

ROMA Connect

User Guide (New Version)

Issue 01
Date 2024-11-28



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Instance Management	1
1.1 Creating a ROMA Connect Instance	1
1.1.1 Preparing Resources	1
1.1.2 Creating an Instance	2
1.2 Buying a Package	5
1.3 Viewing Details of an Instance	6
1.4 Modifying Instance Specifications	8
2 Creating a Composite Application	10
2.1 Creating a Composite Application with a Template	10
2.2 Composite Application Template	11
2.3 ROMA Snap	13
3 Composite Application Management	16
3.1 Viewing Runtime Logs	16
3.2 Viewing System Logs	17
3.3 Viewing Monitoring Data	18
3.4 Importing and Exporting a Composite Application	18
3.5 Creating a Composite Application Template	19
3.6 Referencing Variables	19
3.7 Referencing Functions	22
4 Connectors	27
4.1 Triggers	27
4.1.1 OpenAPI	27
4.1.2 FTP	29
4.1.3 ActiveMQ	30
4.1.4 Jira Software	31
4.1.5 Timer	33
4.1.6 ArtemisMQ	36
4.1.7 RabbitMQ	37
4.1.8 EventGrid	38
4.1.9 QQ Mail	39
4.1.10 IBM MQ	40
4.1.11 Kafka	42

4.1.12 MySQL.....	43
4.1.13 Oracle.....	45
4.1.14 PostgreSQL.....	47
4.1.15 SQL Server.....	48
4.2 HTTP.....	50
4.2.1 HTTP Request.....	50
4.2.2 HTTP Response.....	51
4.3 Databases.....	52
4.3.1 MySQL.....	52
4.3.2 Redis.....	55
4.3.3 Db2.....	57
4.3.4 GaussDB.....	60
4.3.5 DWS.....	62
4.3.6 Milvus.....	64
4.3.7 Oracle.....	66
4.3.8 PostgreSQL.....	70
4.3.9 SQL Server.....	74
4.4 Huawei Cloud Services.....	77
4.4.1 FunctionGraph.....	78
4.4.2 SMS.....	79
4.4.3 SMN.....	81
4.4.4 OBS.....	83
4.4.5 DSC.....	86
4.4.6 Image Recognition.....	95
4.4.7 OCR.....	98
4.4.8 FRS.....	99
4.4.9 CodeArts Artifact.....	100
4.4.10 CodeArts Build.....	101
4.4.11 CodeArts Check.....	105
4.4.12 CodeArts Deploy.....	106
4.4.13 CodeArts Req.....	108
4.4.14 CodeArts Pipeline.....	110
4.4.15 CodeArts Repo.....	114
4.4.16 CodeArts TestPlan.....	115
4.4.17 CodeArts IDE Online.....	124
4.4.18 CodeArts PerfTest.....	129
4.5 SaaS Applications.....	130
4.5.1 GitHub.....	130
4.5.2 WordPress.....	132
4.5.3 SAP.....	133
4.5.4 Jira Software.....	134
4.5.5 WeCom.....	139

4.5.6 DingTalk.....	151
4.5.7 Tencent Questionnaire.....	154
4.5.8 WJX Questionnaire.....	155
4.5.9 Polarion.....	159
4.6 Emails.....	162
4.6.1 QQ Mail.....	162
4.6.2 163 Mail.....	164
4.7 Others.....	167
4.7.1 FTP.....	167
4.8 Message Systems.....	169
4.8.1 ActiveMQ.....	169
4.8.2 ArtemisMQ.....	170
4.8.3 IBM MQ.....	171
4.8.4 Kafka.....	172
4.9 Custom Connectors.....	174
4.9.1 OpenAPI.....	174
5 Processors.....	178
5.1 Conditions.....	178
5.2 Parallel Processor.....	180
5.3 Delay.....	182
5.4 EDI Processor.....	183
5.5 Variable Assignment.....	184
5.6 Variable Updater.....	185
5.7 Sorter.....	186
5.8 Data Conversion.....	186
5.9 Splitter.....	187
5.10 Filter.....	188
5.11 Script.....	189
5.12 Message Logger.....	190
5.13 Data Mapper.....	191
5.14 Iterator.....	199
5.15 Error Monitoring.....	200
5.16 End.....	201
5.17 Symmetric Encrypt/Decrypt.....	202
5.18 Asymmetric Encrypt/Decrypt.....	204
5.19 XML Verification.....	206
6 Debugging a Task Flow Online.....	208
7 Asset Catalog.....	211
8 Related Concepts.....	212
9 Appendix.....	213

9.1 Accessing the Internet from an Instance.....213

1 Instance Management

- [1.1 Creating a ROMA Connect Instance](#)
- [1.2 Buying a Package](#)
- [1.3 Viewing Details of an Instance](#)
- [1.4 Modifying Instance Specifications](#)

1.1 Creating a ROMA Connect Instance

1.1.1 Preparing Resources

Before creating a ROMA Connect instance, you need to ensure availability of resources, including a virtual private cloud (VPC) and subnet. Each ROMA Connect instance is deployed in a VPC and bound to a subnet. In this way, ROMA Connect provides an isolated virtual network environment that can be easily configured and managed by users.

Required Resources

The following table lists the resources required by a ROMA Connect instance.

Table 1-1 Resources

Resource	Requirements	Operation
VPC and subnet	<p>Different ROMA Connect instances can use the same or different VPCs and subnets based on site requirements. Note the following when creating a VPC and subnet:</p> <ul style="list-style-type: none"> • The created VPC and ROMA Connect must be in the same region. • Retain the default settings unless otherwise specified. 	<p>For details on how to create a VPC and subnet, see Creating a VPC. If you need to create and use a new subnet in an existing VPC, see Creating a Subnet for the VPC.</p>

1.1.2 Creating an Instance

Before using ROMA Connect, you need to create a ROMA Connect instance. A ROMA Connect instance is an independent resource space. Resources of different instances are isolated from each other. You can use one or more instances as required.

NOTE

If you delete an instance, all resource data created in the instance will be deleted. Exercise caution when performing this operation. Before deleting an instance, back up data by [exporting asset data of the instance](#).

Prerequisites

A VPC is available and a subnet has been configured. Otherwise, create a VPC and subnet by referring to [1.1.1 Preparing Resources](#).

Procedure

1. Go to the [Buy ROMA Connect Instance](#) page.
2. On the Buy ROMA Connect Instance page, configure basic instance information and click **Next**.

Table 1-2 Basic configurations

Parameter	Description
Region	Select the region where the instance resides. Instances in different regions cannot communicate with each other. Select the nearest region to reduce network latency.

Parameter	Description
AZ	Select the availability zone (AZ) where the instance resides. Different AZs are physically isolated but can communicate with each other via a private network. <ul style="list-style-type: none"> To enhance ROMA Connect availability, create instances in different AZs. To shorten network latency, create instances in the same AZ.
CPU Architecture	Currently, only x86 is supported.
Instance Name	Customize the instance name.
Description	Describe the instance.
RCU Allocation	Allocate RCUs to integration capabilities. Only integration capabilities with RCUs are available. <ul style="list-style-type: none"> The ROMA compute unit (RCU) can be allocated to different integration capabilities. More RCUs indicate higher capability performance. You can use the RCU values recommended by the system or customize them for different capabilities. The total RCU value cannot be 0. Some integration capabilities depend on each other. <ul style="list-style-type: none"> Composed Application (Flow running) and Service Integration (Data API/Function API) are dependent on Service Integration (API management). Device Integration is dependent on Message Integration.

- On the **Advanced Configuration** page displayed, complete the information and click **Next**.

Table 1-3 Advanced Configuration

Parameter	Description
Network Configuration	
VPC	Select the VPC and subnet associated with the instance. The VPC and subnet must have been created in 1.1.1 Preparing Resources .
Accessing Internet	Whether to enable the function of accessing ROMA Connect from a public network based on service requirements.
Public Network Bandwidth	Mandatory only when Accessing Internet is enabled. Select the bandwidth of the public network.

Parameter	Description
Message Integration Configuration (Displayed only when RCUs are allocated to Message Integration .)	
Engine Version	Version number of the Kafka server for message integration. Currently, version 2.7 is supported.
MQS Capacity Threshold Policy	<p>Select a processing policy that will be used when the number of messages in MQS reaches the capacity threshold (95% of the storage space).</p> <ul style="list-style-type: none"> • Stop creation: New message creation requests are rejected until messages are deleted after the configured aging time and the available storage space is greater than 5%. This policy applies when message consumption is slow or stacked messages need to be repeatedly consumed and cannot be deleted in advance. • Automatic deletion: The earliest messages (10% of the total number of messages) are automatically deleted to free up storage space for new requests. This policy has no adverse impact on message creation. It applies when message consumption is fast and there are no stacked messages.
SASL_SSL	<p>Whether to enable SASL authentication and SSL encryption for message transmission. Enabling this parameter better secures data transmission.</p> <p>Enabling SASL_SSL cannot be undone.</p>
Intra-VPC Plaintext Access	<p>Available only when SASL_SSL is enabled.</p> <p>Whether plaintext access is used in the VPC.</p> <p>If intra-VPC plaintext access is enabled, you do not need to connect to topics in the VPC via SASL. In this case, no certificate is required for connecting to topics.</p>

4. On the **Configuration confirmation** page, select **I agree to the Customer Agreement and Privacy Statement**, and click Buy Instance.

The instance creation takes about 15 to 25 minutes. After the instance is created, the instance status is **Available** in **Resource**.

If the ROMA Connect instance fails to be created, delete the failed instance and create a new one. If the creation still fails, contact technical support.

Related Operations

After the instance is created, relevant integration capabilities are available.

- For details about how to create a composite application, see [Creating a Composite Application](#).
- For details about how to use FDI, see [Data Integration Guide](#).

- For details about how to use APIC, see [Service Integration Guide](#).
- For details about how to use MQS, see [Message Integration Guide](#).
- For details about how to use LINK, see [Device Integration Guide](#).

1.2 Buying a Package

ROMA Connect supports package billing. Package RCU-hour quota is deducted first and the excess usage is billed in the pay-per-use mode. You can purchase RCU packages based on service requirements.

Constraints

- The package quota is valid only in one effective month. The remaining quota expires and cannot be transferred to the next month. The purchased package takes effect upon order confirmation. Please note that orders cannot be unsubscribed from.
- Automatic renewal has yet to be supported. RCU hours beyond the package quota are billed in pay-per-use mode.

Procedure

1. Go to the [Buy ROMA Connect Instance](#) page.
2. Configure the package details, select **I have read and agree to the preceding rules**, and click **Next**.

Table 1-4 Package configuration

Parameter	Description
Region	Select the region where the corresponding ROMA Connect instance is located.
Specification Selection	Select a package specification of the RCU-hour quota. Each specification contains a different RCU-hour quota per month. If you select 1,000 RCU Hours/Month for a five-RCU instance, you can use the instance for 200 hours in the effective month. After the 200 hours, the instance is billed in the pay-per-use mode. If there are multiple instances, the total number of RCUs are calculated.
Duration	Select the validity period (renewal period) of the package. For example, if you purchase a package for two months, you can use the package for two consecutive months. The package specification is the same for each month.
Quantity	Enter the number of packages to be purchased. If you purchase multiple packages at the same time, the packages take effect at the same time.

3. On the configuration confirmation page, select **I have read and agree to the Customer Agreement and Privacy Statement**, and click **Next** to make the payment for successful purchase.

1.3 Viewing Details of an Instance

Overview


After an instance is created, you can view and edit its basic information, configuration parameters, data dictionaries, and background tasks, and import and export assets.






For details about how to view background tasks, see [Viewing Background Tasks](#). For details about how to configure parameters, see [Modifying Instance Configuration Parameters](#). For details about data dictionaries, see [Creating a Data Dictionary](#). For details about how to import and export assets, see [Importing and Exporting Assets](#).

Procedure

- Step 1** Log in to the [new ROMA Connect console](#).
- Step 2** In the navigation pane on the left, choose **Resource**. On the page displayed, click a container instance to access the instance console.
- Step 3** On the **Instance Information** page of the console, click the **Basic Information** tab to view and edit basic information about the instance.

Table 1-5 Basic information

Item	Description
Instance	Basic information about the instance, including the instance name, instance ID, description, availability zone, and creation time. <ul style="list-style-type: none">You can modify Instance Name and Description as required.Click  next to Instance ID to copy the instance ID.
VPC	VPC associated with the instance. You can click the VPC name to view the VPC configuration.
Subnet	Subnet associated with the instance. You can click the subnet name to view the subnet configuration.
Security Group	Security group associated with the instance. You can click the security group name to view the security group configuration or click Edit to bind a new one.
Routes	Private network segments. A ROMA Connect instance can communicate with the VPC subnet segment specified during instance creation by default. To communicate with other private network segments, configure the routes.

