# Alibaba Cloud

DataWorks DataService Studio

Document Version: 20220705

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	
A 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.	
다) 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.	
⑦ 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	06
2.Prepare the resource environment	80
2.1. Plan and configure resource groups	80
2.2. Configure network connectivity and whitelists	09
3.Create and manage business processes	12
4.API	16
4.1. CreateApi	16
4.1.1. Generate API	16
4.1.1.1. Overview	16
4.1.1.2. Configure a data source	16
4.1.1.3. Create an API in the codeless UI	20
4.1.1.4. Create an API in the code editor	30
4.1.1.5. Sample code for the advanced SQL syntax used for	39
4.1.2. Register an API	40
4.2. Test an API	44
4.3. Publish an API	45
4.4. Authorize access to an API	46
4.5. View and use APIs	49
4.5.1. Call an API	49
5.Function	51
5.1. Create and manage functions	51
5.2. Test a function	54
5.3. Publish a function	55
5.4. Use filters	56
5.4.1. Create an Aviator function and use the Aviator functio	56
5.4.2. Create and use Python functions as a filter	60

5.4.3. Best practices of using Aviator functions as filters	63
6.Service Orchestration	73
6.1. Use service orchestration	73
6.2. Manage workflows	80
7.Appendix: DataService Studio error codes	83

# 1.0verview

DataService Studio allows you to create and register APIs. You can publish APIs that are created or registered in DataService Studio to API Gateway and sell them in Alibaba Cloud Marketplace. This topic provides an overview on the features that DataService Studio provides.

DataService Studio aims to build a data service bus to help enterprises centrally manage private and public APIs. You can create APIs in DataService Studio based on tables or register existing APIs for centralized management and distribution.

DataService Studio is integrated with API Gateway to allow you to publish APIs to API Gateway with one click. DataService Studio works together with API Gateway to provide a secure, stable, cost-effective, and easy-to-use data sharing service.

DataService Studio adopts a serverless architecture. All you need to care is the query logic of APIs, instead of the infrastructure such as the runtime environment. DataService Studio prepares the computing resources for you, supports elastic scaling, and requires zero O&M cost.

#### **Create APIs**

DataService Studio allows you to create APIs in the codeless UI based on tables in relational databases and NoSQL databases. The codeless UI allows you to create APIs without writing code. For more information, see Create an API in the codeless UI.

To meet the personalized query requirements of advanced users, DataService Studio provides a code editor for customizing SQL statements of APIs. The code editor allows you to use table join queries, complex queries, and aggregate functions. For more information, see Create an API in the code editor.

#### **Register APIs**

DataService Studio allows you to register existing APIs so that you can manage existing APIs and the APIs that are created based on tables in a centralized manner. For more information, see Register an API.

#### Publish APIs to API Gateway

API Gateway provides API lifecycle management services, including API publishing, management, maintenance, and monetization. It provides a simple, fast, cost-effective, and low-risk service for you to aggregate microservices, separate the frontend from the backend, integrate systems, and provide features and data to partners and developers. For more information, see What is API Gateway?

You can publish APIs that are created or registered in DataService Studio to API Gateway, on which you can perform a variety of management operations. For example, you can manage permissions, configure bandwidth throttling, configure access control, and view statistics about APIs. After you create or register APIs in DataService Studio, you must publish the APIs to API Gateway before the APIs can be called by other users or in your own applications. DataService Studio is integrated with API Gateway to allow you to publish APIs to API Gateway with one click.

#### Sell APIs in Alibaba Cloud Marketplace

The API sector of Alibaba Cloud Market place provides thousands of API products in the following categories: finance, artificial intelligence (AI), e-commerce, transportation and geography, living services, corporate management, and public affairs. Alibaba Cloud Market place provides a platform where you can monetize your data.

After you publish APIs that are created or registered in DataService Studio to API Gateway, you can publish them to Alibaba Cloud Market place with one click. This is an easy way to achieve financial gains for your company.

# 2.Prepare the resource environment2.1. Plan and configure resource groups

When users use the DataService Studio service of DataWorks to call data APIs, the API calls are processed by resource groups. This topic describes the definition, types, connectivity, and performance of resource groups. This topic also compares resource groups of different types to help you select the resource groups that best suit your needs.

#### Definition

A resource group provides the computing resources that are needed to make API calls in the DataService Studio of DataWorks. In most cases, a resource group consists of one or more servers that provide CPU, memory, and network resources. Process of API calls: A user initiates an API call. API Gateway receives the request and forwards the request to a server of the DataService Studio service. The request is then forwarded to the destination data source for data query.

#### Types

Shared resource groups

Shared resource groups are shared by all the users of DataWorks. During peak hours, some users may preempt resources. For more information about shared resource groups, see Shared resource groups and resource plans for shared resource groups. For more information about how to use shared resource groups, see Use the default resource group.

#### Key to resource planning: connectivity and performance

When you use resource groups, you must pay attention to their connectivity and performance.

• Connectivity

After an API call is initiated, the API call is first sent to a server of the DataService Studio service and then to the destination data source for data query. Make sure that the resource group that is used to process the API call can access the destination data source and the network on which the data source resides. Before you use DataService Studio, make sure that the resource group is connected to the data source. Otherwise, the API call fails.

• Performance

API call nodes consume CPU, memory, and network resources of the servers on which the nodes are run. Insufficient resources may lead to various issues. For example, an exception may occur during an API call, throttling may be imposed on frequent API calls, or query results may not be returned in time. Therefore, you must provide sufficient resources for API calls.

#### Instructions on resource group configuration

If you use a shared resource group, add the CIDR blocks that are used by Alibaba Cloud services in the specified region to a whitelist of the data source. For more information, see Configure network connectivity between the shared resource group for DataService Studio and a data source.

# 2.2. Configure network connectivity and whitelists

To generate APIs for a data source in DataService Studio, you must configure network connectivity to ensure that the resource group used by DataService Studio can access the data source over the specified network. You must also add CIDR blocks used by the resource group to a whitelist of the data source. This topic describes how to configure network connectivity and whitelists for data sources that are deployed on different types of networks.

# Configure network connectivity between the shared resource group for DataService Studio and a data source

Before you generate APIs for a data source in the DataService Studio, you must configure the data source. To ensure that the shared resource group for DataService Studio can access the data source, add the CIDR blocks used by Alibaba Cloud services in each region to a whitelist of the data source.

Region	CIDR block
China (Hangzhou)	100.64.0.0/10,11.193.102.0/24,11.193.215.0/24,11.194.110.0/24,11.194.73.0/24,118.31 .157.0/24,47.97.53.0/24,11.196.23.0/24,47.99.12.0/24,47.99.13.0/24,114.55.197.0/24,1 1.197.246.0/24,11.197.247.0/24,118.31.243.0/26,118.31.243.64/26,118.31.243.128/26, 118.31.243.192/26,11.193.55.0/24,101.37.74.122,114.55.197.231,114.55.198.83,101.37. 74.206
China (Shanghai)	11.193.109.0/24,11.193.252.0/24,47.101.107.0/24,47.100.129.0/24,106.15.14.0/24,10.1 17.28.203,10.143.32.0/24,10.152.69.0/24,10.153.136.0/24,10.27.63.15,10.27.63.38,10.2 7.63.41,10.27.63.60,10.46.64.81,10.46.67.156,11.192.97.0/24,11.192.98.0/24,11.193.10 2.0/24,11.218.89.0/24,11.218.96.0/24,11.219.217.0/24,11.219.218.0/24,11.219.219.0/2 4,11.219.233.0/24,11.219.234.0/24,118.178.142.154,118.178.56.228,118.178.59.233,11 8.178.84.74,120.27.160.26,120.27.160.81,121.43.110.160,121.43.112.137,100.64.0.0/10, 10.117.39.238,11.193.96.0/24,11.193.48.0/24,11.193.108.0/24,101.132.31.146,106.15.1 4.240,106.15.14.75,101.132.31.221
China (Shenzhen)	100.106.46.0/24,100.106.49.0/24,10.152.27.0/24,10.152.28.0/24,11.192.91.0/24,11.192 .96.0/24,11.193.103.0/24,100.64.0.0/10,120.76.104.0/24,120.76.91.0/24,120.78.45.0/2 4,47.106.63.0/26,47.106.63.128/26,47.106.63.192/26,47.106.63.64/26,11.193.94.0/24,1 20.78.45.154,120.78.46.137,120.78.46.107,120.78.45.140
China (Chengdu)	11.195.52.0/24,11.195.55.0/24,47.108.22.0/24,100.64.0.0/10
China (Beijing)	100.106.48.0/24,10.152.167.0/24,10.152.168.0/24,11.193.50.0/24,11.193.75.0/24,11.19 3.82.0/24,11.193.99.0/24,100.64.0.0/10,47.93.110.0/24,47.94.185.0/24,47.95.63.0/24,1 1.197.231.0/24,11.195.172.0/24,47.94.49.0/24,182.92.144.0/24,11.193.100.0/24,11.193 .199.0/24,39.106.244.50,47.95.63.101,47.95.63.93,39.106.244.48
China (Zhangjiakou)	11.193.235.0/24,47.92.22.0/24,100.64.0.0/10,11.112.227.0/24
China (Hong Kong)	10.152.162.0/24,11.192.196.0/24,11.193.11.0/24,100.64.0.0/10,47.89.61.0/24,47.91.17 1.0/24,11.193.118.0/24,47.75.228.0/24,47.56.45.0/25,47.244.92.128/25,47.101.109.0/2 4,11.193.200.0/24,11.193.12.0/24,47.90.71.152,47.90.71.141,47.91.171.178,47.91.172.3

Region	CIDR block		
Singapore (Singapore)	100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11. 192.153.0/24,11.192.40.0/24,11.193.8.0/24,100.64.0.0/10,47.88.147.0/24,47.88.235.0/ 24,11.193.162.0/24,11.193.163.0/24,11.193.220.0/24,11.193.158.0/24,47.74.162.0/24,4 7.74.203.0/24,47.74.161.0/24,11.197.188.0/24,11.197.227.0/24,47.74.161.218,47.74.16 1.181,161.117.140.83,47.88.143.36		
Australia (Sydney)	11.192.100.0/24,11.192.134.0/24,11.192.135.0/24,11.192.184.0/24,11.192.99.0/24,100. 64.0.0/10,47.91.49.0/24,47.91.50.0/24,11.193.165.0/24,47.91.60.0/24,11.195.113.0/24, 47.74.100.0/24		
US (Silicon Valley)	10.152.160.0/24,100.64.0.0/10,47.89.224.0/24,11.193.216.0/24,47.88.108.0/24,47.88.9 9.153,47.254.58.215,47.88.108.192,47.254.58.135		
US (Virginia)	11.193.203.0/24,11.194.68.0/24,11.194.69.0/24,100.64.0.0/10,47.252.55.0/24,47.252.8 8.0/24,11.194.69.0/24,10.128.135.0/24,47.88.98.0/24		
Malaysia (Kuala Lumpur)	11.193.188.0/24,11.221.205.0/24,11.221.206.0/24,11.221.207.0/24,100.64.0.0/10,11.21 4.81.0/24,47.254.212.0/24,11.193.189.0/24,47.250.29.0/26,47.250.29.128/26,47.250.29 .192/26,47.250.29.64/26		
Germany (Frankfurt)	11.192.116.0/24,11.192.168.0/24,11.192.169.0/24,11.192.170.0/24,11.193.106.0/24,10 0.64.0.0/10,11.192.116.14,11.192.116.142,11.192.116.160,11.192.116.75,11.192.170.27, 47.91.82.22,47.91.83.74,47.91.83.93,47.91.84.11,47.91.84.110,47.91.84.82,11.193.167.0 /24,47.254.138.0/24,11.194.61.0/24,47.254.185.0/24		
Japan (Tokyo)	100.105.55.0/24,11.192.147.0/24,11.192.148.0/24,11.192.149.0/24,100.64.0.0/10,47.91 .12.0/24,47.91.13.0/24,47.91.9.0/24,11.199.250.0/24,47.91.27.0/24,11.59.59.0/24,47.2 45.51.128/26,47.245.51.192/26,47.91.0.128/26,47.91.0.192/26		
India (Mumbai)	11.194.10.0/24,11.246.70.0/24,11.246.71.0/24,11.246.73.0/24,11.246.74.0/24,100.64.0. 0/10,149.129.164.0/24,11.194.11.0/24,11.59.62.0/24,147.139.23.0/26,147.139.23.128/ 26,147.139.23.64/26,149.129.165.192/26		
UK (London)	11.199.93.0/24,100.64.0.0/10,8.208.72.0/26,8.208.72.128/26,8.208.72.192/26,8.208.72. 64/26		
Indonesia (Jakarta)	11.194.49.0/24,11.200.93.0/24,11.200.95.0/24,11.200.97.0/24,100.64.0.0/10,149.129.2 28.0/24,10.143.32.0/24,11.194.50.0/24,11.59.135.0/24,147.139.156.0/26,147.139.156.1 28/26,147.139.156.64/26,149.129.230.192/26,149.129.229.0/26,149.129.229.64/26,149 .129.229.128/26,149.129.229.192/26		

#### Notes about whitelist configuration for data sources

To prevent that a data source is inaccessible to DataService Studio due to the whitelist configuration of the data source, you must add the IP addresses of the resource groups for DataService Studio to a whitelist of the data source. This section describes the notes about whitelist configuration for data sources.

In this example, an ApsaraDB RDS instance is used as a data source. ApsaraDB RDS supports standard IP address whitelists and enhanced IP address whitelists. The type of the whitelist that you configure may affect the network connectivity between DataService Studio and your ApsaraDB RDS instance.

• If you configure a standard IP address whitelist, take note of the following items:

- A standard IP address whitelist does not distinguish between the classic network and VPCs.
- You can add the IP addresses of the shared resource group and exclusive resource groups for DataService Studio to the same standard IP address whitelist.

**Note** The IP addresses in a standard IP address whitelist are granted access to your ApsaraDB RDS instance over both the classic network and VPCs.

- If you configure an enhanced IP address whitelist, take note that of the following items:
  - An enhanced IP address whitelist distinguishes between the classic network and VPCs.

**Note** You must specify the network isolation mode of each enhanced IP address whitelist. For example, if the Network Isolation Mode parameter is set to Classic Network for an IP address whitelist, the IP addresses in the IP address whitelist are granted access to your ApsaraDB RDS instance only over the classic network. In this case, you cannot connect to your ApsaraDB RDS instance over VPCs from these IP addresses.

- To allow the shared resource group for DataService Studio to access your ApsaraDB RDS instance over a VPC, add the IP addresses of the shared resource group to an IP address whitelist for which the network isolation mode is set to VPC. For example, you can allow DataService Studio to access an ApsaraDB RDS for MySQL instance that is deployed in a VPC in this way.
- To allow resource groups for DataService Studio to access your ApsaraDB RDS instance over the Internet or the classic network, add the IP addresses of the resource groups to an IP address whitelist for which the network isolation mode is set to Classic Network.
- If you change a standard IP address whitelist to an enhanced IP address whitelist in your ApsaraDB RDS instance, take note of the following item:

The standard IP address whitelist is replicated into two enhanced IP address whitelists that contain the same IP addresses. The two enhanced IP address whitelists have different network isolation modes.

