger: '2.0'
basePath: /
info:
 version: '0.9'
 title: Aliyun Api Gateway Swagger Sample
schemes:
 - http
 - https
paths:
  '/http/get/mapping/{userId}':
   get:
     operationId: case1
     schemes:
       - http
        - https
     x-aliyun-apigateway-parameter-handling: MAPPING
     x-aliyun-apigateway-backend:
        type: HTTP-VPC
       vpcAccessName: vpcName1
       path: '/builtin/echo/{userId}'
       method: get
       timeout: 10000
      parameters:
        - name: userId
         in: path
         required: true
         type: string
        - name: swaggerQuery
         in: query
          required: false
         default: '123465'
         type: integer
         format: int32
         minimum: 0
         maximum: 100
        - name: swaggerHeader
         in: header
          required: false
          type: number
          format: double
```

# Create API-Import Swagger files to create APIs

```
minimum: 0.1
       maximum: 0.5
       x-aliyun-apigateway-backend-location: query
       x-aliyun-apigateway-backend-name: backendQuery
   responses:
      '200':
       description: 200 description
      '400':
       description: 400 description
'/echo/test/post':
 post:
   operationId: testpost
   schemes:
     - http
     - https
   x-aliyun-apigateway-parameter-handling: MAPPING
   x-aliyun-apigateway-backend:
     type: HTTP-VPC
     vpcAccessName: vpcName2
     path: '/builtin/echo'
     method: post
     timeout: 10000
   consumes:
     - application/x-www-form-urlencoded
   parameters:
     - name: swaggerQuery1
       in: query
       required: false
       default: '123465'
       type: integer
       format: int32
       minimum: 0
       maximum: 100
     - name: swaggerQuery2
       in: query
       required: false
       type: string
       x-aliyun-apigateway-backend-location: header
       x-aliyun-apigateway-backend-name: backendHeader
     - name: swaggerHeader
       in: header
       required: false
       type: number
       format: double
       minimum: 0.1
       maximum: 0.5
       x-aliyun-apigateway-backend-location: query
       x-aliyun-apigateway-backend-name: backendQuery
     - name: swaggerFormdata
       in: formData
       required: true
       type: string
   responses:
      '200':
```

```
aescription: 200 aescription
    '400':
      description: 400 description
x-aliyun-apigateway-any-method:
 operationId: case2
 schemes:
   - http
    - https
 x-aliyun-apigateway-parameter-handling: PASSTHROUGH
  x-aliyun-apigateway-backend:
   type: HTTP-VPC
   vpcAccessName: vpcName3
  path: '/builtin/echo'
   method: post
   timeout: 10000
  responses:
   '200':
     description: 200 description
    '400':
     description: 400 description
```

3.3 Example with Function Compute as the backend service type

```
swagger: '2.0'
basePath: /
info:
 version: '0.9'
  title: Aliyun Api Gateway Swagger Sample
schemes:
 - http
 - https
paths:
  '/http/get/mapping/{userId}':
   get:
     operationId: case1
     schemes:
       - http
       - https
     x-aliyun-apigateway-parameter-handling: MAPPING
     x-aliyun-apigateway-backend:
       type: FC
       fcRegion: cn-shanghai
       serviceName: fcService
        functionName: fcFunction
       arn: acs:ram::111111111:role/aliyunapigatewayaccessingfcrole
      parameters:
       - name: userId
         in: path
         required: true
         type: string
      responses:
        '200':
          description: 200 description
        '400':
          description: 400 description
```

#### 3.4 Example with Mock as the backend service type

```
swagger: '2.0'
basePath: /
info:
 version: '0.9'
 title: Aliyun Api Gateway Swagger Sample
schemes:
 - http
paths:
 '/mock/get/mapping/{userId}':
   get:
     operationId: case1
     schemes:
       - http
       - https
     x-aliyun-apigateway-parameter-handling: MAPPING
     x-aliyun-apigateway-backend:
       type: MOCK
       mockResult: mock resul sample
       mockStatusCode: 200
       mockHeaders:
         - name: server
           value: mock
         - name: proxy
           value: GW
     parameters:
       - name: userId
         in: path
         required: true
         type: string
     responses:
        '200':
         description: 200 description
        '400':
         description: 400 description
```

# 4. Usage notes

Pay attention to the following instructions, which may affect Swagger importing.

#### 4.1 Support for global scope

- x-aliyun-apigateway-backend
- x-aliyun-apigateway-api-market-enable
- x-aliyun-apigateway-api-force-nonce-check
- x-aliyun-apigateway-parameter-handling
- x-aliyun-apigateway-auth-type

#### 4.2 Description of the Definition field in Swagger

Swagger importing supports model definition, which differs from that in the original Swagger specifications. Model definition is used to generate SDKs. The following limits are added to the original Swagger specifications:

- The schema tag only supports the \$ref type.
- The model in the Definition field only supports model definition of the object type.
- An array is defined in the model of the Definition field. If \$ref is used for a reference, the title tag is required. If an array type is specified, an array list is also generated during the generation of SDKs.

# 6.Use a resource in a VPC as the backend service of an API operation

This topic describes how to create a high-availability backend service. Elastic Compute Service (ECS) instances and Server Load Balancer (SLB) instances in a Virtual Private Cloud (VPC) can be used as backend services of API operations.

# Overview

Alibaba Cloud VPC allows you to build an isolated network and customize IP address ranges, CIDR blocks, route tables, and gateways for the network. API Gateway allows you to create API operations for resources that are deployed in VPCs. To use a resource in a VPC as the backend service of an API operation, you must first authorize API Gateway to access the resource.

# 1. Authorize API Gateway to access resources in a VPC

To create an API operation for a resource that is deployed in a VPC, you must first authorize API Gateway to access the resource. To authorize API Gateway to a resource, you must specify the resource and an access port, such as port 443 of an SLB instance or port 80 of an ECS instance.

- After authorization, API Gateway can access the resource in the VPC over the specified port.
- The authorization only allows API Gateway to access the resources in the VPC to call API operations.
- API Gateway can access only resources that it has permissions to access over an authorized port. For example, if you authorize API Gateway to use only port 80 of an SLB instance in a VPC, API Gateway can access this SLB instance only over port 80.

ECS instances and SLB instances in VPCs can be used as backend services of API operations.

- ECS instance: When you create VPC access authorization, enter the ID or private IP address of the ECS instance in the Instance Id Or IP field in the Create VPC Access dialog box.
- SLB instance: Only internal SLB instances are supported. When you create VPC access authorization, enter the ID or private IP address of the SLB instance in the Instance Id Or IP field in the Create VPC Access dialog box.

# 2. Build an HA architecture

To build an HA architecture, we recommend that you use an internal SLB instance as the backend service of an API operation. The SLB instance can be used to distribute access traffic to multiple ECS instances based on the forwarding policy. This improves the overall system performance and availability of applications.



#### 2.1 Create instances in a VPC

Purchase and create SLB and ECS instances in a VPC. In this example, the SLB instance listens to port 80 of ECS instances, and the ECS instances are deployed in NGINX.

The following figure shows the details of the internal SLB instance.

E C-J Alibaba (	Cloud			Q Search		Expenses	Tickets ICP	Enterprise S	upport C	fficial Site 💽	<b>۵.</b>	₩ @	EN O
Server Load Balancer		Server Load Balancer /	Instances / auto	o_apitest_slb/172.17.2	08.66								
Overview		← auto_api	test_slb	/2	.U.UU		Start   Stop	🗣 Edit T	ags 🖌	Change Configur	ations	⊗ Health	Check
Instances	^	Instance Details	Listener	VServer Groups	Default Server Group	Primary	/Secondary Serve	r Groups N	Ionitoring				
Instances													
Expired Instances		<b>Basic Information</b>											
Certificates		Name	auto_apitest_	slb Edit			ID	Ib-2200	ບຊຸມບຸບ ເຊິ່ງ	ур. у Сору			
oortinootoo		Status	✓ Running				Network Type	VPC					
Access Control		Address Type	Private Netwo	ork			Zone	Beijing	Zone E(Prim	ary) / Beijing Zone	F(Secondar	y)	
Logs	^	< Deletion Protection	盘 Disabled	EnableDeletion Prot	ection		Configuration Read Mode	d-only 🔒 Disa	bled Enabl	eConfiguration Rea	ad-only Moo	ie	
Operation Logs													
Access Logs		Billing Information											
Health Check Logs		Bandwidth Billing Method	Pay-As-You-0	Go			Billing Method	Pay by	Traffic Billin	g Details			
SLB Lab	~	Instance Type	Guaranteed-I	Performance slb.s1.s	mall 🞯		IP Address	1. 2. 17.	Priva) نځ.ځې	e Network) 53.10	u.uh.iv.₄.EIP	) Unbind E	P API
		Creation Time	Aug 10, 2020	), 17:32:11			Associated Networ	rk vpu	ho	92tvs) ., vsw-22.		hve ground	
Idle SLB Instances		Bandwidth	5120 Mbps										
Quota Management													ПÔ
Pasauroas													<u>B</u> Ö

#### 2.2 Authorize API Gateway to access the VPC

Log on to the API Gateway console. In the left-side navigation pane, choose Publish APIs > VPC Access. On the VPC Access List page, click Create VPC Access. In the Create VPC Access dialog box, configure the required parameters.

VPC Access Name: the name of the current authorization entry. You need to select this name when you configure an API operation. To facilitate subsequent management, make sure that the name is unique in API Gateway.

Region:	China North 2 (Beijing)
*VPC Access Name:	
	It may contain Chinese characters, English letters, numbers, and English- style underlines. It must start with a letter or Chinese character and be 4-50 characters long
*VPC Id:	
*Instance Id Or IP:	
	Please enter the instance ID of your ECS or SLB (for example: i- uf1dfwexxxxx or Ib-jiwb2342xxxxxx), or the private network IP of the corresponding instance
*Instance Port:	
	It must be numbers and 2-6 characters long, for example: 80

#### 2.3 Create an API operation

The procedure for creating an API operation with a service in a VPC as the backend service is the same as that for creating an API operation with HTTP or Function Compute as the backend service. For more information, see Create an API operation.

For more information about application creation and authorization, see Create an API operation with a service in a VPC as the backend service.

#### 2.4 Test the API operation

You can test the API operation by using one of the following methods:

- Debug the API operation
- Download the SDK
- Call an API operation by using an AppCode

#### 2.5 Security

API Gateway calls backend services in a VPC over an internal network. If you require higher security or your internal SLB instance has a blacklist and whitelist, you must add outbound IP addresses of API Gateway to the whitelist. For more information about SLB blacklist and whitelist settings, see Enable access control.

If you have configured a security group for ECS instances, you must add a security group rule to allow the outbound IP addresses of API Gateway. For information about how to add security group rules for ECS instances, see Add security group rules.

For information about how to obtain the outbound IP addresses of API Gateway, see Create an API operation with a resource in a VPC as the backend service.

# FAQ

#### 1. Does API Gateway support public SLB instances?

No, API Gateway supports only internal SLB instances when it calls SLB instances over an internal network. If API Gateway needs to call SLB instances over the Internet, you can create an API operation with HTTP or HTTPS as the backend service.

#### 2. Can I authorize API Gateway to access multiple VPCs?

Yes. If you need to use multiple resources that are deployed in multiple VPCs as backend services, you can create multiple authorization entries in the API Gateway console to authorize API Gateway to access these VPCs.

#### 3. Why am I unable to authorize API Gateway to access a VPC?

If you are unable to authorize API Gateway to access a VPC, check whether the ID of the VPC, the ID of the instance on which the backend service is deployed, and the port number that you entered are correct. Make sure that the authorization entry is created in the region where the VPC resides.

#### 4. Is the security of my VPC affected after I authorize API Gateway to access my VPC?

No, the security of your VPC is not affected.

- API Gateway can call resources in your VPC only after you authorize it to access your VPC.
- Only the API operation that you authorized can call the resources in your VPC.
- You can configure access control policies for ECS and SLB instances that are used as backend services.

#### 5. Does API Gateway support VPCs in different regions?

Yes. You can use Cloud Enterprise Network (CEN) to allow API Gateway to access VPCs in different regions.

# 7.Use Function Compute as the backend service of an API operation

Function Compute is an event-driven compute service. Functions are event-driven, that is, when an event occurs, the event triggers the execution of the corresponding function. API Gateway is an event source for Function Compute. After API Gateway receives a request for an API operation that uses Function Compute as the backend service, API Gateway triggers the execution of the corresponding function in Function Compute and Function Compute sends the execution result to API Gateway.