Item	Description
Billing Information	Billing mode of the instance.
RCU Distribution Configuration	RCU values distributed to different integration capabilities.
Message Storage Space	Information about the MQS message storage space of the instance, including the storage class and space usage (percentage).
MQS Information	Basic MQS configuration information about the instance. Corresponding to the Kafka engine type. You can modify MQS Capacity Threshold Policy as required.
Connection Address	<p>Connection addresses of the instance:</p> <ul style="list-style-type: none"> ● ROMA Connect Address ● LINK Address ● APIC Address ● APIC Intranet Address (for backend and gateway components) ● MQS Intranet Address ● MQS Public Address (available only when a public access is enabled) <p>You can click  next to a connection address to copy the address.</p>
Public Network Access Entry	<ul style="list-style-type: none"> ● If the public access entry is not enabled for the instance, you can click  on the right to enable it. ● If the public network access entry is enabled, you can click  next to the EIP address to copy the address. ● If the public network access entry is enabled, you can click  next to Public Network Bandwidth to modify it. ● If the public network access entry is enabled, you can click  on the right disable it. <p>NOTICE EIPs are randomly assigned. The EIP may differ after it is rebound.</p>
Public IP Address Access Control (Whitelist)	<p>Available only when Public Network Access Entry is enabled.</p> <p>The public IP address access control is whitelist-based. If this parameter is enabled, only IP addresses in the whitelist can be accessed. Otherwise, all IP addresses can be accessed by default.</p> <p>Click Edit on the right to add or delete IP addresses.</p>

Item	Description
Private IP Address Access Control (Whitelist)	The private IP address access control is whitelist-based. If this parameter is enabled, only IP addresses in the whitelist can be accessed. Otherwise, all IP addresses can be accessed by default. Click Edit on the right to add or delete IP addresses.

----End

1.4 Modifying Instance Specifications

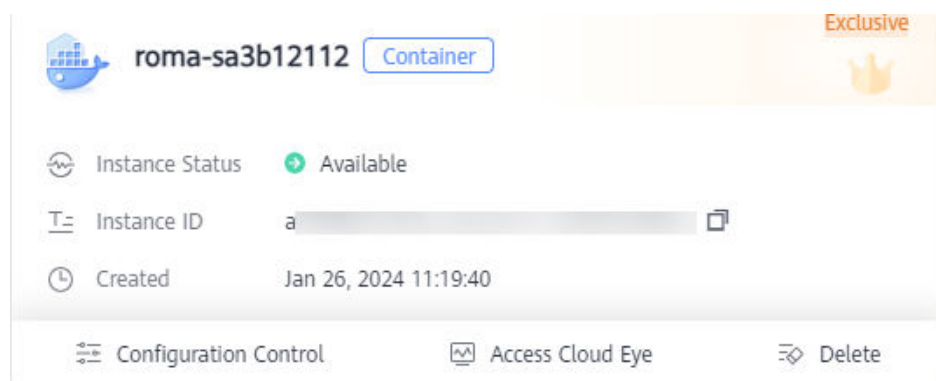
If the current instance specifications cannot meet service requirements or an integration capability needs to be enabled or disabled, you can reallocate RCUs to dynamically scale in or out the integration capability.

NOTE

- Message integration supports only scale-out. You can set RCU to **0** to disable it.
- Ensure that the integration capability is no longer used before disabling it.

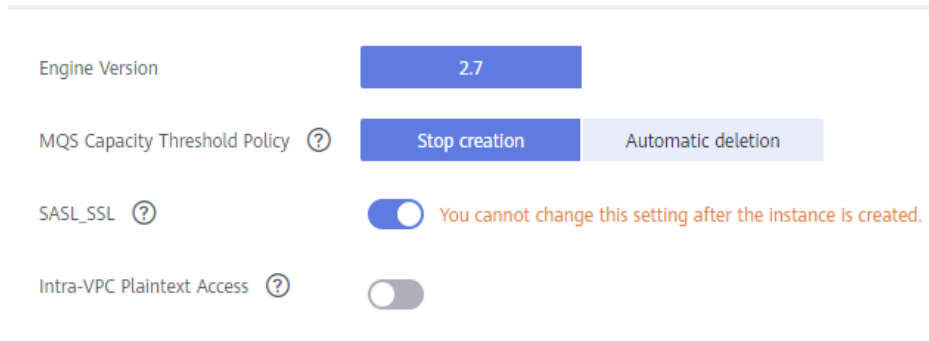
Modifying Instance Specifications

1. Log in to the [new ROMA Connect console](#).
2. In the navigation pane on the left, choose **Resource**. Hover the pointer over the instance whose RCU needs to be adjusted and click **Configuration Control**.



3. Adjust RCUs on the **Configuration Control** page. The system will recalculate the fee based on the new RCU configuration.
If RCUs are allocated to message integration for the first time, configure MQS information by referring to [Table 1-3](#). Then click **Next**.

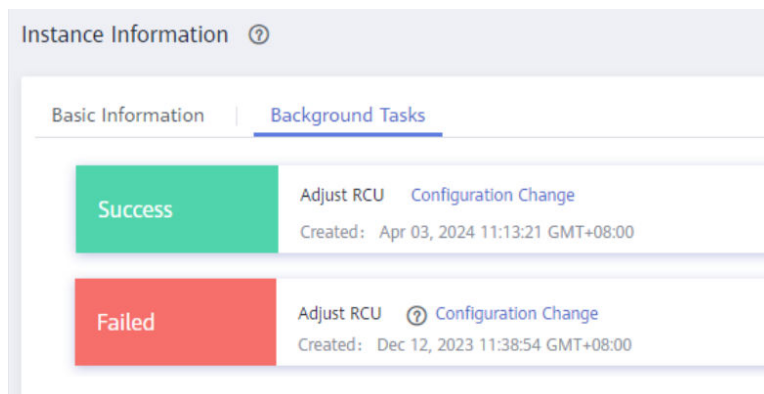
Figure 1-1 Configuring MQS



4. On the **Confirm** page, confirm the new configuration and click **Configuration Control**.
5. Wait for the instance to start to reallocate RCUs. The scale-out takes 10 to 20 minutes, and the scale-in takes about 10 minutes.
6. On the **Resource** page, click the instance that is undergoing specification changes to go to the instance console.
7. On the **Instance Information** page, click the **Background Tasks** tab and view the progress of the instance specification change task.

Viewing Background Tasks

1. On the **Instance Information** page of the console, click the **Background Tasks** tab to view the result of the **instance specification change** task. The task status can be **Success** or **Failed**.



2. Hover the pointer over **Configuration Change** to view the configurations before and after the change.

2 Creating a Composite Application

[2.1 Creating a Composite Application with a Template](#)

[2.2 Composite Application Template](#)

[2.3 ROMA Snap](#)

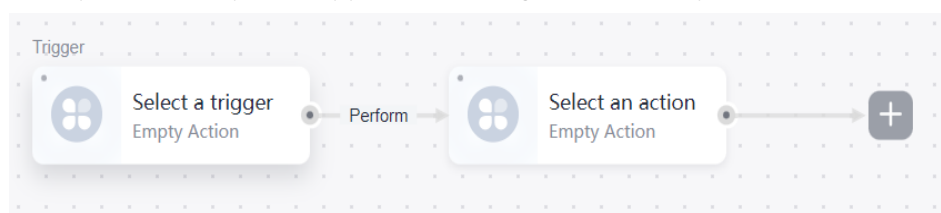
2.1 Creating a Composite Application with a Template

Composite applications are built by integrating multiple existing applications via open APIs and event channels, implementing fast service replication and innovation.

A composite application is a basic unit for running one or more flow tasks.

Procedure

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click **Create**.
3. In the **Select Template** box, you can select the blank template or an existing service template to create a composite application.
4. On the canvas page for creating a composite application, configure and orchestrate flow tasks.
 - For a composite application using an existing service template, the flow task configurations and orchestration corresponding to the service scenario are provided. Modify the settings as required.
 - For a composite application using the blank template, customize the configurations and orchestrate flow tasks as required. The following example is a composite application using a blank template.



5. Click the trigger box and action box on the page to configure and orchestrate the flow task.
 - Trigger: Only trigger connectors are supported. For details about how to configure a trigger, see [4.1 Triggers](#).
 - Action: You can select various types of connectors and processors as an action. For details about how to configure triggers and connectors, see [4 Connectors](#) and [5 Processors](#).
6. Click **Save** in the upper right corner.
7. In the dialog box displayed, enter the name and description of the composite application, and click **Yes**.

2.2 Composite Application Template

ROMA Connect offers tailored integration templates that help enterprises across various sectors—such as e-commerce, healthcare, and retail—swiftly integrate applications for specific use cases.

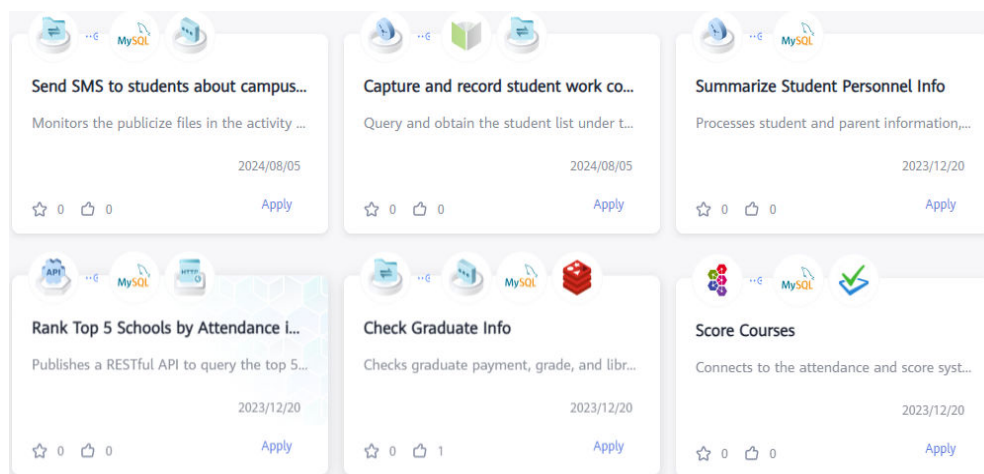
Below are examples of industry-specific integration templates.

Education

Designed for schools, educational institutions, and online learning platforms, these templates facilitate student data management, teaching resource sharing, and course management. Key functionalities include:

- **Query and record homework completion status:** Retrieve the list of students in a specified course and record their homework completion status.
- **Query student off-campus information:** Check graduate payment, grades, and library information, then send SMS notifications to students with outstanding transactions.

Figure 2-1 Templates

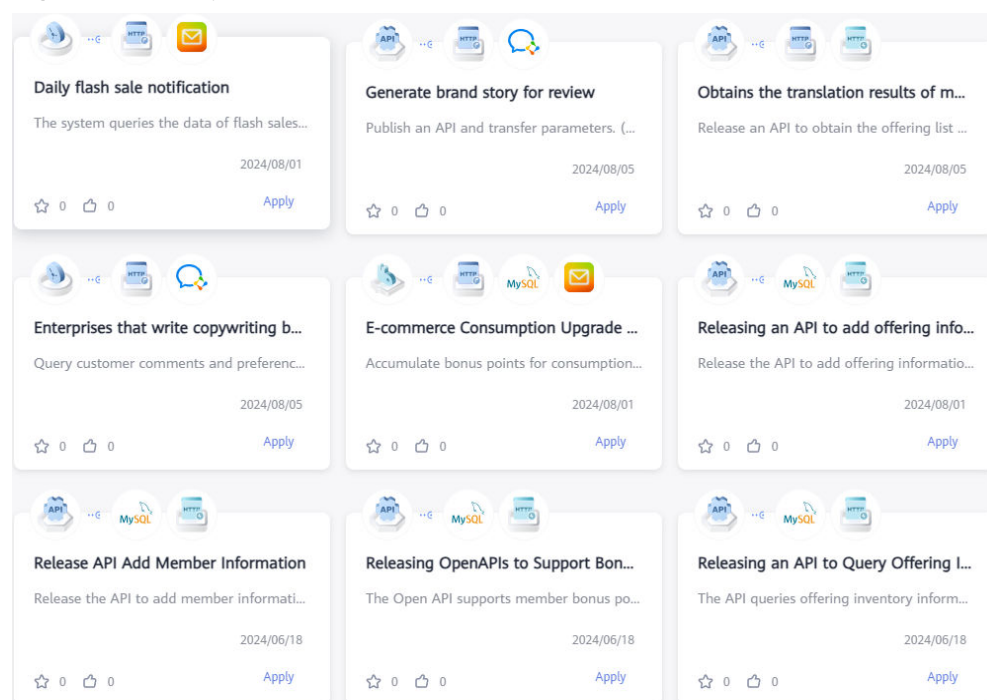


E-Commerce

The iPaaS efficiently integrates supply chain, inventory, order, customer, and payment systems, enabling e-commerce platforms to respond quickly to market demands and customer feedback. Key functionalities include:

- **Auto-welcome and notifications:** Send welcome SMS messages to new customers and notify customer managers via SMS, email, and WeCom for follow-up.
- **Inventory alerts:** Connect to the warehouse management system to monitor inventory levels based on sales volume and send alerts for low stock.
- **Automated marketing:** Generate copy based on keywords and distribute it to target customers via WeCom.