Other notes about whitelist configuration:

- When you configure IP address whitelists, the workloads on your RDS instance are not interrupted.
- The IP address whitelist labeled default can be cleared, but cannot be deleted.
- Do not modify or delete the IP address whitelists that are generated by other Alibaba Cloud services. If you delete these IP address whitelists, the related Alibaba Cloud services cannot connect to your ApsaraDB RDS instance. These whitelists include **ali\_dms\_group** that is created by **Database** Management (DMS) and **hdm\_security\_ips** that is created by Database Autonomy Service (DAS).

(?) Note We recommend that you create a separate IP address whitelist for DataWorks in your ApsaraDB RDS instance.

• The IP address whitelist labeled default contains only the 127.0.0.1 IP address. This indicates that no IP addresses can access your ApsaraDB RDS instance.

For more information about how to configure a whitelist in your ApsaraDB RDS instance, see Use a database client or the CLI to connect to an ApsaraDB RDS for MySQL instance. You can use a similar method to configure a whitelist for another type of data source. To configure whitelists for other types of data sources, see the related instructions.

# 3.Create and manage business processes

This topic describes how to create, modify, and delete a business process.

#### Context

DataWorks allows you to organize different types of resources in a business process. This helps you analyze data by business. Each business process contains APIs, functions, and workflows.

#### Create a business process

- 1.
- 2. In the Service Development pane, move the pointer over the price icon and select New business process.
- 3. In the Create Workflow dialog box, set the parameters.

New	business pro	cess		×
	second-level limited to 10 to remove th	are the units that API Gateway manages. The number of I domain name calls automatically assigned by the group is 100 times per day. You can bind an independent domain name he call restriction. er a business process belong to the API Group specified by the pocess.		
	Business Name : * API Group :	The business name must be unique. It can contain 4 to 50 characters, incl Chinese characters, English letters, numbers, and underscores in English format. It must start with an English letter or Chinese character. Please Select To create a new group, you can jumpAPI GatewayCreate new grouping	0/50 luding	
Busin	ness Description :	Business Description, no more than 180 characters	0/180	
		ок с	ancel	

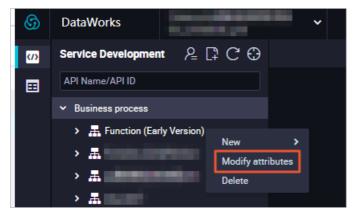
Parameter	Description		
Business Name	<ul> <li>The name of the business process.</li> <li>The name must be unique in the workspace to which the business process belongs.</li> <li>The name can contain letters, digits, and underscores (_).</li> <li>The name must start with a letter.</li> <li>The name must be 4 to 50 characters in length.</li> </ul>		
API Group	The API group to which the APIs in the business process belong. An API group is the unit of API management in API Gateway. Each API group maps an API product in Alibaba Cloud Marketplace. You can create an API group in the API Gateway console.		
Business Description	The description of the business process, which can be up to 180 characters in length.		

#### 4. Click OK.

After the business process is created, you can view it in the business process list.

#### Modify a business process

1. In the Service Development panel, right-click the business process that you want to modify and select Modify attributes.



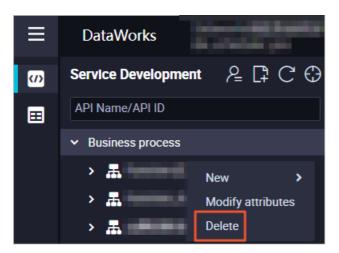
2. In the Edit business process dialog box, modify the Business Name and Business Description parameters as required.

Edit business process	×
<ul> <li>API groups are the units that API Gateway manages. The number of second-level domain name calls automatically assigned by the group is limited to 1000 times per day. You can bind an independent domain name to remove the call restriction.</li> <li>All APIs under a business process belong to the API Group specified by the business process.</li> </ul>	
* Business Name : 11/50 The business name must be unique. It can contain 4 to 50 characters, includin Chinese characters, English letters, numbers, and underscores in English format. It must start with an English letter or Chinese character.	_
* API Group : To create a new group, you can jumpAPI GatewayCreate new grouping	
Business Description :	
Business Description, no more than 180 characters	
ОК Сапсе	I
⑦ Note You cannot modify the Creator or API Group parameter of a business process	s.

3. Click OK.

#### Delete a business process

1. In the Service Development panel, right-click the business process that you want to delete and select Delete.



2. In the Notes message, click OK.

#### ? Note

- You can delete only business processes that do not contain objects such as folders, APIs, functions, or workflows.
- If you want to delete a business process that contains such objects, delete the objects before you delete the business process.

## **4.API 4.1. CreateApi** 4.1.1. Generate API

#### 4.1.1.1. Overview

This topic describes the two modes for creating APIs and the differences between the modes.

DataService Studio allows you to create APIs on the codeless user interface (UI) based on tables in relational databases and NoSQL databases. The codeless UI allows you to create APIs without writing code. For more information, see Create an API in the codeless UI.

To meet the personalized query requirements of advanced users, DataService Studio provides a code editor for you to customize the SQL statements of APIs. The code editor allows you to use table join queries, complex queries, and aggregate functions. For more information, see Create an API in the code editor.

Category	Feature	Codeless UI	Code editor
Query object	Query a single table in a single data source	Supported	Supported
	Query joined tables in a single data source	Not supported	Supported
	Query an exact number	Supported	Supported
Query condition	Query a range of numbers	Supported	Supported
	Query a string in exact match mode	Supported	Supported
	Query strings in fuzzy match mode	Supported	Supported
Query result	Return field values without changes	Supported	Supported
	Return a mathematical calculation of field values	Not supported	Supported
	Return an aggregate calculation of field values	Not supported	Supported
	Return results on multiple pages	Supported	Supported

The following table describes the differences between the codeless UI and code editor.

#### 4.1.1.2. Configure a data source

You can configure data sources for DataService Studio to read table schemas and process API query requests.

#### Prerequisites

To ensure that resource groups for DataService Studio can access data sources, you must configure network connectivity between the resource groups and data sources and configure IP address whitelists of the data sources. Otherwise, DataService Studio cannot work as expected.

- For more information about how to configure network connectivity for a shared resource group, see Configure network connectivity between the shared resource group for DataService Studio and a data source.
- For more information about how to configure network connectivity for an exclusive resource group, see Configure network connectivity between an exclusive resource group for DataService Studio and a data source.

#### Procedure

- 1. Log on to the DataWorks console.
- 2. In the left-side navigation pane, click **Workspaces**.
- 3. Select the region in which the workspace to be managed resides. Then, find the workspace and click **Workspace Settings** in the Actions column. In the panel that appears, click More.
- 4. In the left-side navigation pane, click **Data Source**.

Add a data source on the page that appears. The following table describes the supported data sources and the configuration methods.

Data source	Generate an API in the codeless UI	Create an API in the code editor	Configuration method
MaxCompute	Supported	Supported	Add a MaxCompute data source <b>Note</b> DataService Studio APIs use the MaxCompute Query Acceleration (MCQA) feature to query data from MaxCompute data sources. For more information about the scenarios and limits of this feature, see MaxCompute Query Acceleration. If the prerequisites of using the MCQA feature are not met, the calls of the DataService Studio APIs fail.
ClickHouse	Supported	Supported	Add a ClickHouse data source
ApsaraDB RDS	Supported	Supported	Supported RDS engine types are MySQL, PostgreSQL, and SQL Server.

#### Dat aWorks

Data source	Generate an API in the codeless UI	Create an API in the code editor	Configuration method
	Supported	Supported	Add a DRDS data source
Distributed Relational Database Service (DRDS)			<b>Note</b> You can add this data source only by using a connection string.
MySQL	Supported	Supported	Add a MySQL data source
PostgreSQL	Supported	Supported	Add a PostgreSQL data source
SQL Server	Supported	Supported	Add an SQL Server data source
		Supported	Add an Oracle data source
Oracle	Supported		<b>Note</b> You can add this data source only by using a connection string.
AnalyticDB for MySQL V2.0	Supported	Supported	Add an AnalyticDB for MySQL 2.0 data source
AnalyticDB for MySQL 3.0	Supported	Supported	Configure an AnalyticDB for MySQL 3.0 connection
AnalyticDB for PostgreSQL	Supported	Supported	Add an AnalyticDB for PostgreSQL data source
			Add a Tablestore data source
Tablestore	Supported	Not supported	<b>Note</b> You can add this data source only by using a connection string.
MongoDB	Supported	Not supported	Add a MongoDB data source

5. In the dialog box for adding the data source, set the Resource Group connectivity parameter to Data Service and use a resource group for DataService Studio to test network connectivity. In the example shown in the following figure, a MySQL data source is added.

dd MySQL data source					
* Default Database : Name				?	
* User name :					
* Password :					
Read Secondary : 🔵 Database First	0				
Resource Group : Dat connectivity	ta Integration Data Service	Schedule			
can be connected.		essary to ensure that the public reso			
Resource Group Name	Туре	(Click status to view details)	Test time	Actions	
公共	资源组	Not Tested		Test connectivity	
, - ,	fail due to the following possible arted. Make sure that the data s	e causes. Troubleshoot the failures	as instructed.		
	ss the network where the data so	ource resides. Make sure that the ne	etwork where the data	a source resides is	
<ol> <li>DataWorks is prohibited to blocks used by DataWork</li> </ol>		e data source resides by a network t	firewall. Add the IP ac	dresses or CIDR	
<ol> <li>The domain name of the resolved.</li> </ol>	data source cannot be resolved	. Make sure that the domain name	of the data source ca	n be properly	

**Note** If the network connectivity test fails, check your configurations. For more information, see **Configure network connectivity and whitelists**.

6. After the data source passes the network connectivity test, return to DataService Studio to create an API for the data source.

#### 4.1.1.3. Create an API in the codeless UI

DataWorks allows you to create APIs by setting parameters in the codeless user interface (UI) without the need to write code. This topic describes how to create an API in the codeless UI.

#### Prerequisites

A data source is configured on the **Data Source** page. For more information, see **Configure a data** source.

#### Create an API

- 1. Go to the DataService Studio page.
  - i. Log on to the DataWorks console.
  - ii. In the left-side navigation pane, click **Workspaces**.
  - iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **DataService Studio** in the Actions column.
- 2. In the Service Development pane, move the pointer over the icon and choose Create API > Generate API.

You can also click a business process, right-click API, and then choose Create API > Generate API.

3. In the Generate API dialog box, set the parameters that are described in the following table.

Generate API	×	<
* API mode :	Wizard Mode     Script Mode	
* API Name :	0/50	
	An API name must start with a letter or Chinese character, and can contain numbers, letters, Chinese characters, and underscores (_). The name must be 4 to 50 characters in length.	
* API Path :	0/200	
	A path must start with a forward slash (/), and can contain letters, numbers, underscores (_), and hyphens (·), for example, /user. The path can be up to 200 characters in length.	
* Protocol :	V HTTP V HTTPS	
* Request Method :	GET V	
* Response Content	JSON	
Туре :		
* Visible Range :	Work Space 🗸 💿	
Label :	Please Select V	
	Up to 0-5 tags can be set up, Chinese characters, English, numbers, underline, and no more than 20 characters per label	
* Description :		
	0,7000	
	OK Cancel	
Parameter	Description	
API Mode	The mode used to create the API. Valid values: <b>Wizard Mode</b> and <b>Script Mode</b> . In this example, select <b>Wizard Mode</b> .	
API Name	The name of the API. The name must be 4 to 50 characters in length, and can contain letters, digits, and underscores (_). The name must start with a letter.	

Parameter	Description
	The path of the API. The path is the API request path that is relative to the service host address, such as <i>/user</i> .
API Path	<b>Note</b> The path can be up to 200 characters in length and can contain letters, digits, underscores (_), and hyphens (-). The path must start with a forward slash (/).
	The protocol used by the API. Valid values: <b>HTTP</b> and <b>HTTPS</b> . If you want to call the API by using HTTPS, you must bind an
Protocol	independent domain name to the API in the API Gateway console after the API is published to API Gateway. You must also upload an SSL certificate in the API Gateway console. For more information, see Enable HTTPS for an API operation.
	The request method. Valid values: GET and POST
Request Method	Note If you select GET for the Request Method parameter, you can select only QUERY for the Parameter Position parameter. If you select POST for the Request Method parameter, you can select QUERY or BODY for the Parameter Position parameter.
Response Content Type	The format of the data returned by the API. Set the value to <b>JSON</b> .
	The range of users to whom the API is visible. Valid values:
	• <b>Work Space</b> : The API is visible to all members in the current workspace.
Visible Range	• <b>Private</b> : The API is visible only to its owner and permissions on the API cannot be granted to other members.
	<b>Note</b> If you set this parameter to Private, other members in the workspace cannot view the API in the API list.
	Select tags from the <b>Label</b> drop-down list. For more information, see Create and manage API tags.
Label	<b>Note</b> A tag can be up to 20 characters in length and can contain letters, digits, and underscores (_). You can set at most five tags for an API.

Parameter	Description
Description	The description of the API. The description can be up to 2,000 characters in length.
Destination Folder	The folder in which you want to store the API. You can select a business process from the drop-down list. After you select a business process, a folder is generated under the business process and the API is stored in the folder. The default folder is in the format of Business Processes/Business process name/API, such as <i>Business Processes/cesh i/API</i> .

#### 4. Click Determine.

#### Configure the API

1. Double-click the API in the API list. On the tab that appears, set the **Datasource Type**, **Datasource Name**, and **Table Name** parameters in the **Select Table** section.

Select Table		
* Datasource Type :	MySQL	
* Datasource Name :	· · · · ·	0
* Table Name :		

#### ? Note

- Before you select a table for an API, you must configure a data source in Data Integration. You can enter a table name in the Table Name field to search for the table that you want to use.
- After you create an API, the API configuration tab appears. Then, you can select a table for the API.

#### 2. In the Environment Configuration section, set the Memory and Function Timeout parameters.

Environment Configuration				
* Memory :	4096M		~	
Function Timeout :	10000	ms		

**?** Note We recommend that you set the Function Timeout parameter to a value equal to or less than 10000 for a shared resource group and 30000 for an exclusive resource group for DataService Studio.

3. In the Select Parameters section, set the request and response parameters for the API.

After you select a table in the Select Table section, all fields in the table appear in the **Select Parameters** section. In the **Set as Req Param** column, select the check boxes of the fields that you want to specify as request parameters. In the **Set as Resp Param** column, select the check boxes of the fields that you want to specify as response parameters.