Based on the integration with Function Compute, API Gateway allows you to provide API operations in a secure manner and supports features such as authentication, throttling, and data conversion. For more information, see Features.

# How it works

After API Gateway receives a request for an API operation with Function Compute as the backend service, API Gateway converts the request parameters to key-value pairs in the map format and sends the request to Function Compute. Function Compute handles the request and returns a response in a specific output format to API Gateway. The response includes information such as the status code, header, and body, as shown in the following figure. API Gateway maps the information in the response that is returned by Function Compute to the status code, header, and body of an API response, and returns the API response to the client.



# Input format of request parameters in each API request that API Gateway sends to Function Compute

After API Gateway receives a request for an API operation with Function Compute as the backend service, API Gateway converts request parameters in the request to key-value pairs in the map format. Then, API Gateway passes the key-value pairs to Function Compute as the value of the event parameter to trigger the execution of the corresponding function. Function Compute obtains required parameters from the key-values pairs that are sent from API Gateway, as shown in the following code snippet:

```
{
   "path":"api request path",
   "httpMethod":"request method name",
   "headers":{all headers,including system headers},
   "queryParameters":{query parameters},
   "pathParameters":{path parameters},
   "body":"string of request payload",
   "isBase64Encoded":"true|false, indicate if the body is Base64-encode"
}
```

- If the value of "isBase64Encoded" is "true", the body content that API Gateway sent to Function Compute is Base64-encoded. To trigger the corresponding function, Function Compute needs to perform Base64 decoding on the body content first.
- If the value of "isBase64Encoded" is "false", API Gateway did not perform Base64 encoding on the body content that was sent to Function Compute.

#### Output format of the response that Function Compute sends to API Gateway

Function Compute sends the execution result of the corresponding function to API Gateway in the JSON format, as shown in the following code snippet. API Gateway will parse the response that Function Compute returns.

```
{
  "isBase64Encoded":true|false,
  "statusCode":httpStatusCode,
  "headers":{response headers},
  "body":"..."
}
```

- If the body content of the response is binary data, Function Compute needs to perform Base64 encoding on the body content. The value of "isBase64Encoded" is "true" in the response that is sent to API Gateway. If Function Compute does not need to perform Base64 encoding on the body content of the response, the value of "isBase64Encoded" is "false". If the value of "isBase64Encoded" is "false". If the value of "isBase64Encoded" is "true" in the response that is sent from Function Compute, API Gateway performs Base64 decoding on the body content of the response before API Gateway sends the mapped API response to the client.
- In the Node.js environment, Function Compute constructs a callback parameter based on different execution results:
  - If the result to be sent to API Gateway is a success response, the value of the callback parameter is in the following format: callback{null,{"statusCode":200,"body":"..."}}.
  - If the result to be sent to API Gateway is an exception, the value of the callback parameter is in the following format: callback{new Error('internal server error'),null}.
  - If the result to be sent to API Gateway is an error response that is caused by the client's request, the value of the callback parameter is in the following format: callback{null, {"statusCode":400,"body":"param error"}}.
- If Function Compute returns a response that is not in the required format, API Gateway returns an error message "503 Service Unavailable" to the client.

### Procedure

To create an API operation with Function Compute as the backend service in API Gateway, perform the following steps:

- 1. Create a function in the Function Compute console.
- 2. Create an API operation with Function Compute as the backend service in the API Gateway console.
- 3. Debug the API operation.
- 4. Publish the API operation to the production environment.

### Create a function in the Function Compute console

 Create a service. Log on to the Function Compute console. Select a region where you want to create the service and function in the top navigation bar. Click Service/Function in the left-side navigation pane. On the Service/Function page, click the drop-down arrow next to Create Function and click Create Service. On the page that appears, set relevant parameters. Note that after a service is created, you cannot change the region where the service resides. Proceed with caution when you select a region.

😑 🕞 Alibaba Cloud	China (Hangzh Q Search	Expenses T	ickets ICP Enterprise Su	ipport Official Site 도 쇼 '귿 ⑦ EN
Function Compute	Function Compute / Service/Function			Onboarding Guide Product Updates Help
Overview	Create Function A Service Na	ame 🗸 Enter a service name	Q	Display as Blocks (Single Row) $m{ u}$
Application Center	Create Service	the state of the s	to share the statement	
Service/Function	Tag:	Tag: 🔖	Tag: 💊	
Custom Domain				
Resource Center	Functions Service Configuration	ons Versions Service Metrics	Provisioned Resources	On-Demand Resources

2. Create a function in the service you created. After you create the service, the service appears on the Service/Function page. Select the service and click **Create Function**. The Create Function wizard appears.

i. In the Create Function step, click Template Function in the Function Type section and select a function template.

The Function Compute console provides the api-gateway-nodejs6 template for you to create a function as the backend service of an API operation in Node.js 6.

If the api-gateway-nodejs6 template is not applicable to your business scenario, click **Event Function** in the Function Type section. If you create an event function, you must upload your own code in the runtime environment of the function. We recommend that you prepare your code in advance.

Function Compute / Service/Function / Create Function ← Create Function				Onbearding Guid Product Updates
1 Create Function			2 Configure Function	
Function Type:				
Event Function	HTTP Function		Template Funtion	
Use the helloworld template to create a blank function.	Use the helloworld templa	te to create a blank HTTP function.	Use a sample code template to create a function.	
Search by name, description, or environment	Q			_
alimebot-nodejs		aliyun-config-python3		api-gateway-nodejs6 Template Details
Runtime nodejs6		Runtime python3		Runtime nodejs6
Description This template is a function template that is po DialogStudio. The template provides a basic developers need to develop code according t	ovided to developers of function structure, and o this structure.	Description Custom rule templa customized compli	ate of Cloud Config allows you to quickly develop ance rules.	Description This template implements a backend service for API Gateway. It shows how to return different content formats, such as HTML pages, JSON documents, and images.
copy-oss-object-python27		data-lake-analytics		flask-web
Runtime python2.7		Runtime python3		Runtime python2.7
Description This template shows how to backup folders f bucket to other destinations. This sample co- to Qiniu cloud.	rom a specified OSS de shows how to backup	Description This is a typical ten data stored in Obje various dimensions	nplate describing how to build a data lake for the ct Storage Service (OSS) and analyze the data in by the integration of Function Compute to Dat	Description Through this template demo, the user can create a serverless flask web project, and invoke the function via URL.
Next				Items per Page 6 9 12 < Previous 1 2 3 4 Next

ii. For information about how to write code for a function, see Programming Languages.

Function Compute	Function Compute / Service/Function / apitest $\checkmark$ / eventfuncTest $\checkmark$	Onboarding Guide Product Updates Help				
Overview	← eventfuncTest service					
Application Center Service/Function	Overview Code Triggers Log Function Metrics Asynchronous Invocation Configuration	ARN - acs:fc:cn-hangzhou:1227466664334133:services/apitest.LATEST/functions/eventfu				
Custom Domain	Code Management					
Resource Center	Invoke Event					
Recently Viewed Functions	In-line Edit Import from OSS Upload Zip File Upload Folder	Use the legacy editor				
apitest/eventfuncTest	∮∲ File Edit Selection View Go Help	▶ Invoke 🕸 Event 🕂 🖍				
apites(/eventText	<pre>DPUCHER " " " notec, p x</pre>	Eggen Xan				
	<b>○</b> ○▲○	Ln 1, Col 1 LF UTF-8 Spaces: 4 JavaScript 🌲				
	Result	P				
		B				

# Create and configure an API operation with Function Compute as the backend service

After you create the function in the Function Compute console, you must create an API operation with Function Compute as the backend service in the API Gateway console.

- 1. Log on to the API Gateway console.
- 2. In the left-side navigation pane, choose Publish APIs > API Groups. Select a region in the top

#### navigation bar and click Create Group. If you have created an API group, skip this step.

(?) Note If the region where you created the API group is not the same region where you created the service in Function Compute, API Gateway will access the service in Function Compute by using the Internet. If you need high data security and low network latency, select the region where you created the service in Function Compute when you create the API group in API Gateway.

Create Group	×
Region:	China North 2 (Beijing) (Each user can create up to 50 groups)
*Instance:	<pre></pre>
*Group Name:	testFunctionComputeGroup
Tag optimus	Group name must be globally unique and may contain Chinese characters, English letters, numbers, and English-style underlines. It must start with a letter or Chinese character and be 4-50 characters long
Tag settings	Cannot exceed 180 obstactors
Description.	
	OK Cancel

After the API Group is created, you can find it on the Group List page and click View Stages in the Operation column. On the Stage Management page, you can configure environment variables for the API group. API Gateway supports three environments: Test, Pre, and Release. To prevent environment switch from changing the backend service address of an API operation, you can configure an environment variable to implement automatic routing of API requests. For more information about how to configure environment variables, see Configure different environments for an API operation.

Group Details t Back to group list								Refresh
Basic Information						Turn on cloue	d monitoring Api Lis	t Modify Group Message
Region: China North 2 (Beijing)		Group Name: and	20 · (	Gro	ip ID: - permenentari - E ci chene contenuenti - e	<u>1.17</u>		
API List								
API Name 💠 Enter the API name			Search 📎 Tags			Create API	Import Swagger	Create Data Service APIs
TFilters: Group Name:testFunctionGroup & Clear								
API Name	Тад	Visibility	Group		Description	Last Modified	Stage (All) +	Operation
testFunctionApi	۲	Private	testFunctionGroup			Nov 30,2020 21:12:38	Release (Running) Pre Test	Deploy   Debug   More *
Export Swagger Authorize Deploy	Undeploy	Delete				Total of 1 entries	, 10 displayed per pag	• 10 ¢ · 1 · »

3. Create and configure an API operation.

- i. On the Group List page, find the API group you created and click View APIs in the Operation column. The API List page of the API group appears.
- ii. Click Create API. The Create API wizard appears.
- iii. In the Basic Information step, configure basic information for the API operation and click Next.

Name And Description			
Group	testFunctionGroup	٣	Create Group
API Name	testFunctionApi		0
Security Certification	Alibaba Cloud APP	\$	
AppCode Certification	Allow AppCode authentication (Header & Query)	¢	AppCode certification usage and risk tips
Signature Method	HmacSHA256	\$	
API Options	Prevent replay attacks (the request header must contain the	X-Ca	-Nonce parameter)
	Prohibit public internet access Application for VPC Intranet	Doma	in Name
	Allow cloud market		
Description	It cannot exceed 2000 characters		

iv. In the Define API Request step, configure request information for the API operation and click Next.

**?** Note If you set Request Mode to Request Parameter Passthrough, API Gateway does not handle request parameters in each API request and directly forwards the request to Function Compute.

Basic Request Definition	
Request Type	
Protocol	
Custom Domain Name	Bind domain name to the group
Subdomain Name	El loudapi.com
Request Path	/data 🗌 Match All Child Paths
	The request path must contain the Parameter Path in the request parameter within brackets ([]). For example: /getUserInfo/[userId]
HTTP Method	GET \$
Request Mode	Request Parameter Passthrough

# v. In the Define API Backend Service step, configure a backend service for the API operation and click Next.

Note In this step, set Backend Service Type to FunctionCompute. In the Service Name field, enter the name of the service that you created in the Function Compute console. In the Function Name field, enter the name of the function that you created in the Function Compute console. Click Get Authorization next to the Role Arn field. If this is the first time you create an API operation with Function Compute as the backend service, the Auto Authorize message appears after you click Get Authorization. Click Authorize in the message. The Cloud Resource Access Authorization page appears. Click Confirm Authorization Policy. Go back to the Define API Backend Service step in the API Gateway console. Click Get Authorization again. A value is automatically entered in the Role Arn field.

Basic Backend Definition					
Backend Service Type	HTTP(s) Service VPC O FunctionCompute Mock				
	If no Function, you should create Function first on FunctionCompute consc For more detail, please see The Function Compute is used as the backend	le. service of API Gateway.			
	For more detail, please see The Function Compute is used as the backend	service of API Gateway.			
Function Type	• Event Function O HTTP Function				
Region	China North 2 (Beijing)	Function Compute Console			
	Function compute communicates with API Gateway through intranet while	they are in same region.			
Service Name	testFunctionServer				
Function Name	testFunctionCompute				
Function Alias	Default Function Alias (LATEST)				
Backend Timeout	10000 ms				

# vi. In the Define Response step, configure response information for the API operation and click Create.

**Note** You must enter a sample response in the Sample of Returned Results field. For information about the response format, see Output format of the response that Function Compute sends to API Gateway.