Figure 2-2 Templates

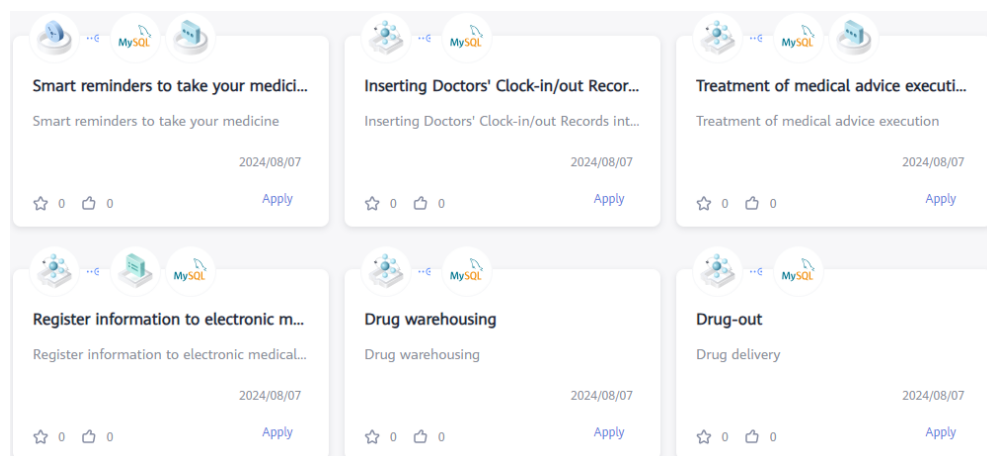


Healthcare

The iPaaS integration templates enhance data sharing among clinical and service systems (such as EMRs, patient management systems, and lab information systems) in medical institutions. Key functionalities include:

- **Pharmaceutical management integration:** Connect pharmaceutical inventory, supply chain, and distribution systems with medical management platforms for real-time accuracy.
- **Medical insurance payment integration:** Push pharmaceutical data from the medical insurance payment system to pharmacies' retail PSI management systems.
- **Pharmacy information management integration:** Track and manage orders across various pharmacy management systems.

Figure 2-3 Templates

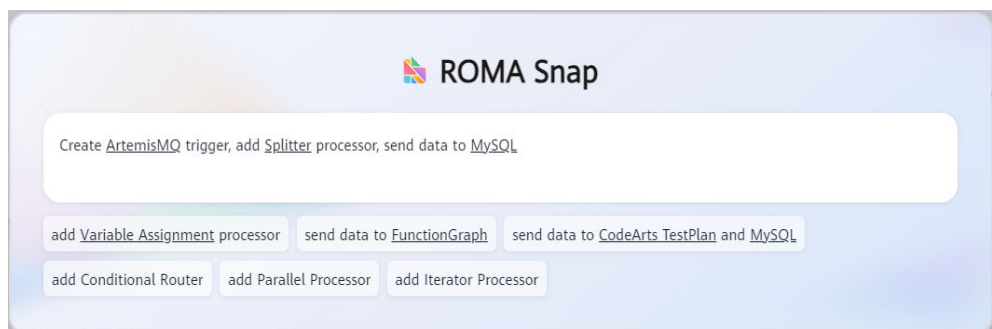


2.3 ROMA Snap

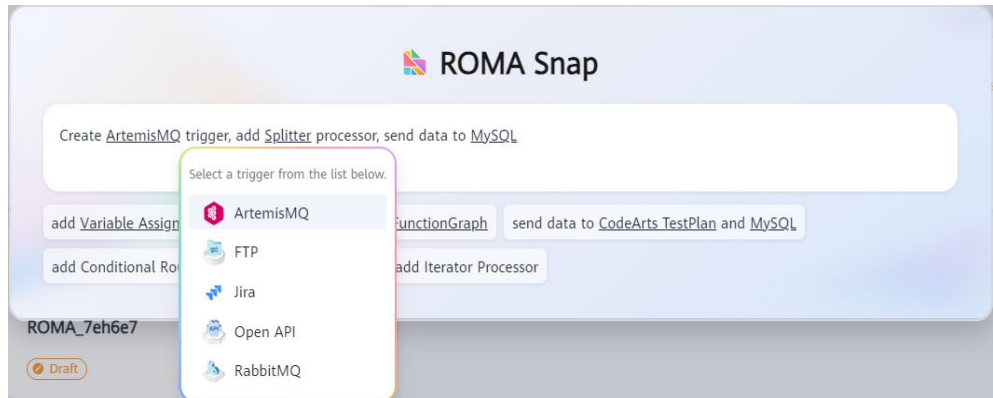
ROMA Snap captures the requirements for service integration based on user inputs by way of Natural Language Processing (NLP) machine learning, matching triggers, connectors, and data processors supported by the system to generate a composite application. You can further configure and orchestrate the generated composite application for building and deployment.

Procedure

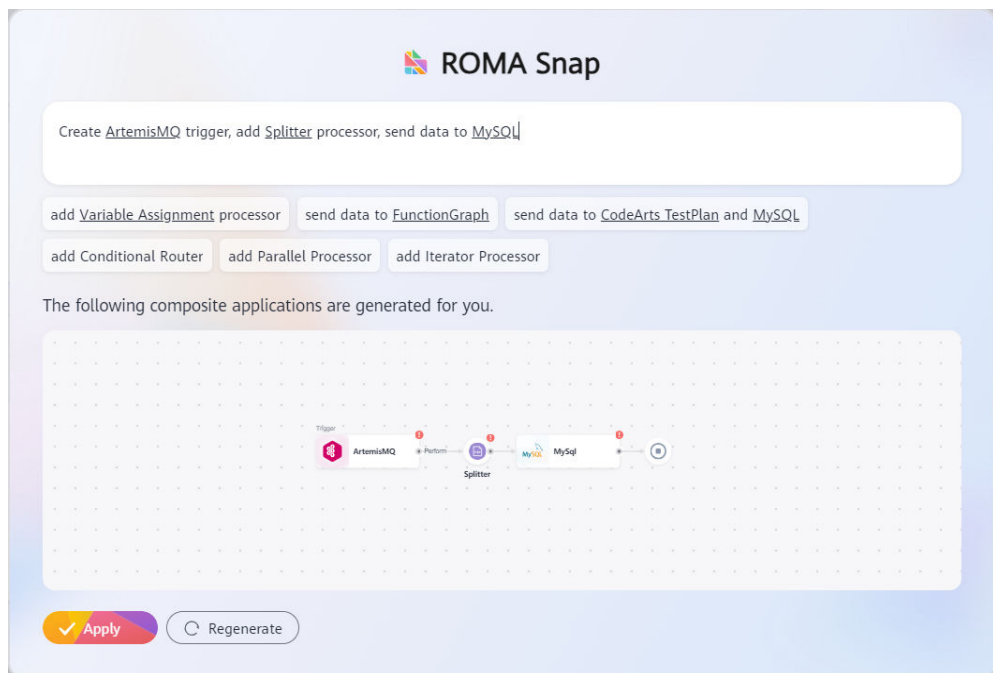
1. Log in to the new ROMA Connect console.
2. On the **Workbench** page, click **ROMA Snap**.
3. In the pop-up box, enter the description of the composite application to be created, or click the action templates under the text box.



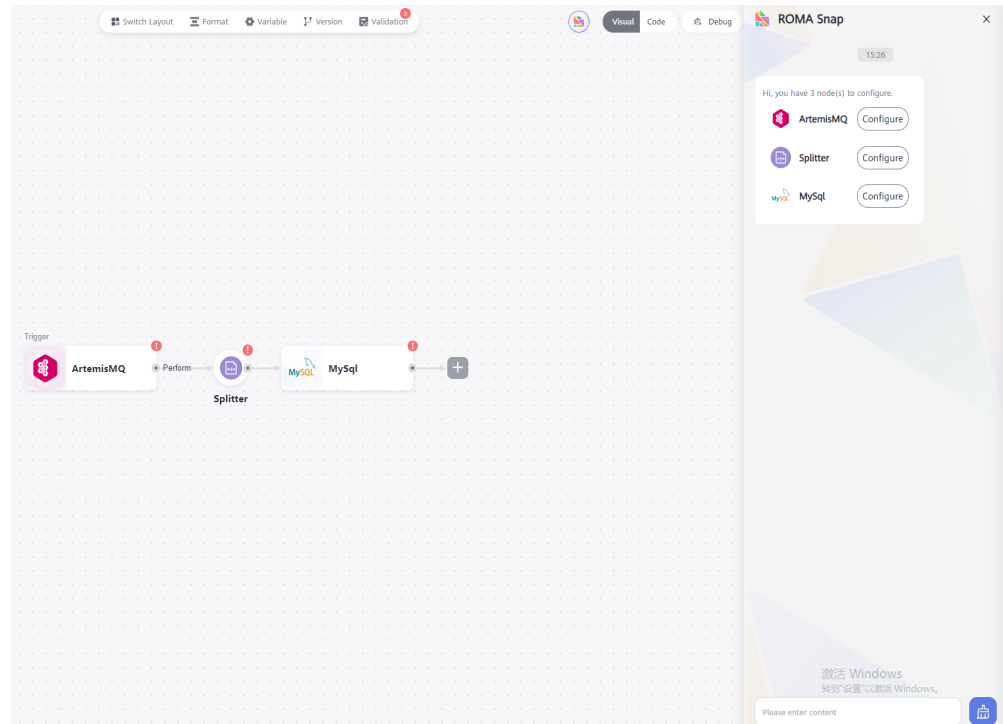
4. Click the underlined connector or processor to replace it with a desired one in the drop-down list.



5. Press **Enter**, and then click **Apply** to generate a composite application.



6. On the canvas for editing the composite application, configure and orchestrate the flow task.
You can also configure components with interactive inputs.



7. Click **Save** in the upper right corner.
8. In the upper left corner, click the edit button next to the composite application name. In the dialog box displayed, modify the name and enter a description, and click **Yes**.

3 Composite Application Management

- [3.1 Viewing Runtime Logs](#)
- [3.2 Viewing System Logs](#)
- [3.3 Viewing Monitoring Data](#)
- [3.4 Importing and Exporting a Composite Application](#)
- [3.5 Creating a Composite Application Template](#)
- [3.6 Referencing Variables](#)
- [3.7 Referencing Functions](#)

3.1 Viewing Runtime Logs

This section describes how to view runtime logs of a composite application.

Viewing Runtime Logs

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target composite application.
3. On the details page, click the **Runtime Log** tab to view the runtime logs of the flows.

Table 3-1 Parameters of runtime logs

Parameter	Description
Start Time	Time when a flow starts.
End Time	Time when a flow ends.
Run ID	Run ID of a flow.
Run Time	The duration that a flow has run.

Parameter	Description
Status	Status of a flow.

4. (Optional) Configure access logs to store exception monitoring logs.
 - a. If the composite application is not started, click **Start and configure LTS logs** and select an instance to start the application.
 - b. If the composite application is started, click **Enable**. In the **Access Logs** dialog box, configure log access information.

Table 3-2 Parameters for configuring access logs

Parameter	Description
Log Group	Select the log group to which the log stream belongs. If no log group is available, click View Log Groups to go to the LTS log management console and create a log group. For details, see
Log Stream	Select the log stream for storing the monitoring logs. If no log stream is available, click View Log Streams to go to the LTS log management console and create a log stream. For details, see .

- c. Click **OK**.

3.2 Viewing System Logs

Overview

This section describes how to view system logs of a composite application.

Viewing System Logs

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target composite application.
3. On the details page, click the **System Log** tab to view the start, stop, and deployment records of the flows.

Table 3-3 Log field description

Field	Description
Time	Log report time
Log level	Levels: Info , Error , and Debug

Field	Description
ID	Task flow ID
Details	Log content

3.3 Viewing Monitoring Data

This section describes how to view monitoring data of a composite application.

Viewing monitoring data

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target composite application.
3. On the details page, click the **Monitoring** tab.
4. View the **Runs**, **Duration**, and **Failures** data of any flow as required.

3.4 Importing and Exporting a Composite Application

ROMA Connect allows you to import and export composite applications using files.

- Importing composite applications in JSON format
- Exporting composite applications in JSON format

CAUTION


- After a composite application with custom connectors is exported, it cannot be imported to other environments.
 - The connection instance information of a connector cannot be exported.
-

Exporting a Composite Application

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target composite application.
3. Click **...** on the composite application and choose **Export** to export the composite application file in JSON format to the local host.