Select Parameters					
Search the field name	Q				
😑 Set as Req Param	🖌 Set as Resp Param	Field Name	Field Type	Field Description	Field sorting
	✓	id	INT		
	✓	name	VARCHAR		

To sort the data that the API returns based on a specific field, find the field and click **Add** in the Add to field sort column to add the field to the **Sort field** section.

Sort field			
The sort field is not required	; If you need to sort the fields,	first select the desired field in the list of selection par	ameters
Serial Number	Field Name	Sort	Actions
1	id	Ascending order \vee	Move up   Move down   Delete

The sorting feature allows you to specify the fields based on which the data that the API returns is sorted. A field with a smaller sequence number in the Sort field section has a higher priority in sorting. You can click **Move up** or **Move down** to adjust the sequence of a field. You can specify the sorting mode for each field by selecting **Ascending order** or **Descending order** in the Sort column.

4. In the right-side navigation pane, click the **Request Param** tab. In the Request Param panel, set the parameters as required.

#### ? Note

- Before you preview the result, you must configure the information such as the example value, default value, and description for each request parameter of the API.
- To improve the match efficiency, specify an indexed field as a request parameter.

×	Request Param						Properties
	Parameter Name						rties
	PageNum	INT	QUERY 🗸	Equal	Yes	1	Rec
	⑦ pageSize	INT	QUERY 🗸	Equal	Yes	10	Request F
							Param
							Res
							Response
							Param
							<mark>≥</mark> Filter

Parameter	Description
Parameter Name	The name of the request parameter. The name can be up to 64 characters in length, and can contain letters, digits, underscores (_), and hyphens (-). The name must start with a letter.
Bound Field	By default, the information cannot be changed.
Parameter Type	The type of the request parameter. Valid values: <b>STRING</b> , <b>INT</b> , <b>LONG</b> , <b>FLOAT</b> , <b>DOUBLE</b> , and <b>BOOLEAN</b> .
Parameter Position	<ul> <li>The position of the request parameter. Valid values: QUERY and BODY.</li> <li>Note If you set the Parameter Position parameter to BODY for one or more request parameters, you must set the Content-Type parameter to define the format of the request parameters in the request body.</li> <li>Valid values of the Content-Type parameter: <ul> <li>application/json; charset=utf-8: the JSON format.</li> <li>application/xml; charset=utf-8: the XML format.</li> <li>application/x-www-form-urlencoded; charset=utf-8: the FORM format.</li> </ul> </li> </ul>

Parameter	Description
Operator	<ul> <li>Description</li> <li>The operator that is used to associate or compare the value of the request parameter with the value that you specify. You can select one of the following operators:</li> <li>Equal: The value of the request parameter is equal to the specified value.</li> <li>LIKE: The value of the request parameter matches the specified pattern.</li> <li>IN: The value of the request parameter is in the specified range.</li> <li>NOT IN: The value of the request parameter does not match the specified pattern.</li> <li>IS: The value of the request parameter is not equal to the specified range.</li> <li>NOT LIKE: The value of the request parameter is not equal to the specified value.</li> <li>I =: The value of the request parameter is greater than the specified value.</li> <li>&gt; : The value of the request parameter is greater than or equal to the specified value.</li> <li>&lt;=: The value of the request parameter is greater than or equal to the specified value.</li> <li>&gt; =: The value of the request parameter is less than or equal to the specified value.</li> <li>&lt;=: The value of the request parameter is less than or equal to the specified value.</li> <li>&lt;=: The value of the request parameter is less than or equal to the specified value.</li> <li>&lt;=: The value of the request parameter is less than or equal to the specified value.</li> </ul>
Required	Specifies whether the request parameter is required.
Example Value	The example value of the request parameter.
Default Value	The default value of the request parameter.
Description	The description of the request parameter.

5. In the right-side navigation pane, click the **Response Param** tab. In the Response Param panel, set the parameters as required.

#### Dat aWorks

Parameter Name					
? pageNum		INT	1	分页默认参数:页编号	
<b>?</b> pageSize		INT	10	分页默认参数:页大小	
totalNum		INT		分页默认参数:总记录数	
amount	amount	INT 🗸			
area	area	STRING V			
vanced Settings					
- 4 ma	timum of 500 records can be	returned if you do not enab	le the pagination feature. V	We recommend that you enable this feature if	
	than 500 records are returned				

Parameter	Description
Parameter Name	The name of the response parameter. The name can be up to 64 characters in length, and can contain letters, digits, underscores (_), and hyphens (-). The name must start with a letter.
Bound Field	By default, the information cannot be changed.
Parameter Type	The type of the response parameter. Valid values: <b>STRING</b> , <b>INT</b> , <b>LONG</b> , <b>FLOAT</b> , <b>DOUBLE</b> , and <b>BOOLEAN</b> .
Example Value	The example value of the response parameter.
Description	The description of the response parameter.

#### You can select Pagination in the Advanced Settings section.

Select Pagination based on your needs, and take note of the following items:

- If you do not select Pagination, the API returns up to 2,000 records.
- If the API returns more than 2,000 records, we recommend that you select Pagination.
- The following common parameters are available when you select Pagination:
- Common request parameters
  - pageNum: the page number of the page to return.
  - pageSize: the number of entries to return on each page.
- Common response parameters
  - pageNum: the page number of the returned page.
  - pageSize: the number of entries returned on each page.
  - totalNum: the total number of returned entries.

**?** Note Request parameters are optional for an API. If you do not specify request parameters for an API, you must select **Pagination**.

#### 6. Configure a filter.

If you need to process the request parameters of an API or process the results returned by an API, you can configure a filter for the API by performing the following steps: In the right-side navigation pane, click Filter. In the Filter panel, select Use Pre-filter or Use Post-filter based on your needs. After you specify the Function Type parameter, select one or more functions from the drop-down list next to the Use Pre-filter or Use Post-filter check box. If you select multiple functions, the functions are used to process parameters based on the sequence in which they are selected. Then, you can click Preview Responses Returned by API Operation and check whether the processing results of the selected functions meet your expectations. For more information about how to create and use a filter, see Create an Aviator function and use the Aviator function as a filter and Create and use Python functions as a filter.

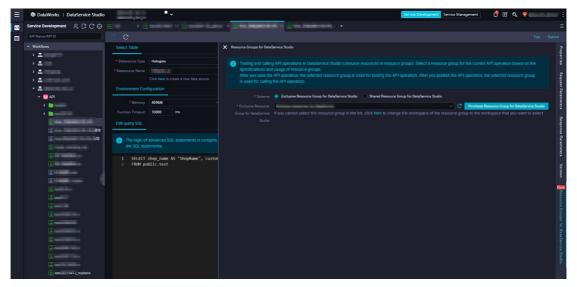
#### ? Note

- If you want to use a Python function as a filter, you must activate DataWorks
   Professional Edition or a more advanced edition and use the shared resource group for
   DataService Studio.
- If you want to use an Aviator function as a filter, all DataWorks editions are supported, but you must use an exclusive resource group for DataService Studio.
- If the function that you want to use is not available in the drop-down list, check whether the function is published. If the function is not published, publish the function. Alternatively, you can create and publish a new function. For more information, see Publish a function.

×	Filter					Prop
			sists of one function or multiple functions that are executed in sequence. classified into prefilters and post-filters. You can use prefilters and post-filters together or separately.			Properties
		If the desir	ed function is not displayed in the drop-down list of a filter, check whether the function is published, or create ction and publish it.			Request Param
FI	unction	Туре:				Param
			After you switch over to another option, the functions of the pre-filter and post-filter are cleared and the types of functions are c to the desired types.	hange		Res
	Use	Pre-filter			3	Response
			You can use a pre-fliter to preprocess request parameters of an API operation.			Param
	Use	Post-filter				New
			You can use a post-filter to perform secondary processing on query results of an API operation.			Ne Filter
						Version
	1					sion

7. Configure a resource group for DataService Studio.

i. In the right-side navigation pane, click the Resource Group tab. Select Exclusive Resource Group for DataService Studio or Shared Resource Group for DataService Studio for the Scheme parameter. If you select Exclusive Resource Group for DataService Studio, select an exclusive resource group from the drop-down list. If you select Shared Resource Group for DataService Studio, you do not need to select a shared resource group. DataWorks automatically manages the shared resource group.



(?) Note If the exclusive resource group that you want to use is not displayed in the drop-down list, log on to the DataWorks console and associate the exclusive resource group with your workspace.

8. Click the **m** icon in the toolbar. After the API is saved, the selected resource group takes effect

during the test.

After the API is configured, you can test it. For more information, see Test an API.

After the test succeeds, click **Submission** in the upper-right corner.

In the right-side navigation pane of the API configuration tab, click **Version**. Find the API version that you want to publish and click **Application for publication** in the Actions column to go to the application page. You can use the default application type and enter the reason for the application. Then, submit the application.

(?) Note If you have defined an approval policy in DataWorks Approval Center, the API must be approved through the approval process before the API can be published. For more information, see Overview.

After the API is published, the configuration of the resource group for DataService Studio takes effect when the API is called.

On the **Service Development** tab, you can find the registered API in the left-side navigation pane and manage the API. For example, you can clone or delete the API. On the **Service Management** tab, you can find the published API in the API list and view the details of the published API. For more information, see View, delete, move, and clone an API.

#### Switch from the codeless UI to the code editor

On the configuration tab of an API, you can switch from the codeless UI to the code editor.

- 1. In the Service Development pane, click the business process to which the required API belongs.
- 2. Double-click the required API. The configuration tab of the API appears.
- 3. Click the 📷 icon in the toolbar.
- 4. In the message that appears, click **OK**. Then, you can view the SQL statements of the API in the **Edit query SQL** section.

#### ♥ Notice

- DataService Studio allows you to switch only from the codeless UI to the code editor.
- After you switch from the codeless UI to the code editor, you cannot switch back to the codeless UI.

#### 4.1.1.4. Create an API in the code editor

To meet the personalized query requirements of advanced users, DataService Studio provides a code editor that you can use to customize the SQL statements of APIs. The code editor allows you to use table join queries, complex queries, and aggregate functions. This topic describes how to create an API in the code editor.

#### Prerequisites

A data source is configured on the **Data Source** page. For more information, see **Configure a data source**.

#### Create an API

1.

2. In the Service Development pane, move the pointer over the price icon and choose Create API >

#### Generate API.

You can also select a business process, right-click **API**, and then choose **Create API > Generate API**.

3. In the Generate API dialog box, set the parameters that are described in the following table.

Generate API		×
* API mode : ( * SQL Mode : (	<ul> <li>Wizard Mode</li> <li>Script Mode</li> <li>Basic SQL</li> <li>Advanced SQL</li> </ul>	
	0/50 An API name must start with a letter or Chinese character, and can contain numbers, letters, Chinese characters, and underscores (_). The name must be 4 to 50 characters in length.	
	0/200 A path must start with a forward slash (/), and can contain letters, numbers, underscores (_), and hyphens (-), for example, /user. The path can be up to 200 characters in length.	
	V HTTP V HTTPS	
* Request Method : * Response Content		
Type : * Visible Range :	Work Space 🗸 💿	
Label :	Please Select V Up to 0-5 tags can be set up, Chinese characters, English, numbers, underline (_), and no more than 20 characters per	
* Description :	label	
* Target Folder :	0/2000	
ruiger folder .	OK Cancel	

Parameter	Description
API Mode	The mode that is used to create the API. Valid values: <b>Wizard Mode</b> and <b>Script Mode</b> . In this example, select <b>Script Mode</b> .
SQL Mode	<ul> <li>The SQL mode. Valid values:</li> <li>Basic SQL: Use basic SQL statements to implement the query logic. This mode provides the SQL capability the same as that in earlier versions.</li> <li>Advanced SQL: Use SQL statements with MyBatis tags to implement the query logic. This mode supports the following tag types: if, choose, when, otherwise, trim, foreach, and where.</li> </ul>
API Name	The name of the API. The name must be 4 to 50 characters in length, and can contain letters, digits, and underscores (_). The name must start with a letter.
API Path	The path of the API, such as <i>/user</i> .

Parameter	Description
Protocol	The protocol used by the API. Valid values: <b>HTTP</b> and <b>HTTPS</b> . If you want to call the API by using HTTPS, you must bind an independent domain name to the API in the API Gateway console after the API is published to API Gateway. You must also upload an SSL certificate in the API Gateway console. For more information, see <b>Enable HTTPS for an API operation</b> .
Request Method	The request method. Valid values: GET and POST.   Note If you select GET for the Request Method parameter, you can select only QUERY for the Parameter Position parameter. If you select POST for the Request Method parameter, you can select QUERY or BODY for the Parameter Position parameter.
Response Content Type	The format of the data returned by the API. Set the value to <b>JSON</b> .
Visible Range	<ul> <li>The range of users to whom the API is visible. Valid values:</li> <li>Work Space: The API is visible to all members in the current workspace.</li> <li>Private: The API is visible only to its owner, and permissions on the API cannot be granted to other members.</li> <li>Note If you set this parameter to Private, other members in the workspace cannot view the API in the API list.</li> </ul>
Label	Select tags from the Label drop-down list. For more information, see Create and manage API tags.           ⑦ Note         A tag can be up to 20 characters in length, and can contain letters, digits, and underscores (_). You can add at most five tags to an API.
Description	The description of the API. The description can be up to 2,000 characters in length.
Destination Folder	The folder that stores the API.

#### 4. Click Determine.

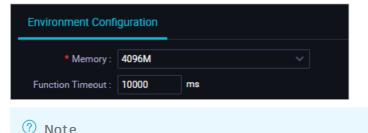
#### Configure the API

1. Double-click the API in the API list. On the tab that appears, set the **Datasource Type** and **Datasource Name** parameters in the **Select Table** section.

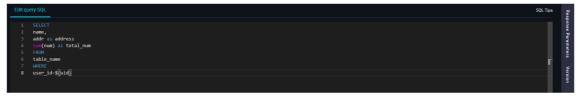
Select Table			
* Datasource Type :	MySQL	/	
* Datasource Name :	Select data source	<ul> <li>()</li> </ul>	0

**?** Note A data source must be selected first, and table join queries are supported only in the same data source.

2. In the Environment Configuration section, set the Memory and Function Timeout parameters.



- 3. In the Edit query SQL section, enter an SQL statement for querying data.
  - If you set the SQL Mode parameter to **Basic SQL**, you can enter only a basic SQL statement.



(?) Note The SELECT statement specifies the parameters that the API returns. The WHERE clause specifies the request parameters of the API. You must use \${} to interpolate a request parameter.

Follow these rules when you enter an SQL statement:

- Single-table queries, table join queries, and nested queries are supported in the same data source.
- The SQL statement must meet the following requirements:
  - You can enter only one SQL statement.
  - An SQL comment is not allowed.
  - Only the SELECT statement is supported. Other statements such as INSERT, UPDATE, or DELETE are not supported.
  - The SELECT \\* statement is not supported. You must specify the columns to be queried.
  - \${param} cannot be enclosed in single quotation marks (''). For example, '\${id}' or 'ab c\${xyz}123' is not allowed. If necessary, you can use concat('abc', \${xyz}, '123') instead.
  - Parameters cannot be configured as optional.