Notice		×
	API has created successfully!	
	Operate successfully	
		Deploy OK

For more information about how to create an API operation, see Create an API operation.

### Debug the API operation

After the configurations of the API operation are completed, you are navigated to the \*\*API List\*\* page. To debug the API operation, perform the following steps:

1. Find the API operation you created and click its name or click **Manage** in the Operation column. The **API Definition** page appears.

2. In the left-side navigation pane, click Debug API.

3. On the page that appears, specify request parameters in the Request Parameters section and click **Send Request**.

A result appears on the right of the page.

If the result is a success response, the API operation can be called.

If the result includes an error code that starts with the digit 4 or 5, an error occurred in the process of calling the API operation. For more information, see How to obtain the error message and Error code table.



4. Publish the API operation to the staging environment for testing.

After the debugging proves the API operation available, go back to the **API Definition** page. Click Deploy in the upper-right corner. In the Deploy API dialog box, select **Pre** to publish the API operation to the staging environment. Then, simulate a user call. Call the API operation by using the default second-level domain name that is automatically bound to the API group to which the API operation belongs.

**Note:** If you have configured an environment variable for the API operation, you must add the keyvalue pair X-Ca-Stage: PRE in the request header when you call the API operation in the staging environment.

# Publish the API operation to the production environment

After the API operation is debugged and proved to be ready for use, you can publish the API operation to the production environment.

1. On the API List page, find the API operation you created and click its name or click **Manage** in the Operation column. The **API Definition** page appears.

2. Click Deploy in the upper-right corner. The Deploy API dialog box appears.

3. Set the Select The Stage To Release To parameter to **Release**, enter your remarks, and then click **Deploy**.

After you publish the API operation to the production environment, other users can call the API operation.

Deploy API	
You will deploy the following API(s):	
testFunctionApi	
Select The Stage To Release To: Release Pre Test	
and release after confirming the scope of modification. Contrast difference	
*Enter Change Remarks:	
Required. It cannot exceed 180 characters	
Description:	
If you have used <b>Mock Enable</b> , the API will not invoke your backend service, please confirm it with caution.	
This action will overwrite API of Release, please confirm it with caution.	
Deploy Cancel	

For more information about how to publish an API operation, see Create an API operation with HTTP as the backend service.

# Sample code

The following code snippets are samples of a function, an API request, and an API response, respectively.

# Sample code of a function

The following code snippet is the sample code for configuring a function in Function Compute:

```
module.exports.handler = function(event, context, callback) {
   var responseCode = 200;
   console.log("request: " + JSON.stringify(event.toString()));
   // Convert an event to a JSON object.
   event=JSON.parse(event.toString());
   var isBase64Encoded=false;
// Construct a response body based on the status code that is specified in a request. Diff
erent response bodies are constructed based on different status codes.
    if (event.queryParameters ! == null && event.queryParameters ! == undefined) {
       if (event.queryParameters.httpStatus ! == undefined && event.queryParameters.httpSt
atus ! == null && event.gueryParameters.httpStatus ! == "") {
            console.log("Received http status: " + event.queryParameters.httpStatus);
            responseCode = event.queryParameters.httpStatus;
        }
    }
    // If the body content of the event parameter is Base64-encoded, perform Base64 decodin
q on the body content.
   if(event.body! ==null&&event.body! ==undefined) {
    if(event.isBase64Encoded! ==null&&event.isBase64Encoded! ==undefined&&event.isBase64En
coded) {
     event.body=new Buffer(event.body, 'base64').toString();
    }
   }
   // The value of the event parameter is the content that API Gateway sends to Function C
ompute.
   var responseBody = {
      message: "Hello World!",
       input: event
    };
// Perform Base64 encoding on the body content of the response if needed.
   var base64EncodeStr=new Buffer(JSON.stringify(responseBody)).toString('base64');
// Return the response in the following format to API Gateway. The value of the isBase64En
coded parameter depends on whether the body content of the response is Base64-encoded.
   var response = {
 isBase64Encoded:true,
 statusCode: responseCode,
 headers: {
 "x-custom-header" : "header value"
 },
 body: base64EncodeStr
   };
   console.log("response: " + JSON.stringify(response));
   callback(null, response);
};
```

# Sample code of an API request

In this sample, the POST method is used to call an API operation whose request path is the following string:

/fc/test/invoke/[type]

```
POST http://test.alicloudapi.com/fc/test/invoke/test?param1=aaa&param2=bbb
"X-Ca-Signature-Headers":"X-Ca-Timestamp,X-Ca-Version,X-Ca-Key,X-Ca-Stage",
"X-Ca-Signature":"TnoBldxxRHrFferGlzzkGcQsaezK+ZzySloKqCOsv2U=",
"X-Ca-Stage":"RELEASE",
"X-Ca-Timestamp":"1496652763510",
"Content-Type":"application/x-www-form-urlencoded; charset=utf-8",
"X-Ca-Version":"1",
"User-Agent":"Apache-HttpClient\/4.1.2 (java 1.6)",
"Host":"test.alicloudapi.com",
"X-Ca-Key":"testKey",
"Date":"Mon, 05 Jun 2017 08:52:43 GMT","Accept":"application/json",
"headerParam":"testHeader"
{"bodyParam":"testBody"}
```

# Sample code of an API response

```
200
Date: Mon, 05 Jun 2017 08:52:43 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 429
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, HEAD, OPTIONS , PATCH
Access-Control-Allow-Headers: X-Requested-With, X-Sequence, X-Ca-Key, X-Ca-Secret, X-Ca-Versio
n,X-Ca-Timestamp,X-Ca-Nonce,X-Ca-API-Key,X-Ca-Stage,X-Ca-Client-DeviceId,X-Ca-Client-AppId,
X-Ca-Signature, X-Ca-Signature-Headers, X-Forwarded-For, X-Ca-Date, X-Ca-Request-Mode, Authoriza
tion, Content-Type, Accept, Accept-Ranges, Cache-Control, Range, Content-MD5
Access-Control-Max-Age: 172800
X-Ca-Request-Id: 16E9D4B5-3A1C-445A-BEF1-4AD8E31434EC
x-custom-header: header value
{"message":"Hello World!","input":{\"body":"{\"bodyParam\":\"testBody\"}","headers":{"X-Ca-A
pi-Gateway":"16E9D4B5-3A1C-445A-BEF1-4AD8E31434EC","headerParam":"testHeader","X-Forwarded-
For":"100.81.146.152", "Content-Type":"application/x-www-form-urlencoded; charset=UTF-8"},"h
ttpMethod":"POST","isBase64Encoded":false,"path":"/fc/test/invoke/test","pathParameters":{"
type":"test"},"queryParameters":{"param1":"aaa","param2":"bbb"}}}
```

# FAQ

# Why am I unable to use a function that I created in Function Compute as the backend service of an API operation in API Gateway?

When you create an API operation with Function Compute as the backend service, make sure that the service name and function name you entered are the same as the names of the service and function that you created in the Function Compute console.

# Can I use multiple functions as the backend service of an API operation?

No, you cannot use multiple functions as the backend service of an API operation. For each API operation with Function Compute as the backend service, you can configure only one function as the backend service of the API operation.

# 8.Use Mock as the backend service of an API operation

A project is typically developed by multiple partners who work together toward a specific goal. The interdependence among various stakeholders often restricts individual members during the process, and misunderstandings may affect the development process or even delay the project schedule. Mock can be used early in the project development cycle to simulate activities and project results. This can greatly reduce miscommunication and misunderstanding among team members in the project development and greatly improve the development efficiency. API Gateway allows you to use Mock as the backend service of an API operation, which requires simple configurations.

# **Configure Mock**

In the Define API Backend Service step of the Create API wizard, set Backend Service Type to Mock and complete relevant configurations.

Basic Backend Definition	
Backend Service Type	HTTP(s) Service VPC FunctionCompute Mock You have used Mock, it won't invoke your backend service, please confirm it?

### 1. Specify a Mock response

Enter a response in the Mock Result field. The response will be returned for all requests for the current API operation. You can specify a Mock response in various formats, including JSON, XML, and text. The following code snippet is a sample of a Mock response:

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

After you create an API operation that uses Mock as the backend service, you can publish the API operation to the test or production environment. You can also go to the debugging page to debug the API operation.

### 2. Specify the HTTP status code

The following table lists the valid HTTP status codes. HTTP/1.1 status codes are supported. If you enter a code that is not in the following table, a message appears in red next to the HTTP Status Code field, indicating that the code you entered is invalid.

HTTP status code	HTTP message
200	ОК
201	Created
202	Accepted

HTTP status code	HTTP message	
203	Non-Authoritative Information	
204	No Content	
205	Reset Content	
206	Partial Content	
300	Multiple Choices	
301	Moved Permanently	
302	Found	
303	See Other	
304	Not Modified	
305	Use Proxy	
306	(Unused)	
307	Temporary Redirect	
400	Bad Request	
401	Unauthorized	
402	Payment Required	
403	Forbidden	
404	Not Found	
405	Method Not Allowed	
406	Not Acceptable	
407	Proxy Authentication Required	
408	Request Timeout	
409	Conflict	
410	Gone	
411	Length Required	
412	Precondition Failed	
413	Request Entity Too Large	

HTTP status code	HTTP message	
414	Request-URI Too Long	
415	Unsupported Media Type	
416	Requested Range Not Satisfiable	
417	Expectation Failed	
450	Parameter Required	
451	Method Connect Exception	
500	Internal Server Error	
501	Not Implemented	
502	Bad Gateway	
503	Service Unavailable	
504	Gateway Timeout	
505	HTTP Version Not Supported	

#### 3. Specify Mock header fields

API Gateway supports custom Mock header fields and duplicate header field names. A header field name cannot be empty and can contain digits, letters, underscores (\_), and hyphens (-). The value of a header field cannot be empty.

### **Remove Mock**

To remove Mock as the backend service of an API operation, you only need to configure a different backend service for the API operation. The configurations of Mock will be retained, so that you can still switch to Mock if needed. After you change the backend service of the API operation, the change takes effect only after you publish the API operation.

# 9.Create API operations for data services

Data services are designed to build a uniform data service bus to help enterprises increase the value of their data assets as well as guarantee data reliability, security, and effectiveness. Data services are applicable to the following scenarios:

- Provide data externally at the minimum necessary unit: When an enterprise needs to provide its business data externally, API operations can be defined to provide data at the row level based on filter conditions in SQL statements, and also at the column level based on specific fields. Compared with providing a whole table, this exposes the smallest unit of data and ensures data security.
- Provide data for visualization tools: Most visualization tools support obtaining data from data sources by calling API operations. Instead of using a username and a password to connect a visualization tool to your database, you can provide data for a visualization tool by calling API operations in the tool. This is an easier way of data sourcing and avoids account exposure.
- Provide processed data for applications: After data is processed and summarized by using a data development tool, API operations can be defined and provided for applications to read and use processed data. If the logic of data reading needs to be modified, users only need to modify the query logic for the corresponding API operation and do not need to republish the application.

API Gateway can integrate with the following Alibaba Cloud services to provide data services:

- Big data service: DataWorks. For information about how to use API operations in the DataWorks console, see DataService studio overview.
- Big data service: Dataphin. For information about how to use API operations in the Dataphin console, see Data services.
- Database service: Data Management Service (DMS). For information about how to use API operations in the DMS console, see DataService StudioData API.

# 10.Create API operations for an online prediction service

Alibaba Cloud Machine Learning Platform for AI (PAI) provides an all-in-one solution for both traditional machine learning and deep learning. This platform integrates data processing, model training, service deployment, and online prediction. The deployment of an online prediction service is an important step of applying algorithm models to actual business. To help users apply algorithms from end to end, PAI provides Elastic Algorithm Service (EAS) to support online prediction services that are used in scenarios where online inference is involved.

EAS allows you to publish models to API Gateway by using RESTful API operations. Models that are published to API Gateway can be called in business systems by using HTTP requests. For information about how to publish models to API Gateway, see Model Calling Methods.

# 11.Parameter mapping and verification rules

### ? Note

This topic applies only to shared instances and dedicated instances of the VPC type. This topic does not apply to instances of the classic network type.

# 1. Overview

API Gateway supports parameter mapping and verification. This topic describes the processing rules used when API Gateway forwards client-initiated HTTP requests to an HTTP-based backend service and forwards HTTP responses from the backend service to clients. If the backend service is Alibaba Cloud Function Compute, see Use Function Compute as the backend service of an API operation

API Gateway supports the following transmission modes:

• Passthrough mode: In this mode, API Gateway only maps and verifies request parameters of the path type and transparently forwards the parameters to the backend service. For more

information, see 3 Passthrough mode .