Importing a Composite Application



1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click **Create**.

3. In the **Select Template** box, choose **Import** under **Import from Files**.
4. On the page displayed, click  to select a composite application file in JSON format.
5. Click **Upload** to import the composite application.

3.5 Creating a Composite Application Template

Users can generate a template from a well-designed composite application. Other users can quickly create composite applications using the template.

Procedure

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target composite application.
3. Click  on the composite application and choose **Modify** to enter the canvas page.
4. Click **Save** > **Save as Template** in the upper right corner of the page.
5. In the dialog box displayed, enter the name and description of the composite application, and click **Yes**.
6. Return to the home page. In the navigation pane on the left, choose **Asset** to view the published composite application template.
 - Click a composite application template to view its details.
 - Click **Apply** on a specific template to quickly create a composite application. You can also select a template when [creating a composite application](#).
 - Click  on a template to add the template to **My Favorites**.

3.6 Referencing Variables

Orchestrating a composite application flow and configuring connectors and processors involve various inputs, outputs, and configurations, which can be referenced as variables when you edit and design a composite application to create efficiency.

Variables of composite applications are classified as system, node, and custom variables. Variables with the same name are prioritized as follows: system variable > node variable > custom variable.

System Variables

System variables indicate system parameters or specific outputs of some connectors. [Table 3-4](#) lists the system variables for composite applications.

System variable reference mode: $\${Variable\ name}$

Table 3-4 System variables

Variable	Description
step_execution_time	Execution duration of the previous node, in milliseconds.
response_code	Response code of the previous HTTP request node.
selected_row_count	Number of data records queried in the previous database node. Only MySQL, Gauss 200, PostgreSQL, Oracle, SQL Server, and Db2 databases are supported.
updated_row_count	Number of data records that are inserted, updated, and deleted in the previous database node. Only MySQL, Gauss 200, PostgreSQL, Oracle, SQL Server, and Db2 databases are supported.
loop_index	Previous loop node; index of the current loop; the value starts from 0.
loop_size	Total number of loops in the previous loop node.
split_index	Previous splitter node; index of the current data block during data splitting; the value starts from 0.
split_size	Number of data splits on the previous splitter node.
split_complete	Whether data splitting is complete in the previous splitter node.

Node Variables

The execution result of the previous node can be used as a node variable and referenced by subsequent nodes, such as the request body of an OpenAPI, database query result, and HTTP request result.

Node variable reference mode: **`${payload}`**

If the execution result is an object, such as a JSON or XML object, use **``${payload.id}`** or **``${payload[0].id}`** to reference certain data in the execution result.

Example:

If the response body received by an HTTP request node is in JSON format as follows, use **``${payload}`** to reference the JSON data, **``${payload.id}`** to reference the value **`00000000`**, and **``${payload.data[0].value}`** to reference the value **`x`**.

```
{
  "id": "00000000",
  "name": "sample01",
  "data": [
    {
      "id": 1,
      "value": "x"
    }
  ]
}
```

```

    "id":2,
    "value":"y"
  },
  {
    "id":3,
    "value":"z"
  }
]
}

```

Custom Variables

Custom variables can be referenced in subsequent nodes, and include the following items: variables added when configuring variables, request parameters added in the OpenAPI and the variable assignment nodes, and output variables configured in the script node.

Custom variable reference mode: **$\$(\text{variable name/parameter name})$**

- Adding variables on the canvas page
 - a. Log in to the new ROMA Connect console.
 - b. In the navigation pane on the left, choose **Composite Applications**. On the page displayed, click the target composite application.
 - c. Click ******* on the composite application and choose **Modify** to enter the canvas page.
 - d. Click **Variable** on the top of the page.
 - e. In the pop-up box displayed, click **Add**.

Table 3-5 Configure variables

Parameter	Description
Variable	Enter a variable name to be referenced.
Type	Data type of the variable: string, integer, long integer, decimal, boolean, and password.
Value	Enter the value of the variable.
Scope	Select the effective scope of the variable: global (all composite applications); current composite application.
Description	Enter a description of the variable.

- f. Click **Save**.
- Adding request parameters in the OpenAPI node
For details, see [4.1.1 OpenAPI](#).
 - Adding variables in the variable assignment node
For details, see [5.5 Variable Assignment](#).
 - Configuring output variables in the script processing node
For details, see [5.11 Script](#).

3.7 Referencing Functions

You can reference functions to quickly configure connectors when editing and designing a composite application.

Figure 3-1 describes how to use functions in composite applications.

Figure 3-1 Referencing functions



Category

Math, string, encryption and decryption, date and time, encoding and decoding, and number system conversion.

Table 3-6 Functions

Category	Function	Description
Math (2)	sum(array arr)	Returns the sum of all numbers in an array.
	avg(array arr)	Returns the average of all numbers in an array.
String (29)	uuid()	Generates a random UUID consisting of letters and digits.
	upper(string str)	Converts a string into uppercase letters.
	lower(string str)	Converts a string into lowercase letters.
	camelize(string str)	Converts words in a string into the camel case format, deletes all spaces, and converts the first letter of each word except the first word into the uppercase format.
	dasherize(string str)	Converts words in a string into the kebab case format, letters into lowercase letters, and all spaces into hyphens (-).
	capitalize(string str)	Converts the first letter of each word into the uppercase format and the other letters into the lowercase format.
	charCode(string char)	Converts characters to character encoding.

Category	Function	Description
	fromCharCode(number charCode)	Converts a character encoding into a string.
	charCodeAt(string str, number index)	Returns the character encoding at the index in a string.
	isAlpha(string str)	Returns a Boolean value that indicates whether a string contains only English letters.
	isAlphanumeric(string str)	Returns a Boolean value that indicates whether a string contains only English letters or digits.
	isNumeric(string str)	Returns a Boolean value that indicates whether a string contains only digits.
	isLowerCase(string str)	Returns a Boolean value that indicates whether a string contains only lowercase letters.
	isUpperCase(string str)	Returns a Boolean value that indicates whether a string contains only uppercase letters.
	isWhitespace(string str)	Returns a Boolean value that indicates whether a string contains only spaces.
	leftPad(string str, number offset)	If the string is shorter than the offset, pads the left side of the string with spaces.
	rightPad(string str, number offset)	If the string is shorter than the offset, pads the right side of the string with spaces.
	substringAfter(string str, string separator)	Gets the substring after the first separator.
	substringAfterLast(string str, string separator)	Gets the substring after the last separator.
	substringBefore(string str, string separator)	Gets the substring before the first separator.
	substringBeforeLast(string str, string separator)	Gets the substring before the last separator.
	unwrap(string str, string wrapper)	Returns a string without the wrapper text. The wrapper text is the value before and/or after a string.
	withMaxSize(string str, number size)	Limits the size of a string.

Category	Function	Description
	wrapIfMissing(string str, string wrapper)	If a string is not wrapped, adds the wrapper at the beginning or end. If the wrapper already exists at the beginning or end, updates only the missing end.
	wrapWith(string str, string wrapper)	Adds the wrapper at the beginning or end of a string.
	ordinalize(number num)	Converts num to an ordered character string, for example, 1st, 2nd, or 3rd.
	appendIfMissing(string str, string value)	If a string does not end with value , adds value there.
	prependIfMissing(string str, string value)	If a string does not start with value , adds value there.
	repeat(string str, number times)	Repeats the given number of times.
Encryption and decryption (4)	hash(string value, string algorithm)	Calculates the hash value using a supported algorithm. The algorithm must be MD2, MD5, SHA-1, SHA-256, SHA-384 or SHA-512.
	hmac(string value, string secret, string algorithm)	Generates a hash-based authentication code using the provided value, key, and hash function algorithm. The algorithm must be HMACSHA1, HMACSHA256, or HMACSHA512.
	encrypt(string value, string secret, string transformation)	Encrypts the value using the specified JDK and key, and converts the encryption into a readable format using Base64.
	decrypt(string value, string secret, string algorithm, string mode, string padding)	Decrypts the Base64 value using the specified JDK and key.
Date and time (20)	now()	Returns the current date and time.
	compare(string datetime1, string datetime2)	If datetime1 > datetime2, returns 1; if datetime1 < datetime2, returns -1; if datetime1 == datetime2, returns 0.
	atBeginningOfDay(string datetime)	Returns the midnight time value of a given time.
	atBeginningOfHour(string datetime)	Returns the given date and time, with minute and second as 0.
	atBeginningOfMonth(string datetime)	Returns the first day of the month, with time as midnight.

Category	Function	Description
	atBeginningOfWeek(string datetime)	Returns the first day of the current week, with time as midnight.
	atBeginningOfYear(string datetime)	Returns the first day of the year, with time as midnight.
	changeTimeZone(string datetime, string timezone)	Changes the time zone, which usually results in a change in the local date and time.
	daysBetween(string datetime1, string datetime2)	Returns the number of days between datetime1 and datetime2.
	format(string datetime, string outputFormat)	Converts a given datetime into the specified format.
	isLeapYear(string datetime)	Returns a Boolean value indicating whether datetime is a leap year.
	plus(string datetime, string period)	Adds a date to a datetime.
	minus(string datetime, string period)	Subtracts a date to a datetime.
	parse(string number datetime, string inputFormat)	Parses datetime using the input format and returns a value in the default format. If the epoch or timestamp value is used as the datetime, epoch or timestamp can be used as inputFormat .
	toLocalDate(string datetime)	Localizes a date.
	toLocalDateTime(string datetime)	Localizes a date and time.
	toLocalTime(string datetime, string format)	Localizes a time.
	today()	Returns the midnight time of the current day.
	tomorrow()	Returns the midnight time of the next day.
	yesterday()	Returns the midnight time of the previous day.

Category	Function	Description
Encoding and decoding (2)	encode(string data, string encoding="UTF-8")	Converts a string to the application/x-www-form-urlencoded format using the provided encoding scheme to obtain insecure bytes. The default encoding is UTF-8.
	decode(string data, string encoding="UTF-8")	Decodes the application/x-www-form-urlencoded string using a specific encoding scheme to determine which characters are represented by a continuous sequence in the form of %xy .
Number system conversion (9)	fromBase64(string value)	Converts a Base64 code to a value.
	toBase64(any value)	Base64-encodes a value.
	fromHex(string value)	Converts a hexadecimal number to a value.
	toHex(any value)	Converts a value to a hexadecimal number.
	fromBinary(number value)	Converts a value from binary to decimal.
	fromHex(number value)	Converts a value from hexadecimal to decimal.
	fromRadixNumber(number value, number baseAsBinary)	Converts a value to a decimal number.
	toBinary(number value)	Converts a value from decimal to binary.
	toRadixNumber(number value, number baseAsDecimal)	Converts a value to a specified number system.

4 Connectors

- [4.1 Triggers](#)
- [4.2 HTTP](#)
- [4.3 Databases](#)
- [4.4 Huawei Cloud Services](#)
- [4.5 SaaS Applications](#)
- [4.6 Emails](#)
- [4.7 Others](#)
- [4.8 Message Systems](#)
- [4.9 Custom Connectors](#)

4.1 Triggers

4.1.1 OpenAPI

The OpenAPI connector is used to generate and release an API. The API is implemented by the subsequent components of the OpenAPI component. Therefore, the OpenAPI component can be used only as the trigger in a composite application and must be followed by other nodes.

After a flow is started, you can choose **API Connect** > **APIs** to view and edit the released API.

Action

Receiving API calls

Configuration Parameters

Parameter	Description
Name	Enter an API name. The name will be displayed on the APIs page of the corresponding ROMA Connect instance.
Security Authentication	<p>Select a security authentication mode. Options:</p> <ul style="list-style-type: none"> • App: ROMA Connect authenticates API requests. When calling an API, a user gets authenticated using the key and secret of an authorized integration application. APIs using this mode can be called by all users. • IAM: IAM authenticates API requests. When calling an API, a user gets authenticated using a token or an AK/SK pair. APIs using this mode can be called only by users on the same cloud service platform. • None: Authentication is not required for API requests. APIs using this mode are less secure and can be called by all users. Be careful when using this mode.
Request Method	Select a request method of the API.
Request Path	Enter the API request path.
Matching	<p>Select a matching mode of the API request path.</p> <ul style="list-style-type: none"> • Exact match: The path in an API request must be the same as the value of Request Path. • Prefix match: The path in an API request must be prefixed with the value of Request Path. For example, if Request Path is set to /test/AA and Matching is set to Prefix match, the API can be accessed using /test/AA/BB or /test/AA/CC but cannot be accessed using /test/AACC.
Request Parameter	<p>Configure the request parameters of the API. Click Add to define API request parameters as required.</p> <ul style="list-style-type: none"> • Name: name the request parameter • Type: Select the type of the request parameter. The options are Header and Query. If the request path contains the Path parameter, the Path parameter is required. • Mandatory: Whether the request parameter is mandatory in an API request. • Default Value: Available when Mandatory is set to No. If no value is assigned to this parameter when the API is called, the default value is used. <p>NOTE Request parameters can be referenced as custom variables in subsequent nodes. For details, see Custom Variables.</p>

Parameter	Description
Example Body	Available when Request Method is set to POST or PUT . Example request body of an API NOTE The request body can be referenced as node variables in subsequent nodes. For details, see Node Variables .
Response Mode	Select a response mode of the API. <ul style="list-style-type: none"> • Synchronous: After receiving a request, the API processes the request immediately and returns a result. • Asynchronous: After receiving a request, the API returns a response body, queues the request, and returns a result after the request is successfully processed.
Response Body	Available when Response Mode is set to Asynchronous . Response body returned after the API receives a request.