- If the name of a column that the SELECT statement specifies has a table prefix, such as t.name, you must set an alias for the corresponding response parameter. For example, you can specify t.name as name.
- If you use an aggregate function, such as min, max, sum, or count, you must set an alias for the corresponding response parameter. For example, you can specify sum(num) as total\\_num
- \${param} in the SQL statement, including \${param} in strings, is regarded as request parameters and replaced. If an escape character (\) is placed before \${param}, \${param} is processed as a common string.
- If you set the SQL Mode parameter to **Advanced SQL**, you can enter an SQL statement with MyBatis tags.



The Advanced SQL mode supports the following types of tags: if, choose, when, otherwise, trim, foreach, and where. You can use SQL statements with MyBatis tags to implement complex query logic, such as null parameter checking, cyclic traversal over multiple values, dynamic queries, dynamic sorting, and aggregate analysis. For more information about sample code for common scenarios, see Sample code for the advanced SQL syntax used for creating an API in script mode.

4. In the right-side navigation pane, click the **Request Param** tab. In the Request Param panel, set the parameters described in the following table as required.

If you set the SQL Mode parameter to **Advanced SQL**, you must manually add all request parameters that are specified in the SQL statement to the parameter list. This ensures that the parameters that are described in the API details are consistent with the parameters that are used.

#### ? Note

- Before you preview the result, you must configure the information such as the example value, default value, and description for each request parameter of the API.
- To improve the match efficiency, specify an indexed field as a request parameter.

X Request Param						Prop
Parameter Name Bound Field						Properties
pageNum	INT	QUERY V	Equal	Yes	1	Req
⑦ pageSize	INT	QUERY ~	Equal	Yes	10	Request Param
						Respo
						Response Param
						Ne Filter
Parameter	Description					
Parameter Name	The name of the characters in leng					
Parameter Name	and hyphens (-).			-	erscores (_),	
Parameter Type	The data type of			l values: S	TRING, INT,	
	LONG, FLOAT , D	OUBLE, and BOU	ULEAN.			
	The position of t	he request para	meter. Valid \	/alues: QL	JERY and BOI	DY.
		you set the <b>Par</b> a				
	Content-Type	or more request e parameter to o	define the for			
		the request bod	-			
Parameter Position		Content-Type ation/json; chars		SON for	mat	
		ition/xml; chars		-		
		ation/x-www-fo	rm-urlencode	ed; charse	t=utf-8: the	
	FORM t	ormat.				
Required	Specifies whethe	r the request pa	rameter is re	quired.		
Example Value	The example valu	ue of the reques	t parameter.			
Default Value	The default value	e of the request	parameter.			
Description	The description c	of the request pa	arameter.			
	-					

5. In the right-side navigation pane, click the **Response Param** tab. In the Response Param panel, set the parameters as required.

If you set the SQL Mode parameter to **Advanced SQL**, you must manually add all response parameters that are specified in the SQL statement to the parameter list. This ensures that the parameters that are described in the API details are consistent with the parameters that are used.

Response Param					
Parameter Name					
⑦ pageNum		INT		分页默认参数:页编号	
⑦ pageSize		INT		分页默认参数:页大小	
⑦ totalNum		INT		分页默认参数:总数	
ld	id	INT V			
Name	name	STRING V			
Addr	addr	STRING V			
Col4	col4	STRING ~			
A maxi		n be returned if you do not enat Irned. If no request parameter i		We recommend that you enable this feature le the pagination feature.	
Pagination A maximore t	han 500 records are retu				
	han 500 records are retu D T Cl	escription he name of the res	s specified, you must enab ponse paramete and can contain	r. The name can be up to 64 letters, digits, underscores	4
Pagination A maximore to Pagination Pagination Pagination	han 500 records are retu D T Cl aı T	escription he name of the res haracters in length, nd hyphens (-). The	s specified, you must enab ponse parameter and can contain name must star response param	r. The name can be up to 64 letters, digits, underscores t with a letter. heter. Valid values: <b>ST RING</b>	4 (_),
Pagination A max more t Parameter Parameter Name	han 500 records are retu D T cl a T L	escription he name of the res haracters in length, nd hyphens (-). The he data type of the	ponse paramete and can contain name must star response param BLE, and BOOLEA	r. The name can be up to 64 letters, digits, underscores t with a letter. heter. Valid values: <b>ST RING</b> <b>AN</b> .	4 (_),

You can select **Pagination** in the **Advanced Settings** section.

Select Pagination based on your needs, and take note of the following items:

- If you do not select Pagination, the API returns up to 2,000 records.
- If the API returns more than 2,000 records, we recommend that you select **Pagination**.

(?) Note If you select Pagination in the Response Param panel, the number of records to return is subject to the pagination settings. Even if you set a limit for the number of records to return in the Edit query SQL section, this limit does not take effect.

The following common parameters are displayed when you select Pagination:

- Common request parameters
  - pageNum: the number of the page to return.
  - pageSize: the number of entries to return on each page.

- Common response parameters
  - pageNum: the number of the page to return.
  - pageSize: the number of entries returned on each page.
  - totalNum: the total number of entries returned.

**?** Note Request parameters are optional for an API. If you do not specify request parameters for an API, you must select **Pagination**.

6. Configure a filter.

If you need to process the request parameters of an API or process the results returned by an API, you can configure a filter for the API by performing the following steps: In the right-side navigation pane, click Filter. In the Filter panel, select Use Pre-filter or Use Post-filter based on your needs. After you specify the Function Type parameter, select one or more functions from the drop-down list next to the Use Pre-filter or Use Post-filter check box. If you select multiple functions, the functions are used to process parameters based on the sequence in which they are selected. Then, you can click Preview Responses Returned by API Operation and check whether the processing results of the selected functions meet your expectations. For more information about how to create and use a filter, see Create an Aviator function and use the Aviator function as a filter and Create and use Python functions as a filter.

- ? Note
  - If you want to use a Python function as a filter, you must activate DataWorks
     Professional Edition or a more advanced edition and use the shared resource group for
     DataService Studio.
  - If you want to use an Aviator function as a filter, all DataWorks editions are supported, but you must use an exclusive resource group for DataService Studio.
  - If the function that you want to use is not available in the drop-down list, check whether the function is published. If the function is not published, publish the function. Alternatively, you can create and publish a new function. For more information, see Publish a function.

	r consists of one function or multiple functions that are executed in sequence.	Properties
If the	s are classified into prefilters and post-filters. You can use prefilters and post-filters together or separately. desired function is not displayed in the drop-down list of a filter, check whether the function is published, or create a function and publish it.	Request Param
Function Type:	Aviator Function Python Function. After you switch over to another option, the functions of the pre-filter and post-filter are cleared and the types of functions are changed.	Param
	to the desired types.	Response
Use Pre-fil	er XV (	onse
	You can use a pre-filter to preprocess request parameters of an API operation.	Param
Use Post-f	tter XV (	New
	You can use a post-filter to perform secondary processing on query results of an API operation.	Ne Filter
Preview Res		Ve
1		Version

- 7. Configure a resource group for DataService Studio.
  - i. In the right-side navigation pane, click the Resource Group tab. In the panel that appears, select Exclusive Resource Group for DataService Studio or Shared Resource Group for DataService Studio for the Scheme parameter. If you select Exclusive Resource Group for DataService Studio, select an exclusive resource group from the drop-down list. If you select Shared Resource Group for DataService Studio, you do not need to select a shared resource group. DataWorks automatically manages the shared resource group.

≡	DataWorks   DataService Studio	R.,	Service Development Service Management 🔮 🗊 🔍 🗣 📑
(C)	Service Development 🛛 🖉 🕃 😷 🕀		e - Canadarana Canadarana, -
	API Name/API ID		
			X Resource Groups for DataService Studio
	the second	Sector Jacob  Click here to create a new data source  Environment Configuration  Managery Automore  Configuration  Managery Configuration  Configuration  Managery  Managery Manage	Comparison of adapting APA operations in DataService Studie consume resources in resource groups. Setect a resource group for the current APA operation based on the set procedulations and usage of resource groups. The processor is seted for testing the APA operation. After you publish the APA operation, the selected resource group in used for testing the APA operation. After you publish the APA operation, the selected resource group in used for testing the APA operation. After you publish the APA operation, the selected resource group in used for testing the APA operation. After you publish the APA operation, the selected resource group in the set of the resource group to the workspace of the resource group to the workspace that you want to select. Source

**?** Note If you cannot select an exclusive resource group from the drop-down list, log on to the DataWorks console and associate the exclusive resource group with your workspace.

8. Click the icon in the toolbar. After the API is saved, the selected resource group takes effect during the test.

After the API is configured, you can test it. For more information, see Test an API.

After the test succeeds, click Submission in the upper-right corner.

In the right-side navigation pane of the API configuration tab, click **Version**. Find the API version that you want to publish and click **Application for publication** in the Actions column to go to the application page. You can use the default application type and enter the reason for the application. Then, submit the application.

(?) Note If you have defined an approval policy in DataWorks Approval Center, the API must be approved through the approval process before the API can be published. For more information, see Overview.

After the API is published, the configuration of the resource group for DataService Studio takes effect when the API is called.

On the **Service Development** tab, you can find the registered API in the left-side navigation pane and manage the API. For example, you can clone or delete the API. On the **Service Management** tab, you can find the published API in the API list and view the details of the published API. For more information, see View, delete, move, and clone an API.

## 4.1.1.5. Sample code for the advanced SQL syntax used

## for creating an API in script mode

DataService Studio allows you to create an API in script mode. The script mode supports the basic SQL syntax and advanced SQL syntax. The advanced SQL syntax supports the following types of tags that comply with the MyBatis framework: if, choose, when, otherwise, trim, foreach, and where. You can use the advanced SQL syntax to implement complex query logic, such as null value check, traverse for multiple values, dynamic query for tables, dynamic sorting, and aggregate query. This topic provides sample code for the advanced SQL syntax used in some common scenarios.

# Example 1: Sort return results based on table fields by using conditions

```
select col01, col02
from table_name
<choose>
    <when test='var == 1'>
   order by col01
    </when>
    <when test='var == 2'>
   order by col02
    </when>
    <when test='var == 3'>
    order by col01, col02
    </when>
    <when test='var == 4'>
    order by col02, col01
    </when>
</choose>
```

## Example 2: Query data from multiple tables by using conditions

```
select col01
from
<choose>
  <when test='var == 1'>
  table_name01
  </when>
  <when test='var == 2'>
  table_name02
  </when>
  </choose>
```

```
Example 3: Determine whether to retain the WHERE clause based on whether the values of fields are null (query logic when request parameters are optional)
```

## 4.1.2. Register an API

You can register existing APIs in DataService Studio to centrally manage, publish, and integrate the APIs. This topic describes how to register and configure an existing API.

## **Register an API**

If you have no APIs, you must create an API. For more information, see Create an API in the codeless UI.

1.

2. In the Service Development pane, move the pointer over the 📑 icon and choose New API >

Register API.

You can also click a business process, right-click API, and then choose New API > Register API.

3. In the Register API dialog box, set the parameters as required.

Register API		×
* API Name :	0/50	
	An API name must start with a letter or Chinese character, and can contain numbers, letters, Chinese characters, and underscores (_). The name must be 4 to 50 characters in length.	
* API Path :	0/200	
	The API path is the alias of the backend request path. It can contain letters, numbers, underscores (_), and hyphen (-). It must start with a forward slash (/), and can be up to 200 characters in length. If the API path contains a path parameter from the request, include the path parameter in brackets []. The path parameter name must be the same as that in the backend request path.	
* Protocol :	🖌 HTTP 🖌 HTTPS	
* Request Method :	GET Y	
* Response Content	JSON 🗸	
Type :		
* Visible Range :	Work Space 🗸 🕥	
Label :	Please Select V	
	Up to 0-5 tags can be set up, Chinese characters, English, numbers, underline, and no more than 20 characters per label	
* Description :		
	0/2000	
* Target Folder :	Please Select V	
	OK	

Parameter	Description
API Name	The name of the API. The name must be 4 to 50 characters in length, and can contain letters, digits, and underscores (_). It must start with a letter.
	The path of the API, such as <i>/user</i> .
API Path	<b>Note</b> The path can be up to 200 characters in length, and can contain letters, digits, underscores (_), and hyphens (-). It must start with a forward slash (/).
Protocol	The protocol used by the API. Valid values: <b>HTTP</b> and <b>HTTPS</b> . If you need to call the API by using HTTPS, you must bind an independent domain name to the API in the API Gateway console after the API is published to API Gateway. In addition, you must upload an SSL certificate in the API Gateway console. For more information, see <b>Enable HTTPS for an API</b> <b>operation</b> .
Request Method	The request method used by the API. Valid values: $\mbox{GET},\mbox{POST},\mbox{PUT},\mbox{and}\mbox{DELETE}.$

Parameter	Description					
Response Content Type	The format of the data returned by the API. Valid values: JSON and XML.					
Visible Range	<ul> <li>The range of users to whom the API is visible. Valid values:</li> <li>Work Space: The API is visible to all members in the current workspace.</li> <li>Private: The API is visible only to its owner, and permissions on the API cannot be granted to other members.</li> <li>Note If you set this parameter to Private, other members in the workspace cannot view the API in the API list.</li> </ul>					
Label	The tags to be added to the API.          Image: The tags to be added to the API.         Image: The tags to be up to 20 characters in length, and can contain letters, digits, and underscores (_). You can set at most five tags for an API.					
Description	The description of the API. The description can be up to 2,000 characters in length.					
Destination Folder	The folder that stores the API.					

4. Click OK.

## Configure the API

1. Double-click the registered API. On the configuration tab of the API, set the parameters as required in the **Define Backend Service** section.

Define the back-end Se	rvice
* Back-end Service	
Host: The s	specified value must start with http:// or https://, and can cannot contain the path.
* Back-end Service	
	h must start with a forward slash (/), and can contain letters, numbers, underscores (), and hyphens (-). The path can be up to 200 characters in length. backend service Path include the Parameter Path, it must be placed in [], such as /user/[userid]
Back-end Service 0	ms
Timeout :	
Parameter	Description
llast	The host of the registered API. The hostname must start with http:// or
Host	https://, and cannot contain the path.
	The path of the registered API. The path can contain parameters that are
	enclosed in brackets []. Example: /user/[userid].
Path	In the next step, parameters that are defined in the Path parameter are
	automatically added to the request parameter list.

Parameter	Description
Back-end Service Timeout	The timeout period of the backend service.