- Mapping mode: In this mode, API Gateway maps and verifies all user-configured parameters. If a request sent by a client contains a parameter that is not configured, API Gateway does not forward it to the backend service. For more information, see 4 Mapping mode.
- Transparent mapping mode: This mode is similar to the mapping mode . However, in

transparent mapping mode , if a request sent by a client contains a parameter that is not configured, API Gateway transparently forwards the parameter to the backend service. For more information, see 5 Transparent mapping mode .

# 2. Locations of parameters and reading rules

API Gateway can read the parameters from different locations of an HTTP request . API Gateway

also supports system parameters and user-configured constant parameters .

### 2.1 <sub>path</sub> parameters

API Gateway can extract parameters from the segmented paths in HTTP requests . If you want to use path parameters, you must configure the API request path in the /path/[parameter] format. Then, API Gateway matches the path in an HTTP request based on your configured parameter path. The following table shows an example.

Configured request path	Input data	Parameter extraction result
/request/to/[path]	/request/to/user1	path=user1
/[path1]/[path2].	/group1/user1	path1=group1, path2=user1
/[root]/*	/root/user1	root=request
/[root]	/root/user1	No match.

If API Gateway detects invalid input data that does not comply with the RFC 3986 specification, it
 returns Error code: I400PH: Invalid Request Path .- The maximum length of a request URI that API
 Gateway supports is 128 KB. If a request URI exceeds the maximum length, API Gateway returns
 Error code: I413RL: Request Url too Large .

### 2.2 query parameters

queryparameters are the values contained in<br/>QueryStringQueryStringof requests. After API Gatewayreceives a request, it splits the key-value pairs in<br/>a and ampersand ( & ) and then usesQueryStringof the request by using the equal sign (=) and ampersand ( & ) and then usesUTF-8for URL decoding. API Gateway considers both? a=and ? aas valid values that are equivalent to empty strings enclosed in a pair of singlequotation marks ( '). The following table shows an example.

Input data	Parameter extraction result
? a=1&b=2	a: "1", b: "2"
? a=1&a=2	a: ["1", "2"]. If this parameter is of a data type other than <b>ARRAY</b> , only the first value is used.
? a	a: ""
? a=	a: ""

Input data	Parameter extraction result		
? =a&b=1	b: "1". API Gateway ignores the =a part of the input data.		

- If API Gateway detects invalid input data that does not comply with the RFC 3986 specification, it returns Error code:I400PH: Invalid Request Path .
- The maximum length of a request URI that API Gateway supports is 128 KB. If a request URI exceeds the maximum length, API Gateway returns Error code: 1413RL: Request Url too Large .

#### 2.3 formData parameters

 formData
 parameters are the values contained in the message body when
 Content-Type
 of a

 request is
 application/x-www-form-urlencoded
 . If
 charset
 in
 Content-Type
 is not specified, API

 Gateway uses
 UTF-8
 encoding. If
 charset
 is specified, API Gateway uses the specified character

 set for URL decoding. API Gateway splits and processes formData parameters in the same way as
 splitting and processing
 QueryString

If Content-Type is multipart/formdata , API Gateway supports parameters of the FILE type.

#### 2.4. header parameters

header parameters are read from HTTP request headers. For example, X-User: aaa is parsed as X-User = aaa . The processing of header parameters has the following special rules:

- Both the spaces before and after the values of header parameters are removed.
- If multiple headers with the same name exist and the parameter type is set to ARRAY, the parameter is parsed as an array. If the parameter type is not set to ARRAY, only the first value is used.
- API Gateway uses ISO-8859-1 encoding to read and forward headers. Invalid characters may cause garbled characters or other unexpected results.

#### 2.5. host parameters

host parameters are valid only if you have bound domain names to a wildcard domain name and added a valid wildcard domain name template. For example, if you bind the wildcard domain name \*.api.foo.com and configure the wildcard domain name template \${user}.api.foo.com , API Gateway reads user = 1234 from the host parameters when it receives a request for 1234.api.foo.com . You can configure multiple wildcard domain name templates. API Gateway parses host parameters based on the first matched record. If no record is matched, API Gateway cannot obtain the host parameters. The following table shows an example.

Configuration of a template for a wildcard domain name	Request host	Parameter extraction result
<pre>\${User}.api.io</pre>	123.api.io	User: "123"
<pre>\${User}. \${Group}.api.io</pre>	123.g01.api.io	User: "123" Group: "g01"
<pre>\${Admin}.admin.api.io \${User}. \${Group}.api.io .</pre>	123.api.io	User: "123"
<pre>\${Admin}.admin.api.io \${User}. \${Group}.api.io</pre>	123.admin.api.io	Admin: "123"
<pre>\${Admin}.admin.api.io \${User}. \${Group}.api.io</pre>	123.u00.api.io	User: "123"GroupId: "u00"
<pre>\${User}. \${Group}.api.io \${Admin}.admin.api.io</pre>	123.admin.api.io	User: "123"Group: "admin"

*In the last row of the preceding table, \${User}.\${Group}.api.io is matched and therefore \${Admin}.admin.api.io is ignored.* 

#### 3. Passthrough mode

The passt hrough mode supports the following methods: GET , PUT , POST , DELETE , PATCH ,

HEAD , OPTIONS .

#### 3.1 Forward client requests to the backend service

In passthrough mode, API Gateway transparently passes requests to the backend service after it processes the signature and authorization. The following passthrough rules apply to parameters in different locations of a request:

• Path: If you set the API request path to the /path/to/[user] format, you can also configure

/path/backend/[user] as the backend service path. API Gateway identifies the frontend path
parameter and maps it to the backend service path.

- **QueryString**: Data of this type is transparently passed to the backend service. The sequence and format of the originally received value of QueryString remain unchanged.
- Header: API Gateway transparently passes all headers except for some system headers and headers that start with  $x_{-Ca-}$  to the backend service and uses ISO-8859-1 encoding to read and

forward these headers. Therefore, if an invalid encoding is passed in a header, an error may occur. For more information about the processing logic of the headers that API Gateway reserve, see

7 HTTP header processing rules

• Body: API Gateway transparently forwards the package body to the backend service. If you set a custom Content-Type in the API configuration, the custom Content-Type is used. Otherwise, API

Gateway forwards a Content-Type header provided by the client.

#### 3.2. Forward backend responses to the client

In passthrough mode, if the backend service returns a response, API Gateway forwards the

HTTP response from the backend service to the client . If the backend service fails to return

a response, API Gateway generates an error code. For more information about how to perform troubleshooting, see Error code table The following passthrough rules apply to the parameters of different types in a request:

- StatusCode: The error code in the response from the backend service is transparently passed.
- Header: API Gateway filters or adds system headers and reserved headers that start with
  - x-ca- and transparently passes other headers in the response from the backend service. For more

information, see 7 HTTP header processing rules .

• **Body**: API Gateway forwards the package body in the response from the backend service to the client. If Content-Type in the response is empty, API Gateway forwards the default value

application/oct-stream .

*You can use the Plug-ins of the Error Mapping type plug-in to change the HTTP status codes in the responses from the client.* 

#### 4. Parameter mapping mode

In parameter mapping mode , API Gateway verifies and maps the parameters that you have

configured. If the client passes a parameter that is not configured, API Gateway does not forward the parameter to the backend service. If you want API Gateway to forward parameters that are not configured to the backend service, see 5 Transparent mapping mode . In parameter mapping mode,

API Gateway supports the parameters of the query , header , host , path , and formData

types. API Gateway can also determine and verify the types of parameter values and forward the parameters to the backend service.

#### 4.1. Parameter types

Verification **Description** Supported format Туре met hod Minimum length, maximum length, String **ST RING** Unlimited enumerated value, and regular expression Minimum, maximum, 100 -1 A 32-bit integer Integer and enumerated values Minimum, maximum, -1233 , 1001 A 64-bit integer Long and enumerated values 100 , 0.1 , Minimum and maximum Double Floating point values 9E-9 , 1.01E16 true and false Boolean BOOLEAN (The value is not casesensitive.) For multipart/form-Minimum length and File File type maximum length data only

The following table lists the parameter types that API Gateway supports.

Туре	Description	Supported format	Verification method
Array	ARRAY	See the array field type.	Verification of array field types

\* Parameters of the FLOAT type are processed in the same way as parameters of the DOUBLE type.

# 4.2. Configuration of parameter verification

• You can configure parameter verification in the API Gateway console , by using

OpenAPI Explorer , or by importing a Swagger file . The following table describes the

configuration of parameter verification.

Parameter verification item	Description	Field on OpenAPI Explorer	Field in Swagger
Param Name	Required. The name must be unique in an API operation.	ApiParameterName	name
Param Location	Required.	Location	location
Туре	Optional. The default type is STRING .	ParameterType	type
Type Of Array Field	Optional. This parameter is required only if the parameter type is ARRAY .	ArrayltemsType	items.type
Required	Optional. The default value is No	Required	required
Default Value	Optional. An empty string enclosed in a pair of single quotation marks ( ' ) is not a valid default value.	DefaultValue	default

Parameter verification item	Description	Field on OpenAPI Explorer	Field in Swagger
Max	Optional. The input value must be less than or equal to the value of this parameter.	MaxValue	maximum
Min	Optional. The input value must be greater than or equal to the value of this parameter.	MinValue	minimum
Max Length	Optional. This parameter is valid only if the parameter type is STRING .	MaxLength	maxLength
Min Length	Optional. This parameter is valid only if the parameter type is STRING	MinLength	minLength
Param Verification	Optional. This parameter is valid only if the parameter type is String	RegularExpression	pattern
Enumeration	Optional.	EnumValue	enum

- For more information about the parameter configuration on OpenAPI Explorer, see Front-end input p arameters
- For more information about parameter configuration on Swagger, see Import Swagger files to create APIs

Matching rules for parameter verification:

- OpenAPI Explorer and Swagger have different definitions for parameter types. The description in this topic is subject to the swagger standard.
- If you do not set the parameter type, the default type string is used.
- If the input format of the parameter type is inconsistent with the format supported by the current type, the error code I400IP Invalid Parameter: ... is reported. ``
- If a request of the client does not contain a required parameter, API Gateway blocks the request and returns the error code I400MP Invalid Parameter Required: ... .
- You can set default values for optional parameters. If the client does not pass an optional parameter for which you set a default value, API Gateway passes the default value to the backend service.
   However, API Gateway considers an empty string enclosed in a pair of single quotation marks ( ') as an invalid default value and does not pass it to the backend service.
- If a parameter is of the query or formData type, API Gateway considers the input value in the format of a or a= such as ? b=1&a as an empty string enclosed in a pair of single quotation marks (').
  - If the parameter is required, no error is returned.
  - If the parameter is optional and a default value is configured, API Gateway passes an empty string rather than the default value to the backend service.
- If the parameter type is INTEGER , LONE , FLOAT , Or DOUBLE and the input value is an empty string enclosed in a pair of single quotation marks ( ' ), the parameter is considered not passed.
  - If this parameter is required, API Gateway blocks the request and returns the error code 400: <I400MP> Invalid Parameter Required: ... .
  - If this parameter is optional and has a default value, API Gateway passes the default value to the backend service.
- Min Length and Max Length take effect only when its configured value is greater than 0.
- The maximum length of a regular expression is 40 characters.
- Parameters of both the STRING and NUMERIC types can be set to enumerated values . The enumerated values must be separated with commas (,), such as river, lake, sea . If the input value is not included in the enumerated values, the error code I400IP: Invalid Parameter:... is returned. ``
- If the parameter type is set to ARRAY , only the parameters of the query , formData , or

header type can be set to arrays. The verification rules for array parameters take effect for each array. The parameter type is specified by the type of the array parameter, and the default type is STRING.
#### 4.3. Parameter mapping rules of backend services

- You can set the backend location and backend name for parameters. API Gateway converts the parameter location and name when it sends requests to the backend service.
- If the parameter type is ARRAY, the backend location can only be query, formData, or
   header. When API Gateway passes such a parameter to the backend service, API Gateway converts the parameter to multiple parameters or multiple headers. For example, it converts a=1,2 to

a=1&a=2 .

- The QueryString that is passed to the backend service is encoded by using UTF-8 and URL encoding .
- If the parameters in a request include formData parameters, API Gateway uses
   application/x-www-form-urlencoded; charset=utf-8
   Or multipart/formdata; charset=utf-8
   to encode the package body and sends it to the backend service.