4.1.2 FTP

The FTP connector is a channel for file transfer based on the FTP/SFTP protocol. Through the FTP connector, files can be exchanged with an external FTP server.

- The File Transfer Protocol (FTP) is in the TCP/IP protocol group. It consists of two parts: FTP server and FTP client. The FTP server is used to store files. Users can use the FTP client to access resources on the FTP server.
- The SSH File Transfer Protocol (SFTP) is a secure version of FTP and a part of the SSH protocol. SFTP enables easy data transmission and data access through the data flow shell. SFTP, also known as the SSH file transfer protocol, provides a secure connection to transfer files and traverses file systems on local and remote systems.

By default, FTP uses TCP ports 20 and 21. Port 20 is used to transmit data, and port 21 is used to transmit control information (commands). Whether to use port 20 as the port for data transmission depends on the FTP transmission mode. In the active mode, the data transmission port number is 20. In the passive mode, the port is determined after the negotiation between the server and the client.

Creating an FTP Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the FTP connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.

Parameter	Description
Protocol	Protocol used by the FTP connector. Options are FTP and SFTP .
Host IP	IP address of the FTP server.
Port	By default, FTP uses TCP ports 20 and 21. Port 20 is used to transmit data, and port 21 is used to transmit control information (commands).
Username	FTP username.
Password	FTP user password.
Description	Enter the description of the connector to identify it.

Action

Listening to a directory

Configuration Parameters

Parameter	Description
Directory Path	Enter the relative path of the file in the FTP server.

4.1.3 ActiveMQ

ActiveMQ is an open-source message middleware based on the Java Message Service (JMS). ActiveMQ provides a reliable, efficient, and scalable message transfer mechanism and supports multiple message and transmission protocols, such as TCP, UDP, SSL, and NIO. ActiveMQ supports multiple queue and topic modes and can be used in scenarios such as asynchronous message transfer, publishing/subscription, and message routing. It provides a visualized management interface, facilitating configuration and monitoring with high reliability and flexibility. ActiveMQ can be used to build distributed systems. It supports multiple programming languages and platforms and provides various APIs.

Creating an ActiveMQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the ActiveMQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Brokers	List brokers of the ActiveMQ.
Username	Username for connecting to ActiveMQ.
Password	Password for connecting to ActiveMQ.
SSL Authentication	Whether SSL authentication is used for connecting ActiveMQ.
Description	Enter the description of the connector to identify it.

Action

Listening to a queue

Configuration Parameters

Parameter	Description
Name	Name of the message queue to listen.

4.1.4 Jira Software

Jira is a popular and powerful project and issue tracking tool widely used in teams and organizations, including IT, operations, sales, marketing, and human resources. The main functions of Jira are as follows: project management, issue tracking, workflow management, reporting and analysis, and integration and expansion. Jira facilitates team management and collaboration and improves work efficiency and quality.

Creating a Jira Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Jira Software connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.

Parameter	Description
Type	<p>Select the type of the Jira connector.</p> <ul style="list-style-type: none"> • Cloud: The Jira instance is deployed on the cloud. Users can use a browser to access it from anywhere. • On-Premise: The Jira instance is installed locally. Jira is installed and configured on your own server or local PC.
Connection Address	<p>Enter the connection address of Jira.</p> <ul style="list-style-type: none"> • When Type is set to Cloud, the format of the address is <code>https://your-domain.atlassian.net</code>. <i>your-domain</i> indicates the domain name of an organization or team. A valid Jira Cloud subscription is required to access the address. If you are not sure about your Jira Cloud address, obtain it from your Atlassian account or administrator. • When Type is set to On-Premise, enter the IP address or host name of the server where Jira is located. Example: <code>http://192.168.0.1:8080</code> or <code>http://Jira.example.com:8080</code>
Authentication	<p>Select the Jira authentication type. The options are Basic and API Token.</p>
Username	<p>Required only when Authentication is set to Basic Username for connecting to Jira</p>
Password	<p>Required only when Authentication is set to Basic Password for connecting to Jira</p>
Email	<p>Required only when Authentication is set to API Token Jira email address</p>
API Token	<p>Required only when Authentication is set to API Token Generate an API token in the personal settings in Jira. The API token is used for identity authentication and allows Jira users to access their accounts and related data via APIs.</p>
Description	<p>Enter the description of the connector to identify it.</p>

Action

Listening to custom events

Configuration Parameters

Parameter	Description
Listen Events	Select the events to listen in on. The listener in Jira can perform a series of operations when a specified event occurs to automatically extend Jira functions.

4.1.5 Timer

A timer is a trigger that can trigger a specific operation at a preset interval. It can trigger an operation periodically within a specified period or at a specified time point. A timer is used to implement scheduled tasks and scheduling.

Action

- Every Minute
- Hourly
- Weekly
- Monthly
- Fixed Interval

Configuration Parameters

Table 4-1 Every Minute

Parameter	Description
Interval	Every hour, the task is executed at a fixed interval from the start of the hour. Range: 1 to 60
Rule	<ul style="list-style-type: none"> • Daily • Weekdays • Weekends • Custom
Time Zone	<ul style="list-style-type: none"> • Shanghai/Asia • GMT
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

Table 4-2 Hourly

Parameter	Description
Start	Time from 00:00 (the task is also executed within the hour). Range: 0 to 23
End	Time from 00:00 (the task is also executed within the hour). Range: 0 to 23. The value must be greater than the start time.
Trigger Minute	The minute the task is triggered. Range: 0 to 59
Rule	<ul style="list-style-type: none"> • Daily • Weekdays • Weekends • Custom
Time Zone	<ul style="list-style-type: none"> • Shanghai/Asia • GMT
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

Table 4-3 Daily

Parameter	Description
Rule	<ul style="list-style-type: none"> • Daily • Weekdays • Weekends • Custom
Time	Example: 00:59
Time Zone	<ul style="list-style-type: none"> • Shanghai/Asia • GMT
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

Table 4-4 Weekly

Parameter	Description
Weekly Trigger Day	Monday to Sunday
Time	Example: 00:59
Time Zone	<ul style="list-style-type: none"> • Shanghai/Asia • GMT
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

Table 4-5 Monthly

Parameter	Description
Weekly Trigger Day	Monday to Sunday
Time	Example: 00:59
Time Zone	<ul style="list-style-type: none"> • Shanghai/Asia • GMT
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

Table 4-6 Fixed Interval

Parameter	Description
Interval	Value of the interval

Parameter	Description
Execution Interval Unit	<ul style="list-style-type: none"> • Second • Minute • Hour • Day • Week
Task Mode	<ul style="list-style-type: none"> • Parallel: The task is executed using the preset period. • Serial: The next task is triggered only after the current task is complete. Actual execution period = Max (task duration, period)
Execute Once at Once	Whether to execute the task immediately after startup.

4.1.6 ArtemisMQ

ArtemisMQ is an open-source high-performance message queue system built on Apache ActiveMQ and Apache Artemis. ArtemisMQ provides a reliable asynchronous message transfer mechanism that allows different applications to communicate with each other through messages. It uses the message-based middleware mode, allowing producers (senders) to send messages to queues or topics, while consumers (receivers) can receive and process these messages from queues or topics.

Creating an ArtemisMQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the ArtemisMQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
IP Address	Enter the IP address of ArtemisMQ.
Port	Port number for connecting to ArtemisMQ
Username	Username for connecting to ArtemisMQ
Password	Password for connecting to ArtemisMQ
SSL Authentication	Whether SSL authentication is used for connecting to ArtemisMQ.
Description	Enter the description of the connector to identify it.

Action

- Listening to a queue
- Listening to a topic

Configuration Parameters

Table 4-7 Listening to a queue

Parameter	Description
Name	Name of the message queue to listen

Table 4-8 Listening to a queue

Parameter	Description
Name	Name of the message topic to listen

4.1.7 RabbitMQ

RabbitMQ is an open-source message queue system with an efficient message transfer mechanism and can asynchronously transfer and process messages in a distributed system. RabbitMQ is based on Advanced Message Queuing Protocol (AMQP), and supports multiple languages, including Java, Python, Ruby, and .NET.

Creating a RabbitMQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the RabbitMQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter a name for the connector.
IP Address	Enter the IP address for connecting to RabbitMQ
Port	Port number for connecting to RabbitMQ
Username	Username for connecting to RabbitMQ
Password	Password for connecting to RabbitMQ
Virtual Host	Virtual host for connecting to RabbitMQ
SSL Authentication	Whether SSL authentication is used for connecting to RabbitMQ.

Parameter	Description
Description	Enter the description of the connector to identify it.

Action

Listening to an exchange

Configuration Parameters

Table 4-9 Listening to an exchange

Parameter	Description
Name	Name of the exchange to listen

4.1.8 EventGrid

EventGrid (EG) is a serverless event bus service for standard and centralized access of Huawei Cloud services and custom or SaaS applications. You can build a loosely coupled, distributed event-driven architecture to flexibly route events via CloudEvents.

EG triggers are used to quickly create event subscriptions that use Huawei Cloud services or custom applications as event sources and composite application flow tasks as event targets. In this way, you can configure and process subscription events detected by EG in workflows.

Action

- Huawei Cloud event source
- Custom event source

Configuring Parameters

Table 4-10 Huawei Cloud event source

Parameter	Description
Event Subscription Name	Name of the event subscription created in EG
Region	Region of EG
Project	Project to which EG belongs
Channel	Event channel: Store received messages and route them to targets as specified by relevant rules.

Parameter	Description
Event Source	Huawei Cloud service event source
Event Type	Event type of a Huawei Cloud service, which is used to generate filter rules.
Filter Rule	Event filtering rule. For details, see Event Rules .
Connections	Target connection of EG
Event Transformation Rule	Pass-through: Route the complete structure of native events directly to the target. Variable: Route only parameters extracted from events with JSONPath to the target. Constant: Route only constants in events to the target as a trigger.

Table 4-11 Custom event source

Parameter	Description
Event Subscription Name	Name of the event subscription created in EG
Region	Region of EG
Project	Project to which EG belongs
Channel	Event channel: Store received messages and route them to targets as specified by relevant rules.
Event Source	Custom event source
Filter Rule	Event filtering rule. For details, see Event Rules .
Connections	Target connection of EG
Event Transformation Rule	Pass-through: Route the complete structure of native events directly to the target. Variable: Route only parameters extracted from events with JSONPath to the target. Constant: Route only constants in events to the target as a trigger.

4.1.9 QQ Mail

The QQ Mail trigger is used to receive new emails as triggers.

Creating a QQ Mail Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the QQ Mail connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Email Address	QQ Mail address of the sender.
Authorization Code	Authorization code of the QQ Mail sender, which is obtained from the settings in the QQ Mail personal center. For details, see Configuring a QQ Mail Authorization Code .
Description	Enter the description of the connector to identify it.

Action

Receiving new mails

Configuration Parameters

Table 4-12 Receiving new mails

Parameter	Description
Delete Upon Receiving	Whether to delete an email after the email is received.
Unread Only	Whether to receive only unread mails.

4.1.10 IBM MQ

IBM MQ is a message queue software for asynchronous communication in a distributed system to secure reliable data transmission between applications, platforms, and systems. It serves as a middleware and prevents message loss, duplication, or disorder during transmission, and converts and formats data between different systems.

Creating an IBM MQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.

3. Select the IBM MQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
IP Address	Enter the IP address of the IBM MQ host.
Port	Enter the port number of the IBM MQ host.
Queue Manager	A system management program that provides queue services, APIs, and queue management services.
CCSID	Character set used to encode character strings in messages.
Channel Name	Channel for transferring messages between queue managers in the MQ system.
Username	User name for connecting to the IBM MQ.
Password	User password for connecting to the IBM MQ.
SSL Enable	Whether to use SSL authentication when connecting to the IBM MQ. <ul style="list-style-type: none"> • True • False
Cipher Suite	Mandatory when SSL Enable is set to True . Enter the cipher suite.
Trust Store	Mandatory when SSL Enable is set to True . Enter the Base64-encoded trust store file.
Trust Store Password	Mandatory when SSL Enable is set to True . Enter the password of the trust store.
Description	Enter the description of the connector to identify it.