2. In the **Define Request Parameters** section, set the parameters as required.

Define Request Parameters										
Request Parameters										
Parameter Name		Parameter Type	Required	Example Value	Default Valu		Description			
	QUERY	String 🗸								
Constant Parameters										
Parameter Name	Parameter Po		arameter Type	Default Value		Description				
	QUERY		STRING V							

Parameter or section	Description
Request Parameters	You can click <b>Add Parameter</b> to add a request parameter for the API. For each request parameter, you can set the <b>Parameter Position</b> field to <b>QUERY</b> , <b>HEAD</b> , or <b>BODY</b> . The valid values of the Parameter Position field vary based on the request method that is used by the API. Select one from the valid values that are displayed.
Constant Parameters	Constant parameters have fixed values and are invisible to API callers. The constant parameters do not need to be specified during an API call. However, the backend service receives the defined constant parameters and their values in each API call. The constant parameters are applicable when you need to fix the value of a parameter or hide the parameter from API callers.
Request Body Description	This parameter is displayed only if you set the Request Method parameter to POST or PUT. In the Request Body Description parameter, you can enter the body description in the JSON or XML format. This way, you can provide an example of the request body for API callers to determine the format of the request body.

- 3. In the **Define Response Content** section, set the **Correct Response Example** and **Error Response Example** parameters. These examples are references for API callers to write the code for parsing the results of the API.
- 4. In the **Define Error Codes** section, set the **Error Code**, **Error Message**, and **Solution** parameters to define an error code. The information helps API callers diagnose the causes of the error.
- 5. Click the 🔛 icon in the toolbar.

After the API is configured, you can test it. For more information, see Test an API.

After the test succeeds, click **Submission** in the upper-right corner.

In the right-side navigation pane of the API configuration tab, click **Version**. Find the API version that you want to publish and click **Application for publication** in the Actions column to go to the application page. You can use the default application type and enter the reason for the application. Then, submit the application.

**Note** If you have defined an approval policy in DataWorks Approval Center, the API must be approved through the approval process before the API can be published. For more information, see **Overview**.

After the API is published, the configuration of the resource group for DataService Studio takes effect when the API is called.

On the **Service Development** tab, you can find the registered API in the left-side navigation pane and manage the API. For example, you can clone or delete the API. On the **Service Management** tab, you can find the published API in the API list and view the details of the published API. For more information, see View, delete, move, and clone an API.

# 4.2. Test an API

This topic describes how to test an API that is being developed or has been published.

## Test an API that is being developed

You can test an API that is being developed on the **Service Development** tab. Before you perform such a test, create or register an API.

1.

- 2. On the Service Development tab, double-click the target API in the API list.
- 3. On the configuration tab of the API, click **Test** in the upper-right corner.

C D		Test Publish
Select Table		Prop
* Datasource Type : MySQL	×	erties

4. In the Test APIs dialog box, click Test to send an API request.

The request and response details appear on the right. If the API fails the test, check the error message, modify the API settings, and then test the API again.

You can select Save the correct response example automatically as required.

• The system automatically generates sample failure responses and error codes when it tests an API. However, the system does not automatically generate sample success responses.

To allow the system to save the success test result as a sample success response, you must select **Save the correct response example automatically** before you perform the test. If the response contains sensitive data that must be de-identified, you can manually edit the response.

- The sample success response is an important reference for API callers, and therefore must be configured.
- The Call Latency value is the latency of the current API request, which is used to evaluate the API performance. If the latency is long, consider optimizing the database.

## Test an API that has been published

You can test an API that has been published on the **Service Management** tab. Before you perform such a test, publish an API. For more information, see Publish an API.

- 1. On the Service Development tab, click the Service Management tab in the upper-right corner.
- 2. In the left-side navigation pane, click **Test APIs**.
- 3. Select the API to be tested from the drop-down list, and set all the request parameters of the API.

≡	ntaWorks   [	DataService Studio	• •				Service Development Service Management	ር ፲ ዲ 🔻		
O Mar	nage APIs	Test APIs								
🖬 арі	I Statistics 🗸 🗸	newlightning00				Request Details				
Tes	st APIs	API Path:								
- → Call		Request Parameters								
- Can	Call API3									
		(2) pageNum		QUERY						
		pageSize		QUERY	10					
						Response Details				

4. Click Test. The request and response details appear on the right.

**?** Note On the Test APIs page, you can only test published APIs, but not update sample success responses for APIs.

## 4.3. Publish an API

This topic describes how to publish an API that is created or registered in DataService Studio to API Gateway and publish the API to Alibaba Cloud Marketplace.

## Prerequisites

API Gateway is activated.

## Context

API Gateway provides API lifecycle management services, including API publishing, management, maintenance, and monetization. It provides a simple, fast, cost-effective, and low-risk service for you to aggregate microservices, separate the frontend from the backend, integrate systems, and provide features and data to partners and developers. For more information, see What is API Gateway?.

You can publish APIs that are created or registered in DataService Studio to API Gateway, on which you can perform a variety of management operations. For example, you can manage permissions, configure bandwidth throttling, configure access control, and measure usage for APIs. After you create or register APIs in DataService Studio, you must publish the APIs to API Gateway before the APIs can be called by other users or in your own applications. DataService Studio is integrated with API Gateway to allow you to publish APIs to API Gateway with a few clicks.

## Publish an API to API Gateway

- 1.
- 2. On the Service Development tab, double-click the API that you want to publish in the API list.
- 3. On the configuration tab of the API, click the 🔤 icon in the toolbar.

After the API is configured, you can test it. For more information, see Test an API.

The system creates a group in API Gateway with the same name as the API group in DataService Studio to which the API belongs, and publishes the API to this group.

4. After the test succeeds, click **Submission** in the upper-right corner.

In the right-side navigation pane of the API configuration tab, click **Version**. Find the API version that you want to publish and click **Application for publication** in the Actions column to go to the application page. Use the default application type **Publish data service API** and enter the reason for the application in the **Reason for application** field. Then, click **Apply for permission** to submit the application.

(?) **Note** If you have defined an approval policy in DataWorks Approval Center, the API must be approved through the approval process before the API can be published. For more information, see **Overview**.

5. After the application is approved, click **Version** in the right-side navigation pane of the API configuration tab. Find the approval record and click **Publish** in the Actions column.

After the API is published, you can go to the API Gateway console to view API details or configure bandwidth throttling and access control for the API.

If you need to call the API in your own application, create the application and authorize the application to use the API in the API Gateway console. Then, call the API in your application by signing the API request with the AppKey and AppSecret. For more information, see Call an API. API Gateway also provides SDKs for mainstream programming languages. You can use the SDKs to integrate APIs with your application. For more information, see SDK downloading and usage guide.

## Publish an API to Alibaba Cloud Marketplace

The API sector of Alibaba Cloud Market place provides thousands of API products in the following categories: finance, AI, e-commerce, transportation and geography, living services, corporate management, and public affairs. Alibaba Cloud Market place provides a platform in which you can monetize your data.

After you publish APIs that are created or registered in DataService Studio to API Gateway, you can publish them to Alibaba Cloud Market place with a few clicks. This is an easy way to achieve financial gains for your enterprise.

Before you publish an API to Alibaba Cloud Market place, you must register with Alibaba Cloud Market place as an independent software vendor (ISV).

**Note** Only enterprise users can register with Alibaba Cloud Market place as an ISV.

- 1. Log on to the ISV portal of Alibaba Cloud Marketplace.
- 2. In the left-side navigation pane, click Listing.
- 3. Click Create Product.
- 4. On the Access Information page, set the parameters as required.

# 4.4. Authorize access to an API

This topic describes how to authorize access to an API, view the APIs that you are authorized to access, and view the APIs that you authorize others to access.

## Prerequisites

You can unpublish, authorize access to, or change the protocols of APIs only after the APIs are published. For more information, see Publish an API.

## Go to the Manage APIs page

- 1. Log on to the DataWorks console.
- 2. In the left-side navigation pane, click **Workspaces**.
- 3. After you select the region where the required workspace resides, find the workspace and click **DataService Studio**.
- 4. Click Service Management in the upper-right corner. The APIs of Published tab of the Manage APIs page appears.

≡	n DataWorks	Works   DataService Studio									Ω
		APIs of Published Authorized to Use Authorize Others to Use									
٥		API Name V Please	e Input	Q	Clear C Refresh Ad						
•	API Statistics										
Ø	Test APIs										

## Authorize access to an API

- 1. On the APIs of Published tab, find the required API and click Authorize.
- 2. In the API Authorization dialog box, set the parameters as required.

API Authorization			>	×
API Name : * Tenant Account ID : Yo		our tenant account ID on the account m	anagement page.	
	lease Select		×	
* Validity Period : 🧕	Limited	Select Date		
			OK Cancel	
Parameter		Description		
API Name		The name of the API to which you wa cannot change the value.	nt to authorize access. You	

Parameter	Description
Tenant Account ID	The ID of the Alibaba Cloud account to which you want to grant the permissions to call the API. You can go to the <b>Account</b> <b>Management</b> page to view the account ID.
Authorized Workspace	The name of the workspace to which you want to grant the permissions to call the API. You can select a workspace under the current Alibaba Cloud account from the drop-down list.
Validity Period	<ul> <li>The validity period of the permissions to call the API. Valid values: Limited and Unlimited.</li> <li>Limited: specifies that the authorized user has the permissions to call the API before the specified expiration date.</li> <li>Unlimited: specifies that the authorized user can permanently call the API.</li> </ul>

#### 3. Click **OK**.

On the **APIs of Published** tab, you can also perform the following operations:

• Find the required API and click **Unpublish**. In the **Unpublish** API message, click OK to unpublish the API.

### ? Note

- If you unpublish or delete an API after you authorize a workspace to access the API, the API can no longer be called by the workspace.
- If you publish an API again after you unpublish or modify the API, you must authorize access to the API again.
- Find the required API and click **Test** to test the API on the **Test APIs** page. For more information, see **Test an API**.
- Find the required API, move the pointer over **More** and select **Change Protocol**. In the **Change Protocol** dialog box, change the protocol used by the API and click OK.

### ? Note

- If you deselect a protocol, the API can no longer be called by using this protocol. Use caution when you perform this operation.
- The protocol change takes effect in real time.

## View the APIs that you are authorized to access

On the **Manage APIs** page, click the **Authorized to Use** tab to view the APIs that you are authorized to access.

You can perform the following operations on the APIs that you are authorized to access:

- Find the required API and click **Test** to test the API on the **Test APIs** page. For more information, see Test an API.
- Find the required API and click Delete. In the Delete authorized message, click OK to remove your

permissions on the API.

## View the APIs that you authorize others to access

On the **Manage APIs** page, click the **Authorize Others to Use** tab to view the APIs that you authorize others to access.

You can perform the following operations on the APIs that you authorize others to access:

- Find the required API and click **Test** to test the API on the **Test APIs** page. For more information, see Test an API.
- Find the required API and click Manage. In the Authorization dialog box, revoke or modify the permissions of a workspace on the API.

# 4.5.1. Call an API

This topic describes how to call an API after the API is published to API Gateway.

## Prerequisites

An API is published to API Gateway. For more information, see Publish an API.

The following conditions are met:

- The parameter definition of the API is obtained.
- The AppKey and AppSecret of the application that needs to call the API are obtained. For more information.
- The application is authorized to call the API. For more information, see Authorize an application to call an API operation.

## Context

API Gateway allows you to authorize access to APIs and use SDKs to integrate APIs with applications. You can authorize your own account, a user in your enterprise, or a third party to call APIs.

## Procedure

1. Obtain the API documentation.

The method of obtaining the API documentation varies depending on how you obtain the API. You can obtain an API in one of the following ways:

- Purchase the API in Alibaba Cloud Market place. For more information about the APIs that you have purchased, log on to the Alibaba Cloud Market place console.
- Obtain the authorization from the independent software vendor (ISV) for calling the API. For more information about the APIs that you are authorized to call, log on to the API Gateway console.
- 2. Create an application in API Gateway.

In API Gateway, applications are identities that you use to call APIs. Each application has a key pair: AppKey and AppSecret, which are equivalent to an account and the corresponding password.

3. Obtain the permission to call the API.

Authorization means granting the application the permission to call an API. Your application must

## be authorized to call the API.

The authorization method varies depending on how you obtain the API. For more information, see Authorize an application to call an API operation.

4. Call the API.

You can send an HTTP or HTTPS request to call the API. Before you call the API, you can use examples of calling APIs in multiple languages in the API Gateway console to test the call. For more information, see Enable HTTPS for an API operation.

## View the authentication information for calling APIs

- 1. Log on to the DataWorks console.
- 2. In the left-side navigation pane, click **Workspaces**.
- 3. In the top navigation bar, select the region where the target workspace resides. Find the target workspace and click **DataService Studio** in the Actions column.
- 4. On the Service Development tab, click Service Management in the upper-right corner.
- 5. In the left-side navigation pane, click Call APIs.

On the **Call APIs** page, view or copy the authentication information that is required to call APIs.

# **5.Function** 5.1. Create and manage functions

This topic describes how to create, clone, delete, or move a function.

## Prerequisites

- A business process is created. For more information, see Create and manage business processes.
- An exclusive resource group for DataService Studio is purchased. For more information, see Create and use an exclusive resource group for DataService Studio.

Onte To run an Aviator function, you must use an exclusive resource group for DataService Studio.

## Context

DataService Studio supports Aviator functions and Python functions.

- The feature that allows you to use a Python function as an API filter is being optimized. You cannot create, clone, or publish a Python function.
- If you want to use a function as an API filter, we recommend that you use an Aviator function. For more information, see Create an Aviator function and use the Aviator function as a filter.

## Limits

- Before you can use a function as an API filter, you must activate DataWorks Basic Edition or a more advanced edition.
- For the same API, the function that you use as the prefilter and the function that you use as the post-filter must be of the same type.

## Create a function

1.

2. On the Service Development page, move the pointer over the price icon and choose Create

Function > Create Aviator Function.

You can also find the desired business process, right-click the name of the business process, and then choose **New > Function > Create Aviator Function**.

3. In the **Create Aviator Function** dialog box, configure the parameters. The following table describes the parameters.

×
elect
0/512
vs/dataworks_test_zs/Function
OK Cancel
Description
The name of the function. The name can be up to 256 characters in length.
The template that you want to use to create the function. Set the value to <b>AviatorScript Standard v1</b> .
The description of the function. The description can be up to 512 characters in length.
The folder in which you want to store the function.
<b>Note</b> The value of this parameter can be a business process name, a folder name, or a function category. If no business process is available, you must create a business process.

- 4. Click Determine.
- 5. Configure the function on the configuration tab.
  - i. In the Edit Code section, enter the code for the function.
  - ii. In the Environment Configuration section, configure the Function Timeout parameter.
- 6. Configure an exclusive service resource group for DataService Studio.