  - If the parameters in a request contain parameters of the FILE type, API Gateway uses

multipart/formdata; charset=utf-8 to assemble the package body. Otherwise, API Gateway

uses application/x-www-form-urlencoded; charset=utf-8 to assemble the package body.

• If you configure a custom Content-Type in the Define API Backend Service step, API Gateway

sends the custom Content-Type to the backend service. If the custom Content-Type is in the

application/x-www-form-urlencoded; charset=??? **Of** multipart/formdata; charset=???

format, API Gateway uses the encoding format specified in ContentType Value to assemble the package body.

• If the parameter to be forwarded is of the header type, API Gateway uses IS08859-1 to convert and forward this parameter.

#### 4.4. Forward backend responses to the client

In parameter mapping mode, if the backend service returns a response, API Gateway forwards the HTTP response from the backend service to the client. If the processing fails, API Gateway returns an error code. For more information about how to perform troubleshooting, see Error code table The following forwarding rules apply to the parameters of different types in a request:

- StatusCode: The error code in the response from the backend service is transparently passed.
- Header: API Gateway filters or adds system headers and reserved headers that start with

x-ca- and transparently passes other headers in the response from the backend service. For more information, see 5 Header forwarding rules.

• Body: API Gateway forwards the package body in the response from the backend service to the client. If Content-Type in the response is empty, API Gateway forwards the default value

application/oct-stream .

You can use the *Plug-ins of the Error Mapping type* plug-in to change the HTTP status codes in the responses from the client.

#### 5. Passthrough mapping mode

The verification and processing mechanisms for the passthrough mapping mode and

parameter mapping modeare similar. The only difference is that unknown parameters in a request aretransparently passed to the backend service inpassthrough mapping mode. Whereas, the unknownparameters are filtered out inparameter mapping mode.

#### 6. Strict mapping mode

The verification and processing mechanisms for the strict mapping mode and

parameter mapping mode are similar. The only difference is that an error is reported if a request

contains unknown parameters in strict mapping mode .

#### 7. HTTP header processing rules

In normal cases, all headers that start with x-ca- are reserved headers in API Gateway. API Gateway filters out these headers or performs special processing. Do not use headers that start with x-ca-.

HeaderName	Request processing method	Response processing method
Connection	Rebuild	Rebuild
Keep-Alive	Rebuild	Rebuild
Proxy-Authenticate	Rebuild	Rebuild
Proxy-Authorization	Rebuild	Rebuild
Trailer	Rebuild	Rebuild
ТЕ	Rebuild	Rebuild

HeaderName	Request processing method	Response processing method		
Transfer-Encoding	Rebuild	Rebuild		
Upgrade	Rebuild	Rebuild		
Host	Rebuild	N/A		
Authorization	Verify, map, or pass through	N/A		
Date	N/A	Pass through or add a default value		
Content-Type	Map or pass through	Pass through or add a default value		
Content-Length	Map or pass through	N/A		
Content-MD5	Verify and pass through	N/A		
Via	Add records of API Gateway	N/A		
X-Forwarded-For	Add the IP address of the client to the right of the existing value	N/A		
X-Forwarded-Proto	Add a client request protocol: 'http', 'https', 'ws', or 'wss'	N/A		
User-Agent	Pass through or add a UserAgent of API Gateway	N/A		
Server	N/A	Pass through or add a default value		

- All the headers marked as rebuild will not be transparently passed and API Gateway adds a defined value for these headers.
- If the passthrough mode is specified in a request from the client for a header that is not listed in the table, API Gateway transparently passes the header to the backend service. If the mapping mode is specified in the request, all headers except the default HTTP headers are filtered out.

• By default, the headers of responses that are not listed in the table are transparently passed to the client.

### 12.Version management

#### 1. Modify an API

You can view and modify the information of an API as required.

#### Procedure

1. Log on to the API Gateway console.

2. In the left-side navigation pane, choose Publish APIs > APIs .

3. On the API List page, find the API that you want to view and choose More > Manage in the

Operation column. You can view the information of the API. Click Edit in the upper-right corner to edit the information of the API as required.

The modification process of an API is similar to the creation process of an API. For more information, see Create an API.

😑 C-J Alibaba Clo	ouc	China (Beijing) 👻 🔍 Searchi Expenses	Tickets ICP Enterprise Support Official Site 🖂 🎝 🛱 🕐 EN 🍈
<	P	uuuuuuuJa74424786 uuu, uuuu Tafinitii t Back to API list	Ctone Edit Deploy Undeploy
Code Demo		Name And Description	
API Definition		Group: testGroup	API Name: testVpcApi1
Authorization		Security Certification: Alibaba Cloud APP	Whether to allow the cloud market to be listed: No
Plugin tied		Whether to allow public network access: Yes	Whether to set to prevent replay attacks: No
Deployment History Monitoring Info		Signature Method: HmacSHA256	AppCode Certification: Allow AppCode authentication (Header & Query)
Security Info		Description:	
Debug API			
Cloud marketplac		Basic Request Definition	
		Request Type: NOTIFY(WEBSOCKET)	
		Path: /web	Protocol: HTTP
		HTTP Method: POST	Request Mode: Request Parameter Mapping(Filter Unknown Parameters)
	I		

After you modify the information of an API, you must publish the API to the Test, Pre, or Release environment before it can be called.

#### ? Note

If you modify the definition of an API that is published, the modifications are not immediately applied. You must republish the modified API to synchronize the changes to the Release environment.

#### 2. Compare API configurations

After an API is modified, you can **compare** the API configurations before and after the modification to check whether the modification scope is correct.

#### Procedure

1. After you modify an API, click Save . In the dialog box that appears, click Deploy .

2. In the Deploy API dialog box, specify Select The Stage To Release To and click

Contrast difference to compare the API configurations. You can publish the API after you confirm that the modification scope is correct.

Deploy API ×
You will deploy the following API(s):
testVpcApi
Select The Stage To Release To:         Release       Pre         Test         NEW! It is recommended to compare the configuration with the API of the selected release environment, and release after confirming the scope of modification.         Contrast difference         *Enter Change Remarks:         Required. It cannot exceed 180 characters
Description: If you have used Mock Enable, the API will not invoke your backend service, please confirm it with caution. This action will overwrite API of Release, please confirm it with caution. Deploy Cancel
Deploy API X You will deploy the following API(s):
Select The Stage To Release To:         Release       Pre         Test         NEW! It is recommended to compare the configuration with the API of the selected release environment, and release after confirming the scope of modification.         Contrast difference         *Enter Change Remarks:         Required. It cannot exceed 180 characters
Description: If you have used Mock Enable, the API will not invoke your backend service, please confirm it with caution. This action will overwrite API of Release, please confirm it with caution. Deploy Cancel

#### 3. Query release history and switch versions

API Gateway records each release of an API. The release history includes the release version, environment, date, and release remarks. You can view the release records on the Deployment History page.

#### 3.1 Query the release history

1. Log on to the API Gateway console. In the left-side navigation pane, choose Publish APIs >

APIs . On the API List page, find and click the API whose release history you want to query.

2. In the left-side navigation pane, click Deployment History. On the Deployment History page, find the version that you want to view and click **View** in the Operation column.

≡	<b>C-)</b> Alibaba (	Clou	d China (Beijing) 👻	Q Search	Expenses	Tickets	ICP	Enterprise	Support	Official Site	۶	٥.	Å	② EN	C	)
	<		testtest - Deployment History	Back to API list												
			Version	Deployment Notes	Stage (All) -		Deplo	yment Time			Operat	ion				
C	Code Demo		20201110173059434	t	Release		Nov 1	0,2020 17:30:5	•		View	Copy	Switch	this vers	on	
A	PI Definition		20201110173344670	t	Release		Nov 1	0.2020 17:33:4			View	Copy	Switch	o this vers	on	-
A	authorization											0000				-
F	lugin tied		20201113160251743	tesat	Release		Nov 1	3,2020 16:02:5			View	Copy	Switch	to this vers	on	
г	Depleyment History		20201120112104849	test	Release		Nov 2	0,2020 11:21:04	1		View	Copy	Switch	to this vers	on	
N	Aonitoring Info		20201120112128334	test	Release		Nov 2	0,2020 11:21:28	3		View	Copy	Switch	to this vers	on	
s	Security Info	æ	20201120112445401	test	Release		Nov 2	0,2020 11:24:4	5		View	Copy	Switch	to this vers	on	
C	ebug API		20201124115908026	test	Release		Nov 2	4,2020 11:59:00	3		View	Copy	Switch	to this vers	on	
+ C	loud marketplac		20201127132617461	发布	Release		Nov 2	7,2020 13:26:1			View	Copy	Switch	to this vers	on	
			20201201112417482	fabu	Release		Dec 0	1,2020 11:24:1	,		View	Copy	Switch	to this vers	on	
			20201201113312946	发布	Release		Dec 0	1,2020 11:33:12	2		View	Copy	Switch	to this vers	on ←	
									Tota	al of 12 entries, 10	) display	ed per pa	age	1 2		5

#### 3.2 Switch release versions

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

- 1. Find the required version and click Switch to this version in the Operation column.
- 2. In the API Version Switch dialog box, enter the description and click Switch.

You will switch API: <b>testVpcApi</b> in <b>Release</b> , it will come into effect immediately,please confirm this. Current version: 20201209162838031 Ewitch to Version: 20201209162404204 Einter the description:	
Current version: 20201209162838031 witch to Version: 20201209162404204 inter the description:	
witch to Version: 20201209162404204	
inter the description:	
Required. It cannot exceed 180 characters	
	//
Switch Cance	el

### 13.Configure different environments for an API operation

#### 1. Overview

API Gateway allows you to configure the test, staging, and production environments for an API operation, which correspond to the Test, Pre, and Release environments in the API Gateway console, respectively. This meets your need to call an API operation in different R&D scenarios. For example, you can configure the test environment for an API operation. In the test environment, the API operation uses appropriate resources in the test environment as its backend service. In this way, testers can simulate API calls in the production environment.

To configure different environments for an API operation in the API Gateway console, you must perform the following operations:

- Configure an environment variable in the API group to which the API operation belongs. Specify different values for the environment variable in the test, staging, and production environments. Use this environment variable to configure the backend service of the API operation. In this way, API Gateway will route requests for the API operation to different backend service addresses as configured.
- Configure that each API request from clients must specify the environment in which the API operation is to be called. API Gateway provides you with two methods. You can bind the API group, to which the API operation belongs, to different domain names for different environments. Alternatively, you can add the

X-Ca-Stage

parameter as a request parameter of the API operation.

This topic provides three examples of how to configure different environments for an API operation. The backend services of the three API operations are a service that is deployed in a virtual private cloud (VPC), an HTTP service, and a function that is created in the Function Compute console, respectively. In the three examples, one or both of the methods that API Gateway provides are used to ensure that each API request specifies the environment in which the API operation is to be called. The examples in this topic will help you understand the fundamental working principles. Even if your business situation is different from all the examples, you will figure out how to configure different environments for an API operation based on your business requirements.

#### 2. Configure different environments for an API operation whose backend service is in a VPC by adding the X-Ca-Stage parameter as a request parameter of the API operation

#### 2.1 Preparations

In this example, create two Elastic Compute Service (ECS) instances in the ECS console, as shown in the following figure. The two ECS instances must reside in two different VPCs. One of the ECS instances hosts the backend service of an API operation in the production environment. The other ECS instance hosts the backend service of the API operation in the test environment.

Elastic Compute Service	Instances			Create Instance	Diagnose Bu	ulk Action
Overview	You have 2 burst instances with limited performance in the last 24 hours. Please	pay attention to and handle them in time so as	not to affect the normal operation of applications on	the instances.Handle		
Events 🔕	* Select an instance attribute or enter a keyword	Q Tags			Advanced Search	<u>z</u> o
Tags	Instance ID/Name     Tag     Monitoring Zone +	IP Address Statu	s - Specifications	Billing Method +		Actions
Resource Orchestration 🖾 🚥	North China	Zone H	unning 1 vCPU 1 GIB (I/O Optimized) ecs.s6-c1m1.small OMbps (Peak Value)	Pay-As-You-Go August 11, 2020, 10:42 Created	Manage Change Instance Typ	e   Connect
Container Service fo 🕻 📼	Seijing Zone	() St	topped 2 vCPU 4 GiB (I/O Optimized) ecs.n1.medium 0Mbps (Peak Value)	Pay-As-You-Go August 6, 2020, 16:20 Created	Change Instance Typ	Manage Ie   More 👻
Instances & Images  Instances Elastic Container Instance	C Beijing Zone	⊙Ru	unning 2 vCPU 4 GiB (I/O Optimized) ecs.n1.medium 0Mbps (Peak Value)	Pay-As-You-Go August 6, 2020, 16:20 Created	Manage Change Instance Typ	3   Connect ie   More ▼

Make sure that TCP port 80 of each ECS instance is available. Install NGINX on each ECS instance as a web server for providing HTTP services. For each HTTP request, the two ECS instances return {"env":"test env"} and {"env":"relese env"}, respectively.