Action

- Listening to a topic
- Listening to a queue

Configuration Parameters

Table 4-13 Listening to a topic

Parameter	Description
Name	Name of the message queue to listen.

Table 4-14 Listening to a queue

Parameter	Description
Name	Name of the message queue to listen.

4.1.11 Kafka

Kafka is a distributed streaming platform developed by LinkedIn. It features high throughput and low latency for massive amounts of real-time data streaming. Kafka consists of producers, consumers, and brokers. Producers send data to the Kafka cluster; consumers subscribe to the cluster and process the data; brokers are the core component of a Kafka cluster as they are responsible for storing and forwarding messages. Featuring scalability, high throughput, low latency, reliability, and durability, Kafka is widely used for big data processing, real-time data streaming, and log collection.

Creating a Kafka Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Kafka connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Broker Address	Enter the Kafka broker address.
Authentication Mode	Select the authentication mode of Kafka. <ul style="list-style-type: none"> • SSL • None
SASL	Mandatory when Authentication Mode is set to SSL . Select the SASL mechanism. <ul style="list-style-type: none"> • PLAIN • SCRAM-SHA-512
SSL Username	Mandatory when Authentication Mode is set to SSL . Enter the SSL username or application key.
SSL Password	Mandatory when Authentication Mode is set to SSL . Enter the SSL password.

Parameter	Description
SSL Certificate Format	Mandatory when Authentication Mode is set to SSL . Select the SSL certificate format. <ul style="list-style-type: none"> • PEM • JKS
SSL Certificate	Mandatory when Authentication Mode is set to SSL . Enter the Base64-encoded certificate content.
SSL Certificate Password	Mandatory when Authentication Mode is set to SSL . Enter the SSL certificate password.
Description	Enter the description of the connector to identify it.

Action

Consuming messages

Configuration Parameters

Parameter	Description
Topic	Topic to listen in on.
GroupId	Unique string to identify the group of consumer processes. Processes with the same group ID belong to the same consumer group.
KeyDeserializer	Deserialization mode of a key. Default: org.apache.kafka.common.serialization.StringDeserializer
ValueDeserializer	Deserialization mode of a value. Default: org.apache.kafka.common.serialization.StringDeserializer
AutoOffsetReset	Where to start consumption when there is no initial offset or the offset is out of range. <ul style="list-style-type: none"> • latest: Start consumption from the latest offset. • earliest: Start consumption from the earliest offset. • none: Report an error.

4.1.12 MySQL

The MySQL connector allows you to connect applications with the MySQL database, and manage data transmission.

- MySQL is a popular, open-source relational database widely used as the backend of web applications. It supports multiple operating systems, such as Windows, Linux, Unix, and macOS X. MySQL features high performance, reliability, and flexibility, and is popular among enterprises and developers.

- The default MySQL port number is 3306. The administrator can change the port number as required. MySQL supports multiple storage engines, including InnoDB, MyISAM, and MEMORY. Each storage engine has its own pros and cons and can be configured flexibly.

Creating a MySQL Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the MySQL connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported.
IP Address	Enter the IP address of the database.
Port	Enter the port number to which the database is connected.
Database Name	Enter the name of the database to be connected.
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

Getting database changes

NOTE

To obtain database changes in real time, enable **binlog** in the database.

Configuring Parameters

Table 4-15 Getting database changes

Parameter	Description
Table Name	Enter or select the name of the data table to get data changes.
Database Operations	Select the change operation on the database.

4.1.13 Oracle

The Oracle connector allows you to connect applications with the Oracle database, and manage data transmission.

- Oracle is a relational database featuring high performance.
- The default Oracle port number is 1521. The administrator can change the port number as required.

Creating an Oracle Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Oracle connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.

Parameter	Description
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Encoding Format	Required when Connection and Security is set to Default . Enter the encoding format of the database.
Timeout	Required when Connection and Security is set to Default . Enter the timeout interval for connecting to the database.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the Oracle database, for example, <code>jdbc:oracle:thin:@{hostname}:{port}:{dbname}</code> .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

Getting database changes

NOTE

To obtain database changes in real time, enable **LogMiner** in the database.

Configuring Parameters

Table 4-16 Getting database changes

Parameter	Description
Table Name	Enter or select the name of the data table to get data changes.
Database Operations	Select the change operation on the database.

4.1.14 PostgreSQL

The PostgreSQL connector allows you to connect applications with the PostgreSQL database, and manage data transmission.

- PostgreSQL is a full-featured, open-source object-relational database.
- The default PostgreSQL port number is 5432. The administrator can change the port number as required.

Creating a PostgreSQL Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the PostgreSQL connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the PostgreSQL database, for example, <code>jdbc:postgresql://{hostname}:{port}/{dbname}</code> .

Parameter	Description
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

Getting database changes

NOTE

To obtain database changes in real time, enable **CDC** in the database.

Configuring Parameters

Table 4-17 Getting database changes

Parameter	Description
Table Name	Enter or select the name of the data table to get data changes.
Database Operations	Select the change operation on the database.

4.1.15 SQL Server

The SQL Server connector allows you to connect applications with the SQL Server database, and manage data transmission.

- SQL Server is an easy-to-use relational database featuring high scalability.
- The default SQL Server port number is 1433. The administrator can change the port number as required.

Creating an SQL Server Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the SQL Server connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.

Parameter	Description
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the SQL Server database, for example, <code>jdbc:sqlserver://{hostname}:{port};DatabaseName={dbname}</code> .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

Getting database changes

NOTE

To obtain database changes in real time, enable **CDC** in the database.

Configuring Parameters

Table 4-18 Getting database changes

Parameter	Description
Table Name	Enter or select the name of the data table to get data changes.
Database Operations	Select the change operation on the database.

4.2 HTTP

4.2.1 HTTP Request

The HTTP request connector calls other OpenAPIs through HTTP/HTTPS to obtain data required for orchestration.

Creating an HTTP Request Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the HTTP request connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Base Address	API request address.
Security Authentication	Authentication mode used to invoke an API. <ul style="list-style-type: none"> • App: Users are authenticated using the key and secret of an authorized integration application. • None: Authentication is not required for API requests.
App Key	Required when Security Authentication is set to App . Key of the integration application for authentication.
APP Secret	Required when Security Authentication is set to App . Secret of the integration application for authentication.
Description	Enter the description of the connector to identify it.

Action

- GET
- POST
- PUT
- DELETE
- HEAD

Configuration Parameters

Parameter	Description
Request URL	API request URL.
Stop on Failure	Whether to stop when an API fails to be called.
Parameter	Configure the request parameters of the API. Click Add to add request parameters based on the API definition.
Request Body Type	Required when Action is set to POST or PUT . Select the request body type of the API.
Body	Required when Action is set to POST or PUT . Enter the API body.

4.2.2 HTTP Response

The HTTP response connector is used to customize the information returned by an API. This way, an OpenAPI will return the specified response code and response body after it is called.

Action

Custom HTTP response

Configuration Parameters

Parameter	Description
Header	Configure the response header parameter of the API response. Use constants, or data from previous nodes by referencing variables .
Body	Configure the API response body. Use constants, or data from previous nodes by referencing variables .
Response Code	Configure the response code of the API response. Select an existing response code or customize one.
Custom Status Code	Required only when Response Code is set to Custom . Enter a custom status code.

4.3 Databases

4.3.1 MySQL

The MySQL connector allows you to connect applications with the MySQL database, and manage data transmission.

- MySQL is a popular, open-source relational database widely used as the backend of web applications. It supports multiple operating systems, such as Windows, Linux, Unix, and macOS X. MySQL features high performance, reliability, and flexibility, and is popular among enterprises and developers.
- The default MySQL port number is 3306. The administrator can change the port number as required. MySQL supports multiple storage engines, including InnoDB, MyISAM, and MEMORY. Each storage engine has its own pros and cons and can be configured flexibly.

Creating a MySQL Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the MySQL connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported.
IP Address	Enter the IP address of the database.
Port	Enter the port number to which the database is connected.
Database Name	Enter the name of the database to be connected.
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Obtaining incremental records
- Adding records
- Updating records
- Replicating records
- Deleting records
- Custom SQL

Configuring Parameters

Table 4-19 Obtaining records

Parameter	Description
Table Name	Enter the name of the data table to be queried.
Field Name	Enter the fields to be queried. Separate multiple fields with commas (,).
WHERE Condition	Enter the conditions for the queried data.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-20 Obtaining incremental records

Parameter	Description
Table Name	Enter the name of the data table to be queried.
Field Name	Enter the fields to be queried. Separate multiple fields with commas (,).
WHERE Condition	Enter the conditions for the queried data.
Order By Field	Enter the sorting field of the returned result.
Start Timezone for incremental database selection	Start time zone for obtaining incremental data.

Parameter	Description
Incremental Selection Timestamp Field Name	Timestamp field for obtaining incremental data.
Incremental Selection Timestamp Initial Value	Initial timestamp for obtaining incremental data.
Reset Initial Incremental Selection Time	Whether to enable the reset of the initial time for obtaining incremental data.
Incremental Selection Compensation Period (ms)	Time compensation for obtaining incremental data.

Table 4-21 Adding records

Parameter	Description
Table Name	Enter the name of the table where data will be inserted.
Data to Insert	Enter the value of the target field in the insert operation.

Table 4-22 Updating records

Parameter	Description
Table Name	Enter the name of the table where data will be updated.
Data to Update	Enter the value of the target field in the update operation.
WHERE Condition	Enter the conditions for the data to be updated.

Table 4-23 Replicating records

Parameter	Description
Table Name	Enter the name of the data table to be replicated.
Clear Table	Whether to clear the table each time data is replicated.

Table 4-24 Deleting records

Parameter	Description
Table Name	Enter the name of the table where data will be deleted.
WHERE Condition	Specifies the conditions for the data to be deleted.

Table 4-25 Custom SQL

Parameter	Description
Statement	<p>Enter the native SQL statement for operating the database.</p> <ul style="list-style-type: none"> • Statements display all your inputs as plain text. Do not include sensitive information. • Be careful when using non-query statements such as drop and truncate.

4.3.2 Redis

The Redis connector is used to connect to a Redis database, helping applications quickly and efficiently interact with the Redis database.

- Redis is a high-performance key-value storage system. It is often used in scenarios such as caching, session management, real-time message transfer, and rankings. Redis supports multiple data structures, including strings, hash tables, lists, sets, and sorted sets. It also provides advanced functions such as transactions, publish/subscription, Lua script execution, persistence, and clustering.
- The default Redis port number is 6379. The administrator can change the port number as required.

Creating a Redis Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Redis connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Host IP Address	Enter the IP address of the database.

Parameter	Description
Port	Enter the port number to which the database is connected.
Database	Enter the name of the database to be connected.
Password	Password for connecting to the database.
Description	Enter the description of the connector to identify it.

Action

- Basic operations on sorted sets
- Basic operations on sets
- Basic operations on hashes
- Basic operations on lists
- Basic operations on strings
- Basic operations on keys
- Basic operations on strings

Configuration Parameters

The following configuration is equivalent to the **ZADD key score member** command to add an element to a sorted set. **key** indicates the name of the sorted set, **score** indicates the score of the element, and **member** indicates the member value of the element. This configuration means to add the element **apple** to the sorted set **myset** and the score of this element is 3.

Parameter	Description
Sorted set <ul style="list-style-type: none"> • zadd • zrem • zcard • zrangebyscore/ zrevrangebyscore • zrange/zrevrange 	<ul style="list-style-type: none"> • Key: name of the sorted set. • Value: member value of an element. • Score: score of an element. • Start Index: lowest score. • End Index: highest score.
Set <ul style="list-style-type: none"> • srem • sismember • srandmember • spop • sadd 	<ul style="list-style-type: none"> • Key: name of the set. • Value: member value of an element.

Parameter	Description
Hash <ul style="list-style-type: none"> • hdel • hlen • hset/hsetnx • hincrby • hget • hgetall 	<ul style="list-style-type: none"> • Key: name of the hash. • Field Name: value of the key in the hash. • Value: value corresponding to the key in the hash. • No operation is performed when the key exists: No: zset; Yes: zsetnx • Increment: adds a specified value to a numeric field in the hash.
List <ul style="list-style-type: none"> • lrange • lpop/rpop • lpush/rpush • lrem • llen • lindex • lset 	<ul style="list-style-type: none"> • Table Name: name of a data table in the database. • Data to Insert: value of each field in the target record of the insert operation.
String <ul style="list-style-type: none"> • set/setnx • append • setbit • get • incrby • decrby • getrange • strlen 	<ul style="list-style-type: none"> • Key: name of the character string. • Value: value of the character string. • Timeout (s): expiration time of the character string. • No operation is performed when the key exists: No: set; Yes: setnx
Key <ul style="list-style-type: none"> • del • lrem • expire • expireat • ttl • persist 	Key: name of the key.