In the right-side navigation pane of the configuration tab of the function, click **Resource Group**. On the Resource Group tab, select **Exclusive Resource Group for DataService Studio** for the **Scheme** parameter and select an exclusive resource group from the **Exclusive Resource Group for DataService Studio** drop-down list. **Note** If the exclusive resource group for DataService Studio that you want to select is not displayed in the drop-down list, you can go to the Resource Groups page in the DataWorks console, and then associate the exclusive resource group with your workspace on the Exclusive Resource Groups tab. You can also click Purchase Resource Group for DataService Studio on the Resource Group tab to purchase an exclusive resource group for DataService Studio. For more information, see Create and use an exclusive resource group for DataService Studio.

7. Click the icon in the upper-left corner of the configuration tab of the function to save the

configurations.

After the function is created, you can test and publish the function. Then, you can associate the function with an API and use the function as the prefilter or post-filter for the API.

## View the information about the function

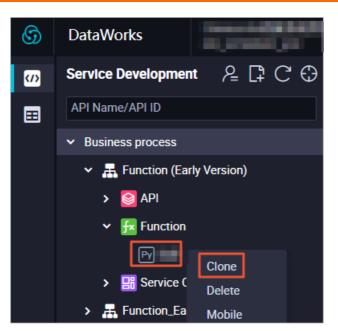
After the function is created, you can view the information about the function. The following figure shows how to view the association relationship of the function.

Av 前	置请求	参数改值 ×							≡
•	C	88							mit
E	dit Cod	le	× Association Relation	iship					Prop
		## AviatorScript ## \$0: the default name c	Published (0) Un	published (2)			C R	efresh	Properties
								Act	New
		## Example 1: ## \$0.a: Get the attribut ## \$0.b[index]: Get the i	417381955945694	Holo字段名称大写小写_向导 /holo/upper_lower/xiangdao	dataworks_test_zs	dataworks_demo2	Post-filter: Enable Prefilter: Enable		Associatio
		## Example 2: ## Input: {"keyA":[1,2],'	417381955947689	Holo字段名称大写小写_向导2 /holo/big_small/x	dataworks_test_zs	dataworks_demo2	Prefilter: Enable		Association Relationship
		<pre>## Function: \$0.keyA[0]=: ## Output: {"keyA":[3,2],</pre>					< Previous 🚺 Ne:		nship
		return <b>\$0;</b>							Version
									Resource Group

- In the right-side navigation pane of the configuration tab of the function, click **Properties**. On the Properties tab, you can view the basic information about the function, such as the **function ID** and creation time. A function ID uniquely identifies a function and can be used to search for the function.
- In the right-side navigation pane of the configuration tab of the function, click Association Relationship. On the Association Relationship tab, you can view the association relationships between the function and published or unpublished APIs. We recommend that you view the association relationships before you delete or modify the function to know the APIs that will be affected by the operation.
- In the right-side navigation pane of the configuration tab of the function, click **Version**. On the Version tab, you can view the details of the function version.
- In the right-side navigation pane of the configuration tab of the function, click **Resource Group**. On the Resource Group tab, you can view the resource group that is used by the function. You can also change the resource group that is used by the function.

## Manage functions

- 1. On the **Service Development** page, find the **function** that you want to use as a filter for an API in a **business process**.
- 2. Right-click the function name. In the shortcut menu that appears, you can select an option to **clone**, **delete**, or **move** the function.



• Select Clone. In the Clone Function dialog box, configure the Function Name and Destination Folder parameters. The function is cloned.

**Note** The name of the function can be 4 to 50 characters in length and can contain letters, digits, and underscores (\_). The name must start with a letter.

• Select **Delete**. In the **Delete Function** message, click **Determine**. The function is deleted.

**?** Note If the function is referenced by specific APIs, the function fails to be deleted. In this case, you must remove the function from the filters of the APIs before you can delete the function.

• Select Mobile. In the **Modify file path** dialog box, configure the **Destination Folder** parameter. The function is moved to the destination folder.

## What's next

- After you create the function, you can test and publish the function. For more information, see Test a function and Publish a function.
- After you publish the function, you can associate the function with an API and use the function as a prefilter or a post-filter for the API. This way, the function can process the request parameters or responses of the API. For more information, see Create an Aviator function and use the Aviator function as a filter and Create and use Python functions as a filter.

# 5.2. Test a function

This topic describes how to test a function.

## Prerequisites

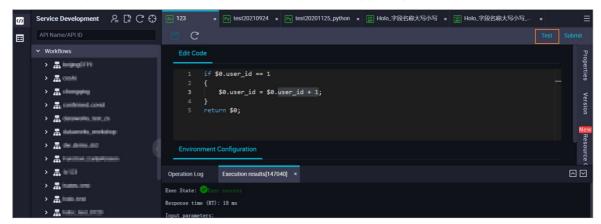
A function is created and saved. For more information, see Create and manage functions.

## Test a function

<sup>&</sup>gt; Document Version: 20220705

1.

- 2. In the left-side navigation pane of the **Service Development** page, find the function that you want to test and double-click the name of the function.
- 3. On the configuration tab of the function, click **Test** in the upper-right corner.



4. In the Function Test dialog box, specify Input parameters and click OK.

You can view the running status on the **Execution results** tab at the bottom. If the test fails, perform modifications based on the error message and test the function again.

# 5.3. Publish a function

You can associate a function with an API and use the function as a prefilter or a post-filter for the API only after you publish the function. This way, the function can process the request parameters or responses of the API. This topic describes how to publish a function on the DataService Studio page.

## Prerequisites

- A function is created and saved. For more information, see Create and manage functions.
- The function is tested and successfully executed. For more information, see Test an API.

## **Publish a function**

1.

- 2. In the left-side navigation pane of the **Service Development** page, find the function that you want to publish and double-click the name of the function.
- 3. On the configuration tab of the function, click **Submit** in the upper-right corner.
- 4. In the right-side navigation pane of the configuration tab, click **Version**. On the Version tab, find the version that is in the **Can Be Published** state and click **Publish** in the Actions column.

ŝ	Service Development $\begin{tabular}{c} \begin{tabular}{c} tab$	A THE X PARTY AND	≡
≡	API Name/API ID	🖱 C Test	
	V Workflows	Edit Code         X         Version           1         if \$0.user_id == 1         Function ID: 514           2         {         s         Actions           3         \$0.user_id = \$0.user_id + 1;         s         Actions           4         }         Future \$0;         Publish Deprecate   Version Details   Roll Back	Properties Version
<	A datamatu, mututop     A datamatu, mututop     A datamatut     A function Latylement	Environment Configuration	Resource (
	> # em	Operation Log Execution results(147040) ×	$\Box$
	<ul> <li>A Index and</li> <li>A Index and</li> <li>A Index and</li> </ul>	Exec State: Skroc success Response time (RT): 18 ms Input parameters:	

5. After the function is published, click **Version** in the right-side navigation pane of the configuration tab to view the publishing record. You can also click **Version Details** or **Roll Back** in the Actions column.

## What's next

You can use the published function as a prefilter or a post-filter for an API. For more information, see Create and use Python functions as a filter and Create an Aviator function and use the Aviator function as a filter.

# 5.4. Use filters

## 5.4.1. Create an Aviator function and use the

## Aviator function as a filter

This topic describes how to create an Aviator function and use the Aviator function as the prefilter or post-filter for an API.

## Context

On the DataService Studio page, you can create, test, and publish a function. Then, you can associate the function with an API, and use the function as a filter to process the request parameters or responses of the API to perform complex queries based on your business requirements. For the sample code that shows how to use an Aviator function as a filter for an API in common scenarios, see Best practices of using Aviator functions as filters.

## Limits

- You must use an exclusive resource group for DataService Studio to run an Aviator function.
- For the same API, the function used as the prefilter and the function used as the post-filter must be of the same type. You are not allowed to use an Aviator function and a Python function as the filters for the same API.

## Precautions

To ensure data security and function performance, you must use an exclusive resource group for DataService Studio to run an Aviator function.

• When you edit or test a function, you must associate the function with an exclusive resource group for DataService Studio. For more information, see Create and manage functions.

• If you enable the filter feature for an API and associate the API with a function when you develop the API, you must associate the API with an exclusive resource group for DataStudio Service. For more information, see Create an API in the codeless UI.

## Syntax of an Aviator function

The system provides the following function template for an Aviator function. For more information about the syntax of an Aviator function, see Aviator function syntax.

```
## --- AviatorScript ---
## $0: the default name of the input param.
## Expressions must be separated by the semicolon ";".
## Example 1:
## $0.a: Get the attribute "a" from the input param "$0".
## $0.b[index]: Get the index of the array "b" from the input param "$0". The index cannot
be an expression or a variable.
## Example 2:
## Input: {"keyA":[1,2],"keyB":"value"}
## Function: $0.keyA[0]=3; $0.keyB="new_value"; return $0;
## Output: {"keyA":[3,2],"keyB":"new_value"}
return $0;
```

You can modify the function based on your business requirements.

- The default name of the input parameter of an Aviator function is \$0 and cannot be changed. You can use the \$0 parameter to obtain the input of the Aviator function.
- The input of a function must be in the JSON format. In DataService Studio, a function is used as a filter to process the request parameters or responses of an API. The request parameters of an API are defined in the single-layer JSON format. The responses of the API are defined in the multi-layer JSON format. Therefore, the input of a function must also be in the JSON format. The so parameter specifies input data in the JSON format.
- If a function is used as the prefilter for an API, the system converts request parameters of the API or outputs of the previous function from key-value pairs into data in the JSON format. Then, the system uses the data in the JSON format to replace the so parameter. The data type of values in the JSON data can only be STRING.
- If a function is used as the prefilter for an API, the outputs of the function are key-value pairs . The key-value pairs are used as the input parameter of an SQL statement. Therefore, the key-value pairs cannot contain nested key-value pairs.

## Create a function

1.

2. On the Service Development page, move the pointer over the picon and choose Create

#### Function > Create Aviator Function.

You can also find the desired business process, right-click the name of the business process, and then choose **New > Function > Create Aviator Function**.

3. In the **Create Aviator Function** dialog box, configure the parameters. The following table describes the parameters.

×
elect
0/512
vs/dataworks_test_zs/Function
OK Cancel
Description
The name of the function. The name can be up to 256 characters in length.
The template that you want to use to create the function. Set the value to <b>AviatorScript Standard v1</b> .
The description of the function. The description can be up to 512 characters in length.
The folder in which you want to store the function.
<b>Note</b> The value of this parameter can be a business process name, a folder name, or a function category. If no business process is available, you must create a business process.

- 4. Click Determine.
- 5. Configure the function on the configuration tab.
  - i. In the Edit Code section, enter the code for the function.
  - ii. In the Environment Configuration section, configure the Function Timeout parameter.
- 6. Configure an exclusive service resource group for DataService Studio.

In the right-side navigation pane of the configuration tab of the function, click **Resource Group**. On the Resource Group tab, select **Exclusive Resource Group for DataService Studio** for the **Scheme** parameter and select an exclusive resource group from the **Exclusive Resource Group for DataService Studio** drop-down list. **Note** If the exclusive resource group for DataService Studio that you want to select is not displayed in the drop-down list, you can go to the Resource Groups page in the DataWorks console, and then associate the exclusive resource group with your workspace on the Exclusive Resource Groups tab. You can also click Purchase Resource Group for DataService Studio on the Resource Group tab to purchase an exclusive resource group for DataService Studio. For more information, see Create and use an exclusive resource group for DataService Studio.

7. Click the icon in the upper-left corner of the configuration tab of the function to save the configurations.

After the function is created, you can test and publish the function. Then, you can associate the function with an API and use the function as the prefilter or post-filter for the API.

## Use a function as a filter for an API

- 1. On the Service Development page, click the business process to which the desired API belongs.
- 2. Find the desired API and double-click the API name. The configuration tab of the API appears.
- 3. In the right-side navigation pane of the configuration tab of the API, click Filter.

On the Filter tab, select **Use Pre-filter** or **Use Post-filter** based on your business requirements. The selected **Aviator** check box is highlighted upon your selection. Select the function that you want to use as a prefilter or post-filter from the Use Pre-filter or Use Post-filter drop-down list. You can select multiple functions. When you execute these functions, the request parameters or responses of an API with which you associate the multiple functions are processed in the same sequence as the functions are selected.

### ? Note

- A filter consists of one or more functions. You can use a filter to preprocess the request parameters of an API or perform secondary processing on query results of the API.
- If the function that you want to use is not displayed in the Use Pre-filter or Use Postfilter drop-down list, check whether the function is published. If the function is not published, publish the function. You can also create and publish another function. For more information, see Publish a function.

Select Table	X Filter			Properties
* Datasource Type:		ne or more functions that are executed in sequence. only in an exclusive resource group for DataService Studio. You can click Resource Groups in the		erties
Datasource Name:     Click here to create a new d     Environment Configuration	right-side navigation A filter can be a pre- If you cannot select	In the exclusive resource group for balaservice studio, Fou can click Resource broups in the pane and select an exclusive resource group for DataService Studio on the page that appears. filter or a post-filter. You can use them separately or use them together. a function that you want to execute from the filter drop-down list, check whether the function is rely, create a function and publish the function.		Request Param
* Memory: 4096M	Function Type: 🚺 Aviat	tor Function O Python Function		
Function Timeout. 10000 ms Edit query SQL	After you	or runction $\bigcirc$ Pyrion runction switch over to another option, the functions of the pre-filter and post-filter are cleared and the types of functions are ch sired types.		Response Param
1 SELECT json_agg(json_build_	Vse Pre-filter	×~	C	Param
2 FROM public.test				New
	Use Post-filter			Ne Filter
		use a post-filter to perform secondary processing on query results of an API operation.		Ver
	Preview Responses Returned	by API Operation		Version
				New
				Resource Group

4. Click **Preview Responses Returned by API Operation** to check whether the result that you obtain by using the filter meets your expectations.

## 5.4.2. Create and use Python functions as a filter

This topic describes how to create and use Python functions as a filter to process the request parameters of an API operation or the results returned by an API operation.

## Context

On the DataService Studio page, you can create, test, and publish a function. Then, you can associate the function with an API, and use the function as a filter to process the request parameters or responses of the API to perform complex queries based on your business requirements. For the sample code that shows how to use an Aviator function as a filter for an API in common scenarios, see Best practices of using Aviator functions as filters.

## **Usage notes**

- You must activate DataWorks Professional Edition or a more advanced edition before you use Python functions as a filter.
- Python functions must be executed on the shared resource group for DataService Studio.
- The feature of using Python functions as a filter is being optimized. You cannot create, clone, or publish new Python functions. If you need to use this feature, we recommend that you use Aviator functions. For more information, see Create an Aviator function and use the Aviator function as a filter.
- If you use one or more functions as a filter to process the request parameters of an API and one or more functions as a filter to process the results returned by the API, you must use the same type of function as the filters. You can use either Aviator or Python functions. You cannot use different types of functions as the filters for an API.

If you use Python functions as a filter, take note of the following limits:

- Only Python 3.0 functions can be used as a filter.
- Python functions support the import of only the following modules: json, time, random, pickle, re, and math .
- The name of a function used as a filter must be def handler (event, context): .