#### ? Note

This example is used only to describe how to configure different environments for an API operation in API Gateway. Therefore, factors that are important to a backend service in reality, such as high availability, scalability, and security, are not carefully considered in this example. We recommend that you take these factors into account when you deploy your own backend services.

#### 2.2 Configure backend service information for the API operation

**Step 1**: In this example, the backend service of the API operation is deployed in two VPCs. Therefore, you must authorize API Gateway to access the two VPCs by creating two authorization entries in the API Gateway console: backend-vpc-release and backend-vpc-test. The two authorization entries are used to authorize API Gateway to access the backend service of the API operation in the production environment and the test environment, respectively. For more information, see Create an API operation with a resource in a VPC as the backend service. The following figure shows the two authorization entries.

VPC Access List							
Please Enter the Authorization Name to search		Search					Create VPC Access
VPC Access Name	Vpc Id		Instance Id	Port	Time Created	Operation	
backend-vpc-release	vpc. <sup>om.7h</sup> jo.4ten.4halddk			1000	Dec 02,2020 17:12:52	Delete	
backend-vpc-test	vpc				Nov 27,2020 16:51:56	Delete	

**Step 2**: Create an API group in the API Gateway console. On the Group List page, find the API group and click view stages in the Operation column.

ApiGateway	Group List							
Overview	Enter the GroupName to que	ry			Search Tags			Create Group
Instances	Group Name	Tag	Monitor	Description	Created Time (Descending order) -	instanceType (All) +	Operation	
✓ Publish APIs	deal Transform Resource	۲			Nov 30,2020 21:08:17	Defeated (PA) and the second state	View APIs   Bind Domain View Stages V Models   Delete	/lew
API Groups	test/relayCourt	۲			Nov 30,2020 20:36:39	Dedicated UPC / animateuro	View APIs   Bind Domain View Stages V Models   Delete	/iew

**Step 3**: On the Stage Management page, create an environment variable on both the Release and Test tabs. In this example, create an environment variable named backend-host. Specify its value as backend-vpc-release on the Release tab and backend-vpc-test on the Test tab, as shown in the following figures.

Stage Management & Back to group list				
Release Pre Test			Add Variable	Refresh
Variable Name	Variable Value	Operation		
backend-host	backend-vpc-test	Delete		

Stage Management				
Release Pre Test			Add Variable	Refresh
Variable Name	Variable Value	Operation		
backend-host	backend-vpc-release	Delete		

#### ? Note

For each environment variable, keep its name consistent in different environments. If you have multiple API operations in the same API group and you need to configure an environment variable for each API operation, we recommend that you set an informative name for each environment variable for easy identification.

Step 4: Create an API operation in the current API group. In the<br/>enter the name of the environment variable in the<br/>wPC Access NameDefine API Backend Servicestep,#backend-host#.VPC Access Namefield. In this example, enter

Create API	✿ Back to API list					
	Basic Information	$\rangle$	Define API Request		Define API Backend Service	efine Response
Basic Back	end Definition					
	Backend Service Type	HTTP(\$) Service OVPC	FunctionCompute Mock			
	VPC Access Name	#backend-host# Fill in vpc name, vpc name su How to use VPC? How to us	pport environment variables e environment variables?	Using the HTTPS p	rotocol Creating Or Searching VPC Access	

#### ? Note

In the API Gateway console, each environment variable must be expressed in the #Variable name# format, for example, #Service# and #Function#.

**Step 5**: Complete other configurations for the API operation. Then, publish the API operation to the production and test environments, as shown in the following figure.

API List							
API Name		Search	N Tags		Create API	Import Swagger	Create Data Service APIs
API Name	Tag	Visibility	Group	Description	Last Modified	Stage (All) +	Operation
0	۲	Public		d		Release (Running) Test (Running) Pre	Deploy   Debug   More -

**Step 6**: Authorize two applications to call the API operation. Note that when you authorize an application, you must specify the environment in which the application can call the API operation, as shown in the following figure. You can also authorize an application to call an API operation in all environments. For demonstration purposes, the API operation can be called anonymously in this example.

<	testVpcApi - Authorization Infor	mation t Back to API list				
Code Demo	APP ID		Enter the APP ID		Search	Add Authorization
API Definition	APP ID	APP Name	Stage (All) -	Authorization Time	Authorizer	Authorization Period
Authorization		testVpcAp	Test	Dec 02,2020 17:23:58	API Provider	Long-term
Plugin tied	0	testVpcAp	Release	Nov 30,2020 18:02:53	API Provider	Long-term
Deployment History	Revoke Authorization					Total of 2 entries, 10 displayed per page 🔷 1 🛶
Monitoring Info						
Security Info						
Debug API						
Cloud marketplac						

# 2.3 Add the X-Ca-Stage parameter as a request parameter of the API operation

• To call the API operation in the production environment, you do not need to specify the X-Ca-Stage parameter in each API request, as shown in the following figure.



• To call the API operation in the staging environment, which is inapplicable in this example, you must add the key-value pair

#### X-Ca-Stage: PRE

in each API request.

• To call the API operation in the test environment, you must add the key-value pair

#### X-Ca-Stage: TEST

in each API request, as shown in the following figure.



#### 3. Configure different environments for an API operation whose backend service is an HTTP service by binding the API group to different domain names for different environments

#### 3.1 Preparations

In this example, create two ECS instances in the ECS console, as shown in the following figure. One of the two ECS instances hosts the backend HTTP service of an API operation in the production environment. The other ECS instance hosts the backend HTTP service of the API operation in the test environment. Make sure that TCP port 80 of each ECS instance is available. Install NGINX on each ECS instance as a web server for providing HTTP services. For each HTTP request, the two ECS instances return {"env":"test env"} and {"env":"relese env"}, respectively.

Elastic Compute Service	Instances								C Create Instance	Diagnose B	Julk Action
Overview	You have 2 burst instances with limi	ted performance	e in the last 24	hours. Please pay atten	ition to and handle them in tin	ne so as not to	affect the normal operation of applications or	the instances.Handle			
Events 🔕	* Select an instance attribute or en	nter a keyword		0	Q, Tags					Advanced Search	<u>z</u> o
Tags	Instance ID/Name	Tag	Monitoring 2	Zone 👻	IP Address	Status 👻	Specifications	Billing Method 👻			Actions
Resource Orchestration 🖾 🚾		* **		North China 2 Zone H		(•) Running	1 vCPU 1 GiB (I/O Optimized) ecs.s6-c1m1.small OMbps (Peak Value)	Pay-As-You-Go August 11, 2020, 10:4	2 Created	Manage Change Instance Typ	e   Connect pe   More +
Container Service fo ピ 🚥		• •		Beijing Zone A		Stopped	2 vCPU 4 GiB (I/O Optimized) ecs.n1.medium 0Mbps (Peak Value)	Pay-As-You-Go August 6, 2020, 16:20	Created	Change Instance Typ	Manage pe   More +
Instances & Images 🔤 🔿	-	• •		Beljing Zone A	-	Running	2 vCPU 4 GiB (I/O Optimized) ecs.n1.medium OMbps (Peak Value)	Pay-As-You-Go August 6, 2020, 16:20	Created	Manage Change Instance Typ	e   Connect pe   More 👻

#### ? Note

This example is used only to describe how to configure different environments for an API operation in API Gateway. Therefore, factors that are important to a backend service in reality, such as high availability, scalability, and security, are not carefully considered in this example. We recommend that you take these factors into account when you deploy your own backend services.

#### 3.2 Configure backend service information for the API operation

**Step 1**: Create an API group in the API Gateway console. On the Group List page, find the API group and click View Stages in the Operation column.

**Step 2**: On the **Stage Management** page, create an environment variable on both the Release and Test tabs. In this example, create an environment variable named backend-host. Specify its value as the public IP addresses of the two ECS instances correspondingly on the Release and Test tabs, as shown in the following figures.

Stage Management & Back to group list				
Release Pre Test			Add Variable Refresh	
Variable Name	Variable Value	Operation		
backend-host	Table Carlos	Delete		
Stage Management t Back to group list				
Release Pre Test			Add Variable Refresh	
Variable Name	Variable Value	Operation		
backend-host	and a state	Delete		

**Step 3**: Create an API operation in the current API group. In the Define API Backend Service step, enter the name of the environment variable in the Backend Service Address field. In this example, enter http://#backend-host#.

Create API & Back to API list				
Basic Information	$\rangle$	Define API Request	Define API Backend Service	$\rangle$
Basic Backend Definition				
Backend Service Type	• HTTP(s) Service VPC	FunctionCompute Mock		
Backend Service Address	http://#backend-host# A backend service address is	the domain name or IP address used by the A	API Gateway to call underlying services, not including the path	

**Step 4**: Complete other configurations for the API operation. Then, publish the API operation to the production and test environments.

**Step 5**: Bind the API group to different domain names for API calls in different environments. On the Group List page, find the API group to which the API operation belongs and click Bind Domain in the Operation column.

Group List						
Enter the GroupName to que	y			Search Search		Create Group
Group Name	Tag	Monitor	Description	Created Time (Descending order) -	instanceType (All) -	Operation
	۲			Nov 30,2020 21:08:17	Television (***) (exceptional of attraction)	View APIs Bind Domain View Stages   View Models   Delete

**Step 6**: Bind the API group to two domain names, one for the test environment and the other for the production environment, as shown in the following figure.

Custom Domain Name				Bind Domain
Custom Domain Name	WebSocket Channel Status	Domain Legal Status	SSL Certificate	Operation
	Not Open (Open)	Normal (RELEASE)	Select Certificate	Delete Domain   Change Stage
uislan feafleang-see	Not Open (Open)	Normal (TEST)	Select Certificate	Delete Domain   Change Stage

#### 3.3 Call the API operation by using the domain names

• To call the API operation in the production environment, use the domain name for the production environment in each API request, as shown in the following figure.



- To call the API operation in the staging environment, which is inapplicable in this example, use the domain name for the staging environment in each API request.
- To call the API operation in the test environment, use the domain name for the test environment in the API request, as shown in the following figure.



Note: In an API request, the domain name of the API group to which the API operation belongs takes precedence over the specified X-Ca-Stage parameter. Assume that you have bound the API group to different domain names for different environments and the X-Ca-Stage parameter is also specified in an API request. API Gateway will route the API request to an appropriate environment based on the domain name in the API request, as shown in the following figure.



4. Configure different environments for an API operation with Function Compute as the backend service by binding the API group to different domain names for different environments and also adding the X-Ca-Stage parameter as a request parameter of the API operation

#### 4.1 Configure backend service information for the API operation

Create an API operation. In the Define API Backend Service step, configure a function that you created in the Function Compute console as the backend service of the API operation, as shown in the following figure. For more information, see Use Function Compute as the backend service of an API operation.

Basic Backend Definition		
Backend Service Type	HTTP(s) Service VPC FunctionCompute Mock If no Function, you should create Function first on FunctionCompute console For more detail, please see The Function Compute is used as the backend s For more detail, please see The Function Compute is used as the backend s	e. ervice of API Gateway. ervice of API Gateway.
Function Type	• Event Function OHTTP Function	
Region	China North 2 (Beijing)	Function Compute Console
	Function compute communicates with API Gateway through intranet while the	ney are in same region.
Service Name	testFunctionServer	
Function Name	testFunctionCompute	]
Function Alias	Default Function Alias (LATEST)	]
Backend Timeout	10000 ms	
Service Name Function Name Function Alias Backend Timeout	testFunctionServer testFunctionCompute Default Function Alias (LATEST) 10000 ms	

To configure different environments for the API operation, you can use the following methods:

• Create an environment variable and set its value to different service names or function names. Then, enter the name of the environment variable in the

Service Name
Or

Function Name

field. In this way, requests for the API operation will be handled by different services or functions in different environments.