4.3.3 Db2

The Db2 connector is used to connect to a Db2 database by establishing and maintaining the connection between applications and the database, and managing data transmission.

- Db2 is a relational database management system.
- The default Db2 port number is 5000. The administrator can change the port number as required.

Creating a Db2 Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Db2 connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the Db2 database, for example, jdbc:db2://hostname:port/dbname .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Adding records
- Updating records
- Deleting records

Configuration Parameters

Table 4-26 Obtaining records

Parameter	Description
Table Name	Name of the table where data will be queried.
Column Name	Column name of the data table.
WHERE Condition	Enter the condition value (enclose it in single quotation marks) of the data to be queried.
Order By Field	Enter the sorting fields of the returned result.
Limits	Enter the number of records on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-27 Adding records

Parameter	Description
Table Name	Name of the table where data will be inserted.
Data to Insert	Enter the value (enclose it in single quotation marks) to be inserted.

Table 4-28 Updating records

Parameter	Description
Table Name	Name of the table where data will be updated.
Data to Update	Enter the updated value (enclose it in single quotation marks).
WHERE Condition	Enter the condition value (enclose it in single quotation marks) of the data to be updated.

Table 4-29 Deleting records

Parameter	Description
Table Name	Name of the table where data will be deleted.
WHERE Condition	Enter the condition value (enclose it in single quotation marks) of the data to be deleted.

4.3.4 GaussDB

The GaussDB connector allows you to connect applications with the GaussDB database, and manage data transmission.

GaussDB is a distributed relational database developed by Huawei. It uses a distributed architecture and features hybrid transactional/analytical processing (HTAP) capabilities. GaussDB supports intra-city deployment across AZs for zero data loss, petabytes of storage, and more than 1,000 nodes. It is highly available, reliable, secure, and scalable, and provides key capabilities including quick deployment, backup, restoration, monitoring, and alarm reporting for enterprises.

Creating a GaussDB Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the GaussDB connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported. The system automatically concatenates data source connection character strings based on configured data.
Host IP Address	Enter the IP address of the database.
Port	Enter the port number for connecting to the database.
Database Name	Enter the name of the database to be connected.
Username	Enter the username used to connect to the database.

Parameter	Description
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Adding a record
- Updating records
- Deleting records

Configuring Parameters

Table 4-30 Obtaining records

Parameter	Description
Table Name	Enter the name of the data table to be queried.
Field Name	Enter the fields to be queried. Separate multiple field names with commas (,).
WHERE Condition	Enter the conditions for querying data.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records displayed on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-31 Adding a record

Parameter	Description
Table Name	Enter the name of the data table to be inserted.
Data to Insert	Enter the target field value in single quotation marks.

Table 4-32 Updating records

Parameter	Description
Table Name	Enter the name of the data table to be updated.
Update Data	Enter the target field value in single quotation marks.

Parameter	Description
WHERE Condition	Enter the conditions for updating data.

Table 4-33 Deleting records

Parameter	Description
Table Name	Enter the name of the data table to be deleted.
WHERE Condition	Enter the conditions for deleting data.

4.3.5 DWS

The DWS connector allows you to connect applications with the DWS database, and manage data transmission.

GaussDB(DWS) is a cloud-based data warehousing solution that provides scalable, ready-to-use, and fully managed analytical database services. It adheres to ANSI/ISO SQL92, SQL99, and SQL 2003 standards, ensuring compatibility with major database ecosystems such as PostgreSQL, Oracle, Teradata, and MySQL. This makes it a competitive option for petabyte-scale big data analytics across diverse sectors.

Creating a DWS Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the DWS connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported. The system automatically concatenates data source connection character strings based on configured data.

Parameter	Description
Host IP Address	Enter the IP address of the database.
Port	Enter the port number for connecting to the database.
Database Name	Enter the name of the database to be connected.
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Adding a record
- Updating records
- Deleting records

Configuring Parameters

Table 4-34 Obtaining records

Parameter	Description
Table Name	Enter the name of the data table to be queried.
Field Name	Enter the fields to be queried. Separate multiple field names with commas (,).
WHERE Condition	Enter the conditions for querying data.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records displayed on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-35 Adding a record

Parameter	Description
Table Name	Enter the name of the data table to be inserted.
Data to Insert	Enter the target field value in single quotation marks.

Table 4-36 Updating records

Parameter	Description
Table Name	Enter the name of the data table to be updated.
Update Data	Enter the target field value in single quotation marks.
WHERE Condition	Enter the conditions for updating data.

Table 4-37 Deleting records

Parameter	Description
Table Name	Enter the name of the data table to be deleted.
WHERE Condition	Enter the conditions for deleting data.

4.3.6 Milvus

The Milvus connector allows you to connect applications with the Milvus database, and manage data transmission.

Milvus is a high-performance, highly scalable vector database that operates efficiently in diverse environments, from laptops to large distributed systems. It is available as both open-source software and cloud-based services.

Creating a Milvus Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Milvus connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.

Parameter	Description
Connection and Security	Select the connection mode for the database. Currently, the Default mode is supported. The system automatically concatenates data source connection character strings based on configured data.
Host IP Address	Enter the IP address of the database.
Port	Enter the port number for connecting to the database.
Database Name	Enter the name of the database to be connected.
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Adding data
- Querying data
- Deleting data

Configuring Parameters

Table 4-38 Adding data

Parameter	Description
Collection Name	Select or enter the name of the collection to which data is to be added.
Data to Insert	Enter the field name and value of the data to be added.

Table 4-39 Querying data

Parameter	Description
Collection Name	Select or enter the name of the collection whose data is to be queried.
Update Data	Enter the name of the field to be queried.
Search Method	Mode of searching for data. <ul style="list-style-type: none"> • Metadata search: queries non-vector fields. • Vector search: searches for the most matched vector based on vector similarity.

Parameter	Description
Vector Field	This parameter is required only when Search Method is set to Vector Search . Enter the similar vector to be queried. Pay attention to the vector type and dimension.
Top K	This parameter is required only when Search Method is set to Vector Search . Returns the top K results most similar to the query vector, where K is a specified number, such as 5.
Metadata Search	Searches for and filters data based on specified metadata, returning results that meet the criteria. If no value is provided, filtering is not applied.

Table 4-40 Deleting data

Parameter	Description
Collection Name	Select or enter the name of the collection whose data is to be deleted.
Metadata Search	Searches for and filters data based on specified metadata, returning results that meet the criteria.

4.3.7 Oracle

The Oracle connector allows you to connect applications with the Oracle database, and manage data transmission.

- Oracle is a relational database featuring high performance.
- The default Oracle port number is 1521. The administrator can change the port number as required.

Creating an Oracle Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Oracle connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.

Parameter	Description
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	Select the connection mode for the database. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Encoding Format	Required when Connection and Security is set to Default . Enter the encoding format of the database.
Timeout	Required when Connection and Security is set to Default . Enter the timeout interval for connecting to the database.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the Oracle database, for example, <code>jdbc:oracle:thin:@{hostname}:{port}:{dbname}</code> .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Adding records
- Updating records

- Obtaining incremental records
- Deleting records
- Replicating records
- Custom SQL

Configuring Parameters

Table 4-41 Obtaining records

Parameter	Description
Table Name	Select or enter the table name in the database to be queried.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-42 Adding records

Parameter	Description
Table Name	Select or enter the name of the table to which data is to be inserted.
Data to Insert	Enter the target field value in single quotation marks.

Table 4-43 Updating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be updated.
Update Data	Enter the target field value in single quotation marks.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-44 Obtaining incremental records

Parameter	Description
Table Name	Select or enter the name of the table to obtain incremental records.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Start Timezone for incremental database selection	Start time zone for obtaining incremental data.
Incremental Selection Timestamp Field Name	Timestamp field for obtaining incremental data.
Incremental Selection Timestamp Initial Value	Initial timestamp for obtaining incremental data.
Reset Initial Incremental Selection Time	Whether to enable the reset of the initial time for obtaining incremental data.
Incremental Selection Compensation Period (ms)	Time compensation for obtaining incremental data.

Table 4-45 Deleting records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be deleted.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-46 Replicating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be replicated.
Clear Table	Whether to clear the table each time data is replicated.

Table 4-47 Custom SQL

Parameter	Description
Statement	<p>Enter the native SQL statement for operating the database.</p> <ul style="list-style-type: none"> Statements display all your inputs as plain text. Do not include sensitive information. Be careful when using non-query statements such as drop and truncate.

4.3.8 PostgreSQL

The PostgreSQL connector allows you to connect applications with the PostgreSQL database, and manage data transmission.

- PostgreSQL is a full-featured, open-source object-relational database.
- The default PostgreSQL port number is 5432. The administrator can change the port number as required.

Creating a PostgreSQL Connection

- Log in to the new ROMA Connect console.
- In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
- Select the PostgreSQL connector.
- In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.

Parameter	Description
Connection and Security	Select the connection mode for the database. <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	Required when Connection and Security is set to Default . Enter the IP address of the database.
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the PostgreSQL database, for example, <code>jdbc:postgresql://{hostname}:{port}/{dbname}</code> .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Adding records
- Updating records
- Replicating records
- Obtaining incremental records
- Deleting records
- Custom SQL

Configuring Parameters

NOTICE

The PostgreSQL connector supports English table names only.

Table 4-48 Obtaining records

Parameter	Description
Table Name	Select or enter the table name in the database to be queried.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records on each page in the returned result.
Offset	Enter the offset of the pagination query.

Table 4-49 Adding records

Parameter	Description
Table Name	Select or enter the name of the table to which data is to be inserted.
Data to Insert	Enter the target field value in single quotation marks.

Table 4-50 Updating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be updated.
Update Data	Enter the target field value in single quotation marks.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-51 Replicating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be replicated.
Clear Table	Whether to clear the table each time data is replicated.

Table 4-52 Obtaining incremental records

Parameter	Description
Table Name	Select or enter the name of the table to obtain incremental records.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Start Timezone for incremental database selection	Start time zone for obtaining incremental data.
Incremental Selection Timestamp Field Name	Timestamp field for obtaining incremental data.
Incremental Selection Timestamp Initial Value	Initial timestamp for obtaining incremental data.
Reset Initial Incremental Selection Time	Whether to enable the reset of the initial time for obtaining incremental data.
Incremental Selection Compensation Period (ms)	Time compensation for obtaining incremental data.

Table 4-53 Deleting records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be deleted.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-54 Custom SQL

Parameter	Description
Statement	<p>Enter the native SQL statement for operating the database.</p> <ul style="list-style-type: none"> • Statements display all your inputs as plain text. Do not include sensitive information. • Be careful when using non-query statements such as drop and truncate.

4.3.9 SQL Server

The SQL Server connector allows you to connect applications with the SQL Server database, and manage data transmission.

- SQL Server is an easy-to-use relational database featuring high scalability.
- The default SQL Server port number is 1433. The administrator can change the port number as required.

Creating an SQL Server Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the SQL Server connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Region	Select a region.
Project	Select a project.
Instance	Select an instance for subsequent connectivity verification.
Connection and Security	<p>Select the connection mode for the database. Currently, the Default mode is supported.</p> <ul style="list-style-type: none"> • Default: The system automatically concatenates data source connection character strings based on configured data. • Professional: You need to specify the data source connection string manually.
IP Address	<p>Required when Connection and Security is set to Default.</p> <p>Enter the IP address of the database.</p>

Parameter	Description
Port	Required when Connection and Security is set to Default . Enter the port number to which the database is connected.
Database Name	Required when Connection and Security is set to Default . Enter the name of the database to be connected.
Connection String	Required when Connection and Security is set to Professional . Enter the JDBC connection string of the SQL Server database, for example, <code>jdbc:sqlserver://{hostname}:{port};DatabaseName={dbname}</code> .
Username	Enter the username used to connect to the database.
Password	Enter the password used to connect to the database.
Description	Enter the description of the connector to identify it.

Action

- Obtaining records
- Obtaining incremental records
- Adding records
- Updating records
- Replicating records
- Deleting records
- Custom SQL

Configuring Parameters

Table 4-55 Obtaining records

Parameter	Description
Table Name	Select or enter the table name in the database to be queried.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Limits	Enter the number of records on each page in the returned result.

Parameter	Description
Offset	Enter the offset of the pagination query.

Table 4-56 Obtaining incremental records

Parameter	Description
Table Name	Select or enter the name of the table to obtain incremental records.
Field Name	Select the field name of the data table to be queried.
WHERE Condition	Set the condition value in single quotation marks.
Order By Field	Enter the sorting field of the returned result.
Start Timezone for incremental database selection	Start time zone for obtaining incremental data.
Incremental Selection Timestamp Field Name	Timestamp field for obtaining incremental data.
Incremental Selection Timestamp Initial Value	Initial timestamp for obtaining incremental data.
Reset Initial Incremental Selection Time	Whether to enable the reset of the initial time for obtaining incremental data.
Incremental Selection Compensation Period (ms)	Time compensation for obtaining incremental data.