## Function template

```
# -*- coding: utf-8 -*-
# event (str) : in filter it is the API result, in other cases, it is your param
# context : some environment information, temporarily useless
# import module limit: json,time,random,pickle,re,math
# do not modify function name
import json
def handler(event,context):
    # load str to json object
    obj = json.loads(event)
    # add your code here
    # end add
    return obj
```

## You can modify the function template to write your own function. You can modify the names of the input parameters as needed.

Parameter 1 [context]: the context of calling the API. The value is of the STRING type. Thi s parameter is not in use and is left empty.

Parameter 2 [event]: the request parameters of the API or the result data returned by the p revious function. The value is of the STRING type.

## ? Note

- The system converts the request parameters of an API or the result data returned by the previous filter from a key-value map to a JSON object, and passes the JSON object to the event parameter. The values in the JSON object must be of the STRING type.
- The processing result of a filter used to process the request parameters of an API is a keyvalue map. The key-value map is used as the input parameter of an SQL statement. Therefore, the key-value map cannot contain nested key-value pairs.

## Create a Python function

1.

2. In the Service Development pane, move the pointer over the picon and choose Create

#### Function > Create Python Function.

You can alternatively click the required business process, right-click Function, and then choose Create Function > Create Python Function.

3. In the Create Python Function dialog box, set the parameters as required.

Create Python Fun	ction		×
* Function Name:			
* Function Template:	Please Select	~	
* Running <sub>.</sub>	Python 3.0	~	
Environment			
Function Description:			
			0/512
* Destination Folder:	Business Processes/datawo	rks_test_zs	~
		ОК	Cancel
Parameter		Description	
Function Name		The name of the function. The name 256 characters in length.	can be up to
Function Template		The template that is used to create t Set the value to Python3 Standard v1	
Running Environment		The runtime environment of the funct value to Python 3.0.	ion. Set the
Function Description		The description of the function. The c can be up to 512 characters in length	
Destination Folder		The folder for storing the function.	

- 4. Click OK.
- 5. Configure the function on its configuration tab.
  - i. In the Edit Code section, enter the function code.
  - ii. In the **Environment Configuration** section, set the **Memory** and **Function Timeout** parameters.
- 6. Click the 🔤 icon in the toolbar.

## Use a function as a filter for an API

- 1. In the Service Development pane, click the business process to which the required API belongs.
- 2. Double-click the required API. The configuration tab of the API appears.

3. In the right-side navigation pane, click the **Filter** tab.

In the Filter panel, select Use Pre-filter or Use Post-filter based on your needs. Select Python Function as Function Type. Then, you can select one or more functions from the drop-down list next to the Use Pre-filter or Use Post-filter check box. If you select multiple functions, the functions are used to process parameters based on the sequence in which they are selected.

## ? Note

- You can specify one or more functions as a filter to process the request parameters of an API or the results returned by an API.
- If the function that you want to use is not available in the drop-down list next to the Use Pre-filter or Use Post-filter check box, check whether the function is published. If the function is not published, publish the function. You can also create and publish a new function. For more information, see Publish a function.

Select Table  * Datasource Type: * Datasource Name:	Lightning API_test Click here to create a new d	Filters are If the desir	nsists of one function or multiple functions that are executed in sequence. classified into prefilters and post-filters. You can use prefilters and post-filters together or separately. ed function is not displayed in the drop-down list of a filter, check whether the function is published, or create ction and publish it.		Properties Reques
Environment Con * Memory: Function Timeout:	figuration 4096M 10003 ms	Function Type:	Aviator Function Python Function After you switch over to another option, the functions of the pre-filter and post-filter are cleared and the types of functions are chat to the desired types.		Request Param Res
Edit query SQL	region AS region, zo	🛃 Use Pre-filter	You can use a pre-filter to preprocess request parameters of an API operation.	C	Response Param
		Use Post-filter	You can use a post-filter to perform secondary processing on query results of an API operation.	C	New Filter
		1	s Returned by API Operation		Version
					Resource Group

4. Click **Preview Responses Returned by API Operation** and check whether the processing results of the selected functions meet your expectations.

# 5.4.3. Best practices of using Aviator functions as filters

In DataService Studio, you can write functions, associate a single function or multiple functions of the same type with an API, and use the functions as filters for the API. This topic provides examples on how to write an Aviator function that has a specific logic based on Aviator function syntax.

## Scenarios

- If you use a function as the prefilter for an API, the function preprocesses the request parameters of the API when the API is called.
- If you use a function as the post-filter for an API, the function processes and reconstructs the responses of the API when the API is called.

## References

- For more information about how to create, test, submit, publish, delete, move, and clone a function, see Create and manage functions, Test a function, and Publish a function.
- For more information about how to use a function as the prefilter or post-filter for an API, see Create an Aviator function and use the Aviator function as a filter.
- For more information about Aviator function syntax, see Function syntax.

## Input parameters of a function

**Note** The name of the default variable in the code for an Aviator function is so, which indicates the entire input of the Aviator function.

• Scenario 1: Use an Aviator function as the prefilter for an API.

The request parameters and the values of the request parameters of the API are used as the entire input for the Aviator function. The input is in a single-layer JSON format. Example:

```
## user_id is the request parameter.
## user_id = (0,1,2) indicates that multiple values are passed in for the request paramet
er when the API is called.
{
    "user_id": [0,1,2]
}
```

• Scenario 2: Use an Aviator function as the post-filter for an API.

The responses of the API are used as the entire input of the Aviator function. The input is in a multilayer JSON format. Example:

```
## The returned data contains three columns: user id, city, and tags.
## Three data entries that meet the conditions are returned after the API is called. Chec
k the code for rows.
  "data": {
   "totalNum": 3,
   "pageSize": 10,
    "rows": [
     {
        "user id": 2,
       "city": "BJ",
        "tags": "B,D,dongcs,ccccc"
      },
      {
        "user id": 3,
       "city": "SH",
       "tags": "A,C,D,F"
     },
      {
        "user_id": 1,
        "city": "HZ",
       "tags": "A,B,C"
     }
   ],
    "pageNum": 1
  },
  "errCode": 0,
  "requestId": "0bb211c516357674333185698eb07b",
  "errMsg": "success",
  "apiLog": null
}
```

```
ſ
```

## Use an Aviator function

Scenario 1: Use an Aviator function as the prefilter for an API. Examples:

- Example 1: The Aviator function is called to change the value of the request parameter that is passed in from the client.
  - $\circ~$  Aviator function:

```
## The original value that is passed in from the client for the request parameter user_
id is 1.
## After you use the Aviator function as the prefilter to preprocess the request parame
ter user_id, the value of the request parameter that you query is 2.
if $0.user_id == 1
{
    $0.user_id = $0.user_id + 1;
}
return $0;
```

• Input and output of the Aviator function:

```
## Input:
{
    "user_id": 1
}
## Output:
{
    "user_id": 2
}
```

- Example 2: The Aviator function is called to assign a value to the request parameter whose value is not passed in from the client. If the request parameter is optional and no value is available for the request parameter, the request parameter can be assigned a value.
  - Aviator function:

```
## user_id= indicates that the request parameter user_id is not specified on the client
.
## After you use the Aviator function as the prefilter to preprocess the request parame
ter user_id, the value of the request parameter that you query is 1.
$0.user_id = 1;
return $0;
```

• Input and output of the Aviator function:

```
## Input:
{
    "user_id": ""
}
## Output:
{
    "user_id": 1
}
```

- Example 3: The Aviator function is called to traverse all values in the array that is passed in from the client for the request parameter user\_id and change the values.
  - $\circ~$  Aviator function:

```
## The original values that are passed in from the client for the request parameter use
r_id is 0, 1, and 2.
## After you use the Aviator function as the prefilter to preprocess the request parame
ter user_id, the values of the request parameter that you query are 1, 2, and 3.
user_id = $0.user_id;
for index in user_id
{
    user_id[index] = index + 1;
}
return $0;
```

• Input and output of the Aviator function:

```
## Input:
{
    "user_id": [0,1,2]
}
## Output:
{
    "user_id": [
    1,
    2,
    3
  ]
}
```

Scenario 2: Use an Aviator function as the post-filter for an API. Examples:

- Example 1: The Aviator function is called to modify a specific data entry in the JSON data that is returned by the API.
  - Aviator function:

```
## Obtain and traverse the innermost array that is named rows in the JSON data to find
the data entry in which city is BJ. Then, change "tags":"" to "tags":"DataService Studi
o" for the data entry.
rows = $0.data.rows;
for row in rows
{
    if(row.city == "BJ")
    {
        row.tags = "DataService Studio";
    }
}
return $0;
```

• Input and output of the Aviator function:

```
## Input:
{
 "data": {
    "totalNum": 3,
   "pageSize": 10,
    "rows": [
      {
        "user id": 2,
       "city": "BJ",
        "tags": "DataService Studio"
      },
      {
        "user_id": 1,
       "city": "HZ",
        "tags": "A,B,C"
      },
      {
        "user id": 3,
        "city": "SH",
```

```
"tags": "A,C,D,F"
    }
   ],
   "pageNum": 1
 },
 "errCode": 0,
 "requestId": "0bb211e016357705412263266e571e",
 "errMsg": "success",
 "apiLog": null
}
## Output:
{
 "data": {
   "totalNum": 3,
   "pageSize": 10,
   "rows": [
     {
       "user_id": 2,
       "city": "BJ",
       "tags": "DataService Studio"
     },
      {
       "user_id": 1,
       "city": "HZ",
       "tags": "A,B,C"
      },
      {
       "user id": 3,
       "city": "SH",
       "tags": "A,C,D,F"
     }
   ],
   "pageNum": 1
 },
 "errCode": 0,
 "requestId": "0bb211e016357705412263266e571e",
  "errMsg": "success"
}
```

- Example 2: The Aviator function is called to add a constant column to each data entry in the JSON data that is returned by the API.
  - Aviator function:

```
## Obtain and traverse the innermost array that is named rows in the JSON data and add
the constant column is_target= "Y" to each data entry.
rows = $0.data.rows;
for row in rows
{
    row.is_target= "Y";
}
return $0;
```

• Input and output of the Aviator function:

```
## input:
{
  "data": {
   "totalNum": 3,
   "pageSize": 10,
   "rows": [
     {
       "user_id": 2,
       "city": "BJ",
       "tags": "DataService Studio"
      },
      {
       "user id": 1,
       "city": "HZ",
       "tags": "A,B,C"
      },
      {
       "user_id": 3,
       "city": "SH",
       "tags": "A,C,D,F"
     }
   ],
    "pageNum": 1
 },
 "errCode": 0,
 "requestId": "0bb211e016357705412263266e571e",
 "errMsg": "success",
  "apiLog": null
}
## Output:
{
  "data": {
   "totalNum": 3,
   "pageSize": 10,
    "rows": [
     {
       "is_target": "Y",
       "user_id": 2,
       "city": "BJ",
        "tags": "DataService Studio"
      },
      {
       "is_target": "Y",
       "user id": 1,
       "city": "HZ",
       "tags": "A,B,C"
      },
      {
       "is_target": "Y",
       "user id": 3,
       "city": "SH",
        "tags": "A,C,D,F"
      }
    ],
    "חסמפאשייים 1
```

}

```
pageNum . 1
},
"errCode": 0,
"requestId": "0bb211e016357705412263266e571e",
"errMsg": "success"
```

- Example 3: The JSON data that is returned by the API contains numerous fields, and you do not want to determine the return values of the fields one by one. The Aviator function is called to traverse all fields and change the values of the fields that have special values.
  - Aviator function:

```
## Obtain the innermost array that is named rows in the JSON data. Traverse the nested
data in the rows array to obtain the map of each data entry. Then, change the value of
the map whose value is null to test.
## In Aviator syntax, nil specifies a null constant, which is equivalent to null in oth
er syntax.
rows = $0.data.rows;
for row in rows
{
   for index in row
    {
       if(index.value == nil)
        {
           index.value = "test"
        }
    }
}
return $0;
```

• Input and output of the Aviator function:

```
## Input:
{
 "data": {
   "totalNum": 3,
    "pageSize": 10,
    "rows": [
     {
       "user id": null,
       "city": "SH",
        "tags": "A,C,D,F"
      },
       "user_id": 2,
        "city": "BJ",
        "tags": "DataService Studio"
      },
      {
        "user id": null,
       "city": null,
       "tags": "A,B,C"
      }
    ],
    "pageNum": 1
```

```
},
  "errCode": 0,
 "requestId": "0bb211f016372870359913841e52d8",
 "errMsg": "success",
 "apiLog": null
}
## Output:
{
  "data": {
   "totalNum": 3,
   "pageSize": 10,
   "rows": [
     {
       "user_id": "test",
       "city": "SH",
       "tags": "A,C,D,F"
      },
      {
       "user id": 2,
       "city": "BJ",
       "tags": "DataService Studio"
      },
      {
       "user id": "test",
       "city": "test",
       "tags": "A,B,C"
     }
   ],
    "pageNum": 1
  },
  "errCode": 0,
  "requestId": "0bb211f016372870359913841e52d8",
  "errMsg": "success"
}
```

## Filter use effect

(?) Note You can use only a published function as a filter for an API.

Input and output of the API:

```
## API input:
{
"user_id": ""
}
## API output:
{
 "data": {
  "totalNum": 1,
   "pageSize": 10,
   "rows": [
    {
       "is_target": "Y",
       "user_id": 2,
       "city": "BJ",
      "tags": "DataService Studio"
    }
   ],
   "pageNum": 1
 },
 "errCode": 0,
 "requestId": "0be30a8716357721257118118e2b0f",
 "errMsg": "success"
}
```

# 6.Service Orchestration 6.1. Use service orchestration

The service orchestration feature of DataService Studio allows you to configure workflows by dragging nodes to directed acyclic graphs (DAGs). You can arrange APIs and functions in a serial, parallel, or branch structure based on the business logic.

## Prerequisites

- DataWorks Enterprise Edition or a more advanced edition is activated so that you can use the service orchestration feature. For more information, see Billing of DataWorks advanced editions.
- A DataWorks workspace is created in the China (Shanghai) region.

## Context

When you run a workflow to call APIs, DataWorks runs the nodes in the workflow in sequence, passes parameters among the nodes, and automatically changes the status of each node. The service orchestration feature simplifies the process of calling multiple APIs or functions and reduces the cost of development and maintenance. This allows you to focus on business development.

The service orchestration feature provides the following benefits:

• Reduced cost of developing APIs

After you drag nodes to a DAG, you can arrange APIs and functions in a serial, parallel, or branch structure without the need to write code. This reduces the cost of developing APIs.

• Higher performance in calling APIs and functions

A workflow allows you to call multiple APIs and functions in a container. Compared with writing code to call APIs and functions, the service orchestration feature reduces the latency of calling APIs and functions and greatly improves the calling performance.