• In the Function Compute console, create an alias for each version of a service that you want to use as the backend service of the API operation. Service aliases are used to support the version management feature of Function Compute. You can create an alias for each version of a service. For more information, see

Alias operations.

• In the API Gateway console, create an environment variable and set its value to the aliases of different versions of the service. Then, enter the name of the environment variable in the Function Alias field. In this way, requests for the API operation will be handled by different versions of the service.

#### 4.2 Bind the API group to different domain names for different environments and add the X-Ca-Stage parameter as a request parameter of the API operation

Assume that you want API callers to specify the X-Ca-Stage parameter in each API request that uses an independent domain name of the API group to which the API operation belongs. When you bind the independent domain name to the API group, set the Stage parameter to Default(Use X-Ca-Stage to determine the stage), as shown in the following figure.

Make sure the custom do	main name to bind has already been resolved to the subdomain name of this
group. Otherwise it will no	ot be invoked after binding. View subdomain name
Up to 5 domain names ca	an be bound to a group.
Group Name:	testHttpGroup
*Domain Name:	
	The VPC instance already supports the function of extensive domain name. Bind the extensive domain name in the format of `*.api.foo.com`. Click to open the help document
Stage:	Default(Use X-Ca-Stage to determine the str \$
	For more information about environmental management, Click to open the help document
Bound Domains:	yichao.fredhuang.com

#### 5. Limits

- After you modify an environment variable for an API operation, the modification takes effect only after you republish the API operation.
- A maximum of 50 environment variables can be created in each environment.

### 14.Enable HTTP/2 for an API group

#### Overview

API Gateway supports HTTP/2 and provides features such as multiplexing of requests and responses and header compression.

• Multiplexing: In HTTP 1.x, sending and parallel processing of requests and responses require multiple TCP connections. However, in HTTP/2, a sender can divide an HTTP request or response into independent frames, send them in disordered streams to a receiver, and then recombine the streams at the receiver. This reduces latency and improves efficiency. Even when a client needs to send a large amount of request data, the client can transmit the request data by using a single or a few TCP connections.



### HTTP/2 Inside: multiplexing

Header compression: The header of an HTTP request contains a large amount of data. In HTTP 1.x, each HTTP request must include a complete header. In HTTP/2, header field tables are created after a connection is established between a client and a server. The header field tables are used to store key-value pairs in the header of each HTTP message. If a key-value pair in the header of an HTTP message already exists in the header field tables, the key-value pair will not be sent to the receiver with the rest of the message. The header field tables are updated every time an HTTP message is sent, and deleted only after the client and the server are disconnected. Each new key-value pair is added to the end of the header field tables or replaces an existing key-value pair with the same key name. The header field tables help reduce the amount of data in each HTTP request.

Request #1	method scheme host path accept	GET https apidemo.com /path text/xml		method:GET scheme:https host:apidemo.com path:/path accept:text/xml	
Client					Server
	method	GET			
	scheme	https			
Request #2	host	apidemo.com	-	path:/New_path	
	path	/New_path			
	accept	text/xml			

#### Enable HTTP/2 for an API group

• Enable HTTP/2 for an API group that is created after July 14, 2017

All API operations that support HTTPS requests automatically use HTTP/2 for communication between clients and API Gateway. HTTP/2 can be used only by API operations that support HTTPS requests. Therefore, if you want an API operation to use HTTP/2, you must enable HTTPS for the API operation. For more information, see Access a domain name by using HTTPS.

• Enable HTTP/2 for an API group that was created before July 14, 2017

HTTP/2 is not supported by API groups that were created before July 14, 2017. You will soon be able to manually enable HTTP/2 for an API group that was created before July 14, 2017.

### 15.Manage models

#### 1. Overview and limits

A model is used to configure HTTP request and response information for API operations. API Gateway allows you to use JSON Schema to create a model that defines API request and response information, such as request and response parameters and their value ranges. If you create a model and reference the model when you create an API operation, API Gateway automatically generates a POJO class in the SDKs of the API group to which the API operation belongs. This makes it easier for you to configure request and response information for an API operation and deserialize response data.

In API Gateway, models are created based on the standards in JSON Schema Draft-04. For more information, see JSON Schema: core definitions and terminology. Take note of the following limits when you create a model: You can create JSON schemas only of the object type. When you define request and response information for an API operation, you can reference only a model that is created by you. You can call the Create Model and Get a model you've created operations to query the URI of a model that you can use to reference the model. Circular reference is not supported.

The following code snippet shows an example of a model definition.

```
{
        "required": ["name", "photoUrls"],
        "type": "object",
        "properties": {
                "id": {
                        "format": "int64",
                        "type": "integer"
                },
                "category": {
                        "$ref": "https://apigateway.aliyun.com/models/bbc725be4b0b48b79bdd2
f6ebbdcc8c0/a5e7741d8a3a4bcb9746275a0db15fcf"
                },
                "name": {
                        "pattern": "^\\d{3}-\\d{2}-\\d{4}$",
                        "type": "string"
                },
                "status": {
                        "type": "string"
                },
                "dogProject": {
                        "type": "object",
                        "properties": {
                                "id": {
                                         "format": "int64",
                                         "maximum": 100,
                                         "exclusiveMaximum": true,
                                         "type": "integer"
                                 },
                                 "name": {
                                         "maxLength": 10,
                                         "type": "string"
                                }
                        }
               }
       }
}
```

#### 2. Create a model

You can call the Create Model operation, provided by Alibaba Cloud, to create a model in API Gateway. Alternatively, you can create a model in the API Gateway console.

Create a model in the API Gateway console

- 1. In the left-side navigation pane, choose Publish APIs > API Groups.
- 2. On the Group List page, find the target API group and click View Models in the Operation column.

ApiGateway	Group List				
Overview	Enter the GroupName to q	luery	Search 📎 Tags		Create Group
Instances	Group Name	Tag Monitor Descrip	ation Created Time (Descending order) -	instanceType (All) ~	Operation
✓ Publish APIs	test <sup>C</sup> upsticeOnur	» []	Nov 30,2020 21:08:17	Dedicated VPC (anisotrougy to introductor)	View APIs   Bind Domain   View Stages   View Models   Delete
API Groups	text I makes Concern	N	Nov 30.2020 20:36:39	Dedicated VPC (	View APIs   Bind Domain   View Stages   View
APIs					Models Delete

Create a model by importing an OpenAPI specification file

API Gateway allows you to create a model by importing an OpenAPI specification file. For more information, see Create APIs By Importing Swagger. After you import an OpenAPI specification file in which a model is defined for an API group, the model is automatically created for the API group in the API Gateway console. Note that if a model in the OpenAPI specification file has the same name as an existing model in the API Gateway console, the model in the OpenAPI specification file will replace the existing model and you will not be notified.

#### 3. Modify and view a model

After you create a model, you can view the model on the Model Management page. On the model details page, you can view the model name, the model definition, and the URI that is allocated to the model by the system. When you define a model, you can reference another model by using the \$ref: {URI} format.

If you need to modify the model, click Modify in the upper-right corner. Note that after a model is modified, its URI is not changed.

ApiGateway	apigateway.view.model.detail.title 1 Back to	pigateway.view.model.detail.title t Back to group list								
Overview	Basic Information	Basic Information								
Instances	Model Name: ServiceRecordDetailVO	URI: https://apigateway.aliyun	Group ID: 371bc0122f8a4cmitthinT0-inT1-indc07c0a							
✓ Publish APIs	Description:									
API Groups	Model Definition									
APIs										
Plugin	<pre>{"type":"object","properties":{"serv cordDetailId":{"description":"</pre>	<pre>iceRecordId":("description":"</pre>	HOTLINE:  } TICKET:  ',"type":"string"},"re							
VPC Access										
Log Manage										
Owned APIs SDK										
Consume APIs										
Documentation										

#### 4. Delete a model

You can delete a model that is defined for an API group. Note that after you delete a model, the request and response information that is defined by the model for an API operation will also be deleted. For an API operation that references the model and is published to the production environment, the SDKs of the API group to which the API operation belongs may fail to be exported. Therefore, proceed with caution when you delete a model.

# 16.Implement bidirectional communication

#### 1. Overview

Most features of a mobile application, such as user registration and commodity list retrieval, are implemented in the process in which a client sends requests to a server and then the server responds to the client.

However, for specific features such as instant messaging, of a mobile application, a server needs to push in-app notifications to clients. In this case, a communication channel needs to be established between the server and each client, so that the server can send notifications to the clients. Bidirectional communication between a server and a client must be supported.

The architecture of a mobile application must support the bidirectional communication feature.

#### Usage notes

API Gateway provides the bidirectional communication feature in all regions based on the WebSocket protocol. The bidirectional communication feature is supported by SDKs for Android, Objective-C, and Java.

If you need SDKs for other programming languages, submit a ticket or search for group ID 11747055 in DingTalk and join the group to seek help.

#### How it works

API Gateway provides the bidirectional communication feature in all regions. To implement bidirectional communication between a server and a client, you only need to create three API operations in API Gateway, download the SDK that is automatically generated by API Gateway, and then install the SDK on the client.

The following flowchart shows the process to implement bidirectional communication between a backend service and a client by using API Gateway.



#### Process

(1) A WebSocket connection is established between the client and API Gateway when the client is started. After the connection is established, the client sends a message that contains its device ID to API Gateway.

(2) The client initiates a registration signaling message to API Gateway by using the WebSocket channel.

(3) API Gateway converts the registration signaling message to an HTTP request, adds the device ID of the client to a header field named x-ca-deviceid in the request, and then sends the request to the backend service.

(4) The backend service receives the HTTP request and verifies the identity of the client. If the client passes the verification, the backend service records the device ID of the client and responds to the client with HTTP status code 200.

(5) The backend service sends a server-to-client notification signaling message to API Gateway by using the HTTP, HTTPS, or WebSocket protocol. The signaling message includes the device ID of the client.

(6) API Gateway parses the server-to-client notification signaling message, finds the connection to the client whose device ID is specified in the signaling message, and then sends the notification to the client by using the WebSocket connection.

(7) If the client no longer wants to receive notifications from the backend service, the client sends a logoff signaling message to API Gateway by using the WebSocket connection. The logoff signaling message does not include the device ID of the client.

(8) API Gateway converts the logoff signaling message to an HTTP request, adds the device ID of the client to the request, and then sends the request to the backend service.

(9) The backend service deletes the device ID of the client and responds to the client with HTTP status code 200.

## 2. Three types of signaling messages that are used for bidirectional communication

Before you use the bidirectional communication feature of API Gateway, you must familiarize yourself with the three types of signaling messages that are used for bidirectional communication. Each type of signaling corresponds to an API operation of API Gateway. Therefore, you must first create three API operations in the API Gateway console.

#### 2.1 Registration signaling

A registration signaling message is sent from a client to a backend service. This type of signaling serves the following purposes:

(1) Carries the device ID of the client. After the backend service receives an HTTP request that is converted from the registration signaling message, the backend service records the device ID of the client. Note that you do not need to specify the device ID of the client in the registration signaling message. The device ID is automatically generated by the SDK of API Gateway.

(2) Works as an API operation that carries a username and a password. After the backend service receives an HTTP request that is converted from the registration signaling message, the backend service checks whether the username and password of the client are valid. If the backend service responds to the client with an HTTP status code other than 200, API Gateway returns an error to the client, which indicates that the registration failed.

If a client wants to receive notifications from a backend service, the client needs to send a registration signaling message to API Gateway. If the backend service responds with HTTP status code 200, the registration is successful.

#### 2.2 Server-to-client notification signaling

After a backend service receives an HTTP request that is converted from a registration signaling message sent from a client, the backend service records the device ID of the client. In this way, the backend service can include the device ID of the client in server-to-client notification signaling messages and sends the signaling messages to API Gateway. If the client is connected, API Gateway can send the server-to-client notifications to the client.

#### 2.3 Logoff signaling

If a client no longer wants to receive server-to-client notifications from a backend service, the client sends a logoff signaling message to API Gateway. After the client receives HTTP status code 200, the client no longer receives server-to-client notifications from the backend service.

#### 3. Configure bidirectional communication in API Gateway

## 3.1 Enable the WebSocket channel feature for an independent domain name of an API group



#### 3.1.1 Create an API group

If you already have an API group, skip this step.

To use the basic features of API Gateway, you must first create an API group.

#### 3.1.2 Bind an independent domain name to the API group

If you already bound an independent domain name to the API group, skip this step.

After you create an API group, you must bind an independent domain name to it. For information about how to bind an independent domain name to an API group, see Bind a wildcard domain name to an API group.

# 3.1.3 Enable the WebSocket channel feature for the independent domain name

After you bind the independent domain name to the API group, you must enable the WebSocket channel feature for the independent domain name.

Perform the following steps: Log on to the API Gateway console. In the left-side navigation pane, choose Publish APIs > API Groups. On the Group List page, find the API group that is bound with the independent domain name and click the group name. On the Group Details page, find the independent domain name for which you want to enable the WebSocket channel feature in the Custom Domain Name section and click Open in the WebSocket Channel Status column.

ApiGateway	Basic Information		Turn on cloud monitoring Api L	ist Modify Group Message						
Overview	Region: China North 2 (Beijing)	Group Name:		Group ID:						
Instances		Internet Subdomain: (The subdomain is only for API test, whe	in the client directly calls it, there will b	Disable Internet Subdomain						
▼ Publish APIs	Subdomain Name	independent domain name for group bin	independent domain name for group binding, and it will not be subject to this restriction. For details, see configuration process )							
API Groups		API gateway self-calling domain name: N VPC Intranet Subdomain: Not activated F	API gateway self-calling domain name: Not activated ,Please activate on the instance first VPC Intranet Subdomain: Not activated Please set Visit to VPC' in 'instance'							
APIs										
Plugin	Instance Type: Dedicated VPC	Group Traffic Limit (QPS): 2500		Modify API Group's Instance	Instance Type And Selection Guide					
VPC Access	Instance Name: *******	(Consistent with the dedicated instance)		Houry for Facoup of Houriso						
Log Manage Owned APIs SDK	Network Access Policy	HTTPS Security Policy: HTTPS2_TLS1_0 (Be consistent with the dedicated instance	HTTPS Security Policy: HTTPS2.TLS1.0 HTTPS Security Policy Documentation (Be consistent with the dedicated instance HttpsPolicy)							
Consume APIs     Documentation	Logal Status: NORMAL									
	Description:									
	Custom Domain Name		Bind Domain							
	Custom Domain Name Web	Socket Channel Status	Domain Legal Status	SSL Certificate	Operation					
	Not	Open (Open)	Normal	rmal Select Certificate						

3.2 Create three API operations, one for registration signaling, one for server-to-client notification signaling, and one for logoff signaling

Create three API operations in the API group you created. For information about how to create an API operation, see Create an API operation.

When you create the three API operations, set the Request Type parameter based on the signaling type that corresponds to the API operation you create.

Basic Request Definition	
Request Type	
Protocol	
Custom Domain Name	Bind domain name to the group
Subdomain Name	17 1001 - 10015 100 - 15 10 - 10 10 10 10 10 10 10 10 10 10 10 10 10
Request Path	Match All Child Paths
	The request path must contain the Parameter Path in the request parameter within brackets ([]). For example: /getUserInfo/[userId]
HTTP Method	GET \$
Request Mode	Request Parameter Mapping(Filter Unknown Parameters)

#### 3.2.1 Create an API operation for registration signaling

A registration signaling message is sent from a client to a backend service. When you create the API operation for registration signaling, take note of the following items:

(1) We recommend that you configure two request parameters: userName and password, as shown in the following figure.

Basic Request Definition											
	Request Type			UNREGISTER(WEBSOCKET	) NOTIFY(WEBSOCKET)						
	Protocol	WEBSOCKET									
Custom Domain Name											
Subdomain Name		CON 1750 00 11 11 - 100 1	enter 17 not the second s								
Request Path		/register	/register Match All Child Paths								
		The request path must conta	The request path must contain the Parameter Path in the request parameter within brackets ([]). For example: /getUserInfo/[userId]								
	UTTO Method	0007									
HTTP Method POST			•								
	Request Mode	Request Parameter Mappin	g(Filter Unknown F	Parameters) \$							
	Body Data Type	Form		\$							
All request	parameters must have unique na	ames, including the dynamic	parameters in th	e path, headers paramete	rs, query parameters, body para	ameters (form parameters).					
Input Para	meter Definition										
Order	Param Name Param Locat	ion Type	Required D	efault Value	Example		Description	Operation			
↓ ↑	userName Body	¢ String ¢						More   Remove			
↓ ↑	passward Body	\$ String \$	•					More   Remove			
+ Add											

(2) The Protocol parameter must be set to WEBSOCKET.

When you create the API operation for registration signaling, you can configure the request parameters as required. Note that if a client sends a registration signaling message, the registration is successful only after the client receives an HTTP status code of 200.

# 3.2.2 Create an API operation for server-to-client notification signaling

A server-to-client notification signaling message is sent from a backend service to a client. When you create the API operation for server-to-client notification signaling, take note of the following items:

(1) We recommend that you do not authorize the application whose service is supported by the backend service to call this API operation. This way, the API operation can be called only by the backend service and cannot be called by clients.

(2) You can set the Protocol parameter to HTTP, HTTPS, or WEBSOCKET.

(3) This API operation is used by a backend service to send notifications to clients. Therefore, you do not need to configure a backend service for this API operation.

(4) A request parameter named x-ca-deviceid is automatically defined for this API operation and cannot be changed. This parameter must be included in each request for this API operation.

The following figure shows the page on which you can configure request information for this API operation.

Basic Request Definition						
Request Type	Соммон					
Protocol		ITTPS 🗹 WE	BSOCKET			
Custom Domain Name		som				
Subdomain Name			14 , d	lapi.com		
Request Path	/notify			Match All Child Paths		
	The request p	ath must con	tain the Parameter Path in the re-	quest parameter within brackets ([]). For example: /getUserInfo/[userId]		
HTTP Method	POST			÷		
Request Mode	Request Pa	ameter Mapping(Filter Unknown Parameters)				
Body Data Type	Other (JSO	N / file etc.)		•		
Input Parameter Definition						
Param Name Param Location	Туре	Required	Default Value	Example		
x-ca-deviceid Head	String		No	6690A727-4E9A-4611-9022-47FD28909831@10025591		
Body Conte	Model	Select Mode	I ema definition for the Body, whic	ch is not required, and perfecting the model definition will make your SDK and document generation more friendly		

#### 3.2.3 Create an API operation for logoff signaling

A logoff signaling message is sent from a client to a backend service. When you create the API operation for logoff signaling, note that the Protocol parameter must be set to WEBSOCKET.

#### 3.3 Generate and download an SDK

After you create the three API operations, authorize specific applications to call the API operations. Then, publish the API operations to the production environment. After the API operations are published to the production environment, choose Publish APIs > Owned APIs SDK in the left-side navigation pane. On the Owned APIs SDK Auto-Generation page, you can generate and download the SDK of the API group to which the three API operations belong.

You can download SDK for Android and install it on a client. Then, the client can use this SDK to establish a WebSocket connection with API Gateway, send a registration signaling message, and receive server-to-client notifications from the backend service. In addition, you can download SDK for Java and install it on the backend service. The backend service uses this SDK to send notifications to the client. The two SDKs work together to implement bidirectional communication. The following figure shows the Owned APIs SDK Auto-Generation page on which you can download SDKs.

ApiGateway	Owned APIs SDK Auto-Generation						
Overview	Enter the GroupName to query		Search	🌑 Tags			
Instances	Group Name	Tag	Description		Created Time	SDK Document auto generate	
▼ Publish APIs		۲			Nov 30,2020 21:08:17	Objective-C Android Java GOlang TypeScript	
API Groups	termination.	۲			Nov 30,2020 20:36:39	Objective-C Android Java GOlang TypeScript	
APIs	19 <b>1</b>	۲			Nov 30,2020 14:40:13	Objective-C Android Java GOlang TypeScript	
Plugin	and and address of	۲			Nov 27,2020 17:56:55	Objective-C Android Java GOlang TypeScript	
VPC Access		۲			Nov 27,2020 17:15:23	Objective-C Android Java GOlang TypeScript	
Log Manage		۲	Create an API operation w		Nov 27,2020 13:52:37	Objective-C Android Java GOlang TypeScript	
Owned APIs SDK		۲			Nov 25,2020 14:42:00	Objective-C Android Java GOlang TypeScript	
Consume APIs		۲			Aug 03,2020 16:02:49	Objective-C Android Java GOlang TypeScript	
Documentation		۲			Jul 29,2020 17:39:03	Objective-C Android Java GOlang TypeScript	
	The state	۲			Jul 17,2020 14:46:59	Objective-C Android Java GOlang TypeScript	

## 3.4 Use the SDK on a client to send a registration signaling message and receive server-to-client notifications

After you download SDK for Android, you must carefully read the ReadMe.txt file and install and use the SDK as instructed.

The SDK is automatically generated by API Gateway and contains all the API operations in the API group. You can find the API operation for registration signaling and call the API operation to send a registration signaling message from a client. The following code snippet is an SDK call example. In this example, a WebSocket channel is established and initialized between the client and API Gateway when a client is started. A function named onNotify is created to notify the client of sending server-to-client notifications. When the backend service sends a notification to the client, the SDK calls the onNotify function to notify the client. Another function named registerWebsocketTest is also provided for external calls.

```
public class Demo_HangZhou {
    static{
        WebSocketClientBuilderParams websocketParam = new WebSocketClientBuilderParams();
        websocketParam.setAppKey("12345678");
        websocketParam.setAppSecret("12345678");
        websocketParam.setApiWebSocketListner(new ApiWebSocketListner() {
            @Override
            // When the backend service sends a notification to the client, the SDK calls t
        he onNotify function to notify the client.
            public void onNotify(String message) {
                System.out.println(message);
            }
        }
    }
}
```

```
ROverride
            public void onFailure(Throwable t, ApiResponse response) {
                if(null ! = t) {
                    t.printStackTrace();
                }
                if(null ! = response) {
                    System.out.println(response.getCode());
                    System.out.println(response.getMessage());
                }
            }
        });
        WebSocketApiClient hangzhou.getInstance().init(websocketParam);
    }
   public static void registerWebsocketTest() {
        WebSocketApiClient hangzhou.getInstance().register("fred", "123456", new ApiCallb
ack() {
            @Override
            public void onFailure(ApiRequest request, Exception e) {
                e.printStackTrace();
            }
            @Override
            public void onResponse(ApiRequest request, ApiResponse response) {
                try {
                    System.out.println(getResultString(response));
                }catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        });
    }
}
```

#### 3.5 Use the SDK on a backend service to send notifications to a client

After a backend service receives an HTTP request that is converted from a registration signaling message, the backend service records the device ID of the client that is included in the request. To send notifications to the client, the backend service only needs to send a request to API Gateway to call the API operation for server-to-client notification signaling. The following sample code demonstrates how to call the API operation:

```
public class Demo Hanzhou {
            static{
                HttpClientBuilderParams param = new HttpClientBuilderParams();
                param.setAppKey("123456");
                param.setAppSecret("123456");
                HttpApiClient BeiJing.getInstance().init(param);
            }
            public static void HanZhouNotifyTest() {
                HttpApiClient HanZhou.getInstance().notify("NotifyContent" , new ApiCallbac
k() {
                    @Override
                    public void onFailure(ApiRequest request, Exception e) {
                        e.printStackTrace();
                    }
                    QOverride
                    public void onResponse(ApiRequest request, ApiResponse response) {
                        try {
                            System.out.println(response.getCode());
                        }catch (Exception ex) {
                            ex.printStackTrace();
                        }
                    }
                });
            }
}
```

The implementation of bidirectional communication between a client and a backend service by using API Gateway includes the following steps:

- 1. Create an API group, bind an independent domain name to the API group, and then enable the WebSocket channel feature for the independent domain name.
- 2. Create three API operations, one for registration signaling, one for server-to-client notification signaling, and one for logoff signaling. Then, authorize applications and publish the API operations to the production environment.
- 3. Configure the registration logic and logoff logic for the backend service. Then, download SDK for Java in the API Gateway console and install the SDK on the backend service. The backend service uses SDK for Java to send notifications to the client.
- 4. Download SDK for Android in the API Gateway console and install the SDK on the client. When the client is started, a WebSocket connection is established between the client and API Gateway. The client sends a registration signaling message by calling the API operation that corresponds to the signaling. After the registration is successful, the client can receive server-to-client notifications.

For information about how an SDK implements bidirectional communication, see Understand how an SDK implements bidirectional communication.

The following flowchart, which is originally provided in the linked topic, helps you better understand how bidirectional communication works.

### Create API-Implement bidirectional communication



If you encounter difficulties when you configure bidirectional communication in API Gateway, search for group ID 11747055 in DingTalk and join the group to seek help. The DingTalk group name is API Gateway-Customer Service.