Table 4-57 Adding records

Parameter	Description
Table Name	Select or enter the name of the table to which data is to be inserted.
Data to Insert	Enter the target field value in single quotation marks.

Table 4-58 Updating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be updated.
Data to Update	Enter the target field value in single quotation marks.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-59 Replicating records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be replicated.
Clear Table	Whether to clear the table each time data is replicated.

Table 4-60 Deleting records

Parameter	Description
Table Name	Select or enter the name of the table where data is to be deleted.
WHERE Condition	Set the condition value in single quotation marks.

Table 4-61 Custom SQL

Parameter	Description
Statement	Enter the native SQL statement for operating the database. <ul style="list-style-type: none"> Statements display all your inputs as plain text. Do not include sensitive information. Be careful when using non-query statements such as drop and truncate.

4.4 Huawei Cloud Services

4.4.1 FunctionGraph

The FunctionGraph connector allows you to call functions created in FunctionGraph to process data. The processing result can be referenced in subsequent nodes by [referencing variables](#).

Prerequisites

To use FunctionGraph, you need to have the **FunctionGraph Administrator** permission for project-level services.

Creating a FunctionGraph Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the FunctionGraph connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Authenticated By	Select an authentication mode for the connector. <ul style="list-style-type: none"> • Cloud service agency. For details, see Cloud Service Delegation. • AK/SK
Access Key	Mandatory when Authentication Mode is set to AK/SK . Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Mandatory when Authentication Mode is set to AK/SK . Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

Invoking a function

Configuring Parameters

Parameter	Description
Region	Region where FunctionGraph is located.
Project	Project to which FunctionGraph belongs.
Function	Select the function to be invoked.
Calling Method	Select an invocation mode for the function. Currently, only Synchronous is supported.
Request Data	The request body for executing a function must be in JSON format. You can use data from previous nodes by referencing variables .

NOTE

FunctionGraph provides powerful data processing capabilities in various scenarios. For example, when you use a **Data Source** component to query BLOB data (such as images) from a database (such as MySQL), the query result is in byte arrays. In this case, the queried data can be further processed with FunctionGraph, for example, Base64 data encoding.

4.4.2 SMS

The SMS connector allows you to send SMS messages via Huawei Cloud SMS.

Prerequisites

SMS is enabled.

Creating an SMS Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the SMS connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
App Key	Access key. When you create a Message&SMS application, the platform automatically generates the APP_Key and APP_Secret, which are used for authentication when the SMS APIs are called.

Parameter	Description
App Secret	Access key. When you create a Message&SMS application, the platform automatically generates the APP_Key and APP_Secret, which are used for authentication when the SMS APIs are called.
Description	Enter the description of the connector to identify it.

Action

Sending SMSs

Configuring Parameters

Parameter	Description
Region	Region where the SMS service resides.
Project	Project to which the SMS service belongs.
Channel ID	<p>Enter the channel number for sending SMSs.</p> <ul style="list-style-type: none"> For Chinese mainland SMSs, set this parameter to the channel number allocated by the platform to the SMS signature, which you get when applying for the SMS signature. Example channel number: csms100000001. The signature type corresponding to the channel number must match the template type corresponding to the template ID. For international SMSs, set this parameter to the channel number allocated when the SMS application is created. Example: isms100000001. View the signature channel ID on the Message & SMS console by choosing Chinese Mainland SMS > Signature Management. The signature corresponds to the template. You can view the template's signature at Chinese Mainland SMS > Template Management.
Send With	Select the SMS sending mode. You can select Custom content or Template .
Message Content	<p>Required when Send With is set to Custom content.</p> <p>Enter the SMS content of maximum 2000 bytes. Chinese mainland SMSs must contain an SMS signature. Example: [SMS notification] Welcome to the SMS service.</p>
Template ID	<p>Required when Send With is set to Template.</p> <p>Enter the unique ID obtained when applying for an SMS template. Template ID must be used together with Template Variable Values.</p>

Parameter	Description
Template Variable Values	<p>Required when Send With is set to Template.</p> <p>Enter the variable value in the SMS template. The number and length of variables must be consistent with those defined in the template specified by Template ID.</p> <p>For example, if the template specified by Template ID contains two variables with lengths of 5 and 6, set two variables with lengths not exceeding 5 and 6, respectively. For template content You have \${1} parcel to be claimed at \${2}, this parameter can be set to <code>'["3", "Park Main Entrance"]'</code>.</p>
Recipient Number	<p>Enter the number of the SMS message recipient.</p> <ul style="list-style-type: none"> Chinese mainland SMSs: If the recipient number is a Chinese mainland mobile number, +86 is default but optional. Other numbers must have the standard format of +{country code}{region code}{recipient number}. Examples: +861234***5555, +86123****6666. International SMSs: Recipient numbers must be in the standard format of +{country code}{region code}{recipient number}. Examples: +2412****00 (Gabon), +3366****212 (France). Separate multiple recipient numbers with commas (.). Maximum number length: 21 digits. Maximum number quantity: 500 numbers.
Callback URL	Enter the callback address for receiving the SMS status report, for example, http://xx.com/receiveSMSReport .
Extended Parameter	Enter the extended parameter to be returned in the status report. Braces ({}) are not allowed.
Signature Name	<p>The signature name must be approved and consistent with the template type.</p> <p>This parameter is valid and required when a general template is specified. This is the signature before the SMS content in the general template. This parameter is not required for international SMSs.</p>

4.4.3 SMN

The SMN connector allows you to efficiently send messages to phone numbers, email addresses, and applications via Huawei Cloud Simple Message Notification (SMN).

Creating an SMN Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.

3. Select the SMN connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Authenticated By	Select an authentication mode for the connector. <ul style="list-style-type: none"> • Cloud service agency. For details, see Cloud Service Delegation. • AK/SK
Access Key	Mandatory when Authentication Mode is set to AK/SK . Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Mandatory when Authentication Mode is set to AK/SK . Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- Publishing a message with template
- Publishing a text message

Configuring Parameters

Table 4-62 Publishing a message with template

Parameter	Description
Region	Region where the SMN service resides.
Project	Project to which the SMN service belongs.
Topic	The SMN topic, which stores a specific type of events for message publishing and subscription.
Message Subject	The email subject when emails are sent to subscribers.
Template	The SMN template, which specifies a fixed message format. When publishing a message using a template, set relevant parameters in the template.

Table 4-63 Publishing a text message

Parameter	Description
Region	Region where the SMN service resides.
Project	Project to which the SMN service belongs.
Topic	The SMN topic, which stores a specific type of events for message publishing and subscription.
Message Subject	The email subject when emails are sent to subscribers.
Message Content	Message body sent to the subscriber.

4.4.4 OBS

The OBS connector provides massive, secure, highly reliable, and low-cost object-based cloud storage capabilities via Huawei Cloud Object Storage Service (OBS).

Creating an OBS Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the OBS connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
App Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
App Secret	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- Downloading an object
- Deleting an object
- Listing buckets
- Creating a bucket

- Deleting a bucket
- Obtaining bucket metadata
- Listing objects in a bucket
- Deleting objects in batches from a bucket
- Uploading an object

Configuring Parameters

Table 4-64 Action parameters

Parameter	Description	Actions with the Parameter
Connection Type	Select a connection mode. <ul style="list-style-type: none"> • Region ID • Endpoint 	All
Region ID	Select a region ID.	All
Endpoint	Endpoint	All
Bucket Name	OBS bucket name.	Downloading, deleting, or uploading an object, creating or deleting a bucket, obtaining bucket metadata, and deleting objects in batches from a bucket
Object Name	Enter an object name. Example: 123.png . Only TXT, CSV, PNG, JPG, and JPEG files are supported. Unlike file system, OBS does not have files or folders. The slash (/) in an object name simulates the folder path. Object name (key) of test/123.jpg is test/123.jpg . In this case, set this parameter to test/123.jpg .	Downloading, deleting, or uploading an object
Character Set	UTF-8 or GBK (for downloaded files containing Chinese characters)	Downloading an object
Bucket Type	Specify the bucket type. Default: OBJECT <ul style="list-style-type: none"> • OBJECT • POXIS 	Listing buckets

Parameter	Description	Actions with the Parameter
Bucket Location	Whether to query the bucket location. Yes No (default)	Listing buckets
Region	Where to create the bucket in. Mandatory unless the endpoint is obs.myhuaweicloud.com . Default for endpoint obs.myhuaweicloud.com without specified region: CN North-Beijing1 (cn-north-1).	Creating a bucket
CORS Origin	Origin (usually a domain name) specified by the cross-origin pre-request. Separate origins with line breaks. Each origin can contain one wildcard character (*) at most.	Obtaining bucket metadata
HTTP Request Header	HTTP headers in the cross-domain request. CORS requests must match the allowed headers.	Obtaining bucket metadata
Object Name Prefix	Objects to list must contain this prefix.	Listing objects in a bucket
Start Position	Where the object listing begins, in lexicographical order.	Listing objects in a bucket
Max.	The upper limit for objects in the response. Range: 1–1000. If the input value is greater than 1,000, only 1,000 objects are returned.	Listing objects in a bucket
Delimiter	For grouping object names. If an object name contains the set delimiter, the string from the first character to the first delimiter in the object name (excluding any prefix) is taken as a group under commonPrefix .	Listing objects in a bucket

Parameter	Description	Actions with the Parameter
Coding Type	Encoding method to apply to some elements in the response. For delimiter , marker , prefix , nextMarker , and key containing control characters not supported by the XML 1.0 standard, set encodingType to encode the response delimiter , marker , prefix (including the prefix in commonPrefixes), nextMarker , or key .	Listing objects in a bucket
Response Mode	Response mode for deleting objects in batches. No (default): The response includes the deletion result of each requested object. Yes : The response includes only results of objects failed to be deleted.	Deleting objects in batches from a bucket
Basic Type	Basic type of the object. Text (default): TXT or CSV. Image : PNG, JPG, or JPEG.	Uploading an object
Content	Mandatory for a text object. The system generates an object based on the content and object name.	Uploading an object
Base64 Code	Mandatory for a binary object. The system generates an object based on the Base64 code and object name.	Uploading an object

4.4.5 DSC

The DSC connector interconnects with Huawei Cloud Data Security Center (DSC). DSC is a next-generation cloud data security management platform that protects data assets by data classification, risk identification, data masking, and watermark-based source tracking. Its asset map provides an overview of the security status of each stage across the data security lifecycle.

Creating a DSC Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.

3. Select the DSC connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- Deleting data asset authorization
- Adding asset authorization
- Viewing assets
- Editing data asset name
- Starting/Stopping a data masking task
- Listing data masking tasks
- Extracting the dark image watermark from an image (image address)
- Extracting the dark text watermark from an image (image address)
- Injecting an invisible watermark into an image (image addresses)
- Extracting an invisible watermark from a document (document address)
- Injecting a watermark into a document (document address)
- Masking sensitive data
- Extracting the data watermark
- Injecting a data watermark

Configuring Parameters

Table 4-65 Deleting data asset authorization

Parameter	Description
bucket_id	Bucket ID.
project_id	Project ID.
region_id	Region ID.

Table 4-66 Adding asset authorization

Parameter	Description
project_id	Project ID.
region_id	Region ID.
type	Asset type.
buckets	OBS bucket list.
asset_name	Asset name.
location	Bucket location.
bucket_name	Bucket name.
bucket_policy	Bucket policy.

Table 4-67 Viewing assets

Parameter	Description
project_id	Project ID.
region_id	Region ID.
added	Whether the asset is authorized. <ul style="list-style-type: none"> • true • false
offset	Page number.
limit	Number of records displayed on each page.

Table 4-68 Editing data asset name

Parameter	Description
asset_id	Asset ID.
project_id	Project ID.
region_id	Region ID.
name	Asset name.

Table 4-69 Starting/Stopping a data masking task

Parameter	Description
project_id	Project ID.
template_id	Template ID.
region_id	Region ID.
status	Status of the task.

Table 4-70 Listing data masking tasks

Parameter	Description
project_id	Project ID.
template_id	Template ID.
region_id	Region ID.
workspace_id	Workspace ID.
offset	Page number.
limit	Number of records displayed on each page.

Table 4-71 Extracting the dark image watermark from an image (image address)

Parameter	Description
project_id	Project ID.
region_id	Region ID.
region_id	ID of the region where the project is located. Example: xx-xx-1 .
src_file	Address of the image that the dark image watermark needs to be extracted from, in the obs://bucket/object format. bucket : name of the OBS bucket in the same region as the current project; object : full path name of the object. Huawei