• Serverless architecture

The service orchestration feature adopts a serverless architecture. A serverless architecture supports automatic resource scaling based on business needs. You can focus on the business logic, without the need to worry about the underlying runtime environment.

## Values of request and response parameters

DataService Studio uses JSONPath to obtain parameter values. JSONPath is a query language that allows you to extract data from JSON files. For more information, see JSONPath.

For example, three nodes are run in the following order: A, B, and then C. Node C needs to use the response parameters of Node A and Node B.

• Response parameter of Node A: {"namea":"valuea"}

Expression for obtaining the value of the response parameter of Node A: \${A.namea}

• Response parameter of Node B: {"nameb":"valueb"}

Expression for obtaining the value of the response parameter of Node B: \$.nameb or \${B.nameb}

The built-in start node provides request parameters for the whole workflow. Assume that a request parameter of a workflow is {"namewf":"valuewf"}. All nodes of the workflow can obtain the value of the request parameter by using the \${START.namewf} expression.

**?** Note The start node and the end node are built-in nodes of a workflow. You can rename the nodes, but cannot delete them. The start node is equivalent to Node 0 of the current workflow.

## Parameters

• Request parameters of a workflow

On the configuration tab of a workflow, click the **Request Parameters** tab in the right-side navigation pane. Then, you can configure request parameters in manual adding or automatic parsing mode.

- Manual adding: Click Add Parameters and manually add a request parameter for the workflow.
- Automatic parsing: If the first node of the workflow is an API node, click Automatically parse request parameters to automatically map the request parameters of this API node to the request parameters of the workflow.
- Request parameters of an API node

Click an API node. In the panel that appears, click **Input Request Parameters** and specify values for request parameters.

 If you do not specify a value for a request parameter, DataService Studio obtains the value of the same parameter in the first layer of the JSON string that is returned by the parent node, and assigns the value to the request parameter.

Onte If the current node is the first node of a workflow, the values of the request parameters of this node are assigned to the same parameters of the workflow.

• If you specify a value for a request parameter, DataService Studio uses the value that you specify.

**?** Note To reference the value of a specified parameter that is returned by a specified ancestor node, you must use a JSONPath expression.

• Response parameters of an API node

Click an API node. In the panel that appears, select **set output results** and customize the output of the node by using JSON expressions similar to the following example:

```
{
   "return1":"$.data.rows.user_id",
   "return2":"$.data.rows.user_name"
}
```

• Request parameters of a Python node

Click a Python node. Specify request parameters in the Request Parameters section.

• Response parameters of a Python node

Click a Python node. In the panel that appears, select **set output results** and customize the output of the node by using JSON expressions similar to the following example:

```
{
   "return1":"$.data.rows.user_id",
   "return2":"$.data.rows.user_name"
}
```

The following table describes common JSONPath expressions that are used to obtain parameter values.

JSONPath expression	Role in request parameters Role in response parameters	
\$.	Obtains the root object of the output of the parent node.	Obtains the root object of the output of the current node.
\$.param	Obtains the value of the param parameter in the output of the parent node.	Obtains the value of the param parameter in the output of the current node.
\${START}	Obtains the output of the start nod	e.
\${NodeID}	Obtains the output of the node with the specified ID.	
\${NodelD.param}	Obtains the value of the param parameter in the output of the node with the specified ID.	

## Example

You must first create a connection. For more information, see Configure a data source. In this example, a MySQL connection is used.

1.

2. Register an API.

In this example, create an API by using the registration method.

i. On the Service Development tab, move the pointer over the 📑 icon and select Register API.

You can also click the required business process, right-click **API**, and then choose **New** > **Regist er API**.

- ii. In the **Register API** dialog box, set the parameters as required. For more information, see Register an API.
- iii. Click OK.
- 3. Register a function.
  - i. On the Service Development tab, move the pointer over the picon and choose Function >

#### Create Python Function.

You can also click the required business process, right-click **Function**, and then choose **New > Create Python Function**.

- ii. In the **Create Python Function** dialog box, set the parameters as required. For more information, see Create and manage functions.
- iii. Click OK.

iv. On the configuration tab of the function, enter the following code in the Edit Code section:

```
# -*- coding: utf-8 -*-
# event (str) : in filter it is the API result, in other cases, it is your param
# context : some environment information, temporarily useless
# import module limit: json,time,random,pickle,re,math
import json
def handler(event,context):
    # load str to json object
    obj = json.loads(event)
    # add your code here
    # end add
    return obj
```

- v. In the **Environment Configuration** section, set the **Memory** and **Function Timeout** parameters.
- vi. Click the 🔳 icon in the toolbar.
- 4. Create a workflow.
  - i. On the Service Development tab, right-click the required business process and choose New > Service Orchestration.

You can also click the required business process, right-click **Service Orchestration**, and then select **Service Orchestration**.

ii. In the Service Orchestration dialog box, set the parameters as required.

Service Orchestra	tion	×
* API Name :	0/50	
	An API name must start with a letter or Chinese character, and can contain numbers, letters, Chinese characters, and underscores (_). The name must be 4 to 50 characters in length.	
* API Path :	0/200	
	A path must start with a forward slash (/), and can contain letters, numbers, underscores (_), and hyphens (-), for example, /user. The path can be up to 200 characters in length.	
* Protocol :	V HTTP V HTTPS	
* Request Method :	GET V	
* Response Content	JSON	
Туре :		
* Visible Range :	Work Space 🗸 💿	
Label :	Please Select V	
	Up to 0-5 tags can be set up, Chinese characters, English, numbers, underline, and no more than 20 characters per label	
* Description :		
	0/2000	
* Target Folder -	Dlaace Select 🗸 🗸 🗸	
	OK	

Parameter	Description	
API Name	The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.	
	The path for storing the API, such as <i>/user</i> .	
API Path	<b>Note</b> The path can be up to 200 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a forward slash (/).	
	Valid values: HTTP and HTTPS.	
Protocol	If you need to call the API by using HTTPS, you must bind an independent domain name to the API in the API Gateway console after the API is published to API Gateway. In addition, you must upload a Secure Sockets Layer (SSL) certificate in the API Gateway console. For more information, see Enable HTTPS for an API operation.	
Request Method	Valid values: GET and POST.	
Response Content Type	Set the value to JSON.	
Visible Range	<ul> <li>Valid values: Work Space and Private.</li> <li>Work Space: The API is visible to all members in the current workspace.</li> <li>Private: The API is visible only to its owner and permissions on the API cannot be granted to other users.</li> </ul>	
5	<b>Note</b> If you set the Visible Range parameter to Private, the API is visible only to you in the API list. Other members in the workspace cannot view the API in the API list.	
	Select tags from the Label drop-down list.	
Label	<b>Note</b> A tag can be up to 20 characters in length and can contain letters, digits, and underscores (_). You can set at most five tags for a workflow.	
Description	The description of the API. The description can be up to 2,000 characters in length.	
Target Folder	The folder for storing the workflow.	

- iii. Click OK.
- 5. Configure the workflow.

i. On the configuration tab of the workflow, drag nodes to the DAG and connect the nodes, as shown in the following figure.

କ୍ କ୍ <b>ଲ</b>		
✓ Service Node		
API API		
Ja JAVA		
PY PYTHON	API1	
<ul> <li>Control Node</li> </ul>	· · · · · · · · · · · · · · · · · · ·	
🚴 switch		
388 onnon	ுக் switch1	
	PYTHON1	

ii. Click the API1 node. In the panel that appears, select the API that you registered earlier from the Select API drop-down list, select set output results, and then enter {"user\_id":"\$.da ta[0].id"}.

API1	Properties
Node ID :	ties
	Red
* Select API :	Request
······································	
Request Parameters :	ame
Input Request Parameters	Parameters
🗸 set output results 🕐 :	꼬
{["user_id":"\$.data[0].id"]}	esp
	Response

Use JSONPath expressions to set response parameters. The syntax for obtaining the value of a parameter is  $\{NodeA.namea\}$ , which is the same as that for setting request parameters. {" u ser\_id":"\$.data[0].id"} assigns the value of the id parameter of the first element in the data array to the user\_id parameter. Then, the API1 node returns {"user\_id":"value"} in JSON format.

- iii. Click the **PYT HON1** node. In the panel that appears, select the function that you registered earlier from the **Select Function** drop-down list.
- iv. Click the SWITCH1 node. In the panel that appears, click Set branch conditions.

You can enter conditional expressions based on the response parameter of the parent node. For example, you can enter expressions in the \${Node ID. Parameter}>1 Or \$. Parameter>1 format. Conditional expressions support the following operators: ==, !=, >=, ><=, <, &&, !, (), +, -, \*, /, and %.

In this example, the user\_id parameter is the response parameter of the API1 node and is used as the request parameter of the SWITCH1 node.

```
Branch node 1: $.user_id != 1, indicating that Branch node 1 is run if the value of
the user_id parameter is not 1.
Branch node 2: $.user_id == 1, indicating that Branch node 2 is run if the value of
the user_id parameter is 1.
```

- v. Click the **The end** node, and then click the **Response Parameters** tab in the right-side navigation pane to configure the response parameters.
- 6. Test the workflow.
  - i. Click **Test** in the upper-right corner.
  - ii. In the Test APIs dialog box, click Determine.
  - iii. View the operational logs and execution results on the **Operation Log** and **Execution results** tabs in the lower part of the configuration tab.

# 6.2. Manage workflows

This topic describes how to view, clone, delete, and move a workflow.

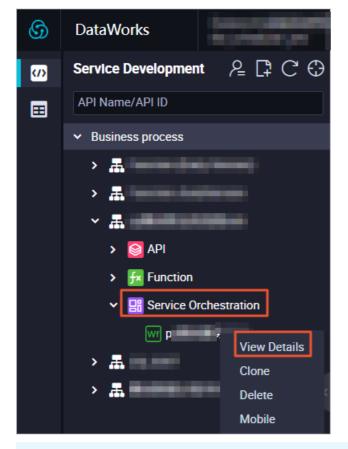
## Prerequisites

Workflows are created and published. For more information, see Use workflows.

## View a workflow

1.

- 2. On the **Service Development** tab, click the business process to which the target workflow belongs.
- 3. Right-click the target workflow and select View Details.



**?** Note The View Details option appears only in the shortcut menu of a workflow that has been published. If a workflow has not been published, double-click the workflow to go to the configuration tab of the workflow. Then, click **Properties** in the right-side navigation pane to view its basic information.

## Clone a workflow

- 1. On the **Service Development** tab, click the business process to which the target workflow belongs.
- 2. Right-click the target workflow and select Clone.

3. In the Clone API dialog box, set the parameters as required.

Clone API	>	<
* API Name :	<b>13/50</b>	
numbe	name must start with a letter or Chinese character, and can contain rs, letters, Chinese characters, and underscores (_). The name must be 4 haracters in length.	
* API Path : 🖊	_сору 26/200	
unders	must start with a forward slash (/), and can contain letters, numbers, cores (_), and hyphens (-), for example, /user. The path can be up to 200 ters in length.	
* Target Folder : Please	e Select 🗸	
	OK Cancel	
Parameter	Description	
API Name	The name of the cloned workflow. The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.	
API Path	The path for storing the cloned workflow, such as <i>/user</i> . The path can be up to 200 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a forward slash (/).	
Target Folder	The folder for storing the API to be arranged in the workflow.	

4. Click OK.

## Delete a workflow

You can delete only workflows that have not been published. To delete workflows that have been published, you must unpublish them first.

1. Optional. Unpublish the target workflow.

If the workflow to be deleted is in the Unpublished state, skip this step.

i. On the Service Development tab, click Service Management in the upper-right corner. The Manage APIs page appears by default.

G DataWorks	ALC: NOT THE OWNER.	~				Service Development	Service Manageme	nt
=	APIs of Published Aut	norized to Use Authorize Others to Use						
Manage APIs	API Name V Pleas	a lonut	٩	Clear C Refresh Ac				
API Statistics								
Test APIs								Publis
🗧 Cell APIs		Sector Sector		-	Service Or	Work Space	-	-

- ii. On the APIs of Published tab, find the target API and click Unpublish in the Actions column.
- iii. In the Unpublish API message, click OK.
- iv. Click Service Development in the upper-right corner to return to the Service Development tab.
- 2. On the **Service Development** tab, click the business process to which the target workflow belongs.
- 3. Right-click the target workflow and select Delete.
- 4. In the Delete API message, click OK.

Onte Deleted workflows cannot be recovered. Use caution when you delete a workflow.

## Move a workflow to another folder

You can move only workflows that have not been published. To move workflows that have been published, you must unpublish them first.

1. Optional. Unpublish the target workflow.

If the workflow to be deleted is in the Unpublished state, skip this step.

i. On the Service Development tab, click Service Management in the upper-right corner. The Manage APIs page appears by default.

6	DataWorks	Contraction of the local division of the loc	~				Service Development	iervice Managemen	
		APIs of Published Auth	orized to Use Authorize Others to Use						
٥	Manage APIs	API Name V Please	: Input	٩	Clear C Refresh Ac				
•	API Statistics								
0	Test APIs								
-	Call APIs		and the second second			Service Or	Work Space		

- ii. On the APIs of Published tab, find the target API and click Unpublish in the Actions column.
- iii. In the Unpublish API message, click OK.
- iv. Click Service Development in the upper-right corner to return to the Service Development tab.
- 2. On the **Service Development** tab, click the business process to which the target workflow belongs.
- 3. Right-click the target workflow and select **Move**.
- 4. In the Modify file path dialog box, set the Target Folder parameter.
- 5. Click OK.

Notice After you move a workflow, the group to which the workflow belongs may change. The endpoint for calling this workflow after it is published may also change. Use caution when you move a workflow to another folder.

# 7.Appendix: DataService Studio error codes

After DataService Studio receives an API request, it returns a response that contains an error code. You can locate and troubleshoot issues based on the error code. This topic describes common error codes that are returned by DataService Studio.

Error code	Error message	Description				
0	SUCCESS	The error message returned because the data query has succeeded.				
1108110583	QUERY_TIMEOUT	The error message returned because the query has timed out. The timeout occurs because the total running duration of the API in DataService Studio and the database exceeds the timeout period that is configured for the API.				
1108110519	PARAM_MISS	The error message returned because some required request parameters are not specified.				
1108110584	API_CONTEXT_FAILED	The error message returned because the system has failed to obtain context information based on the third party. The information includes the connection information of the data source, AccessKey information of the data source, and tenant information.				
1108110622	DAT ASOURCE_QUERY_ER ROR	The error message returned because the system has failed to query the data source. The failure may occur because the SQL syntax is invalid, the data source does not respond within the configured timeout period (10s), or the number of connections to the data source exceeds the upper limit.				
1108110703	DAT ABASE_CONNECTION _ERROR	The error message returned because the data source has failed to be connected.				
Other error codes	Other error messages	If the response contains an error code other than the preceding error codes, you can join the DataWorks DingTalk group for technical support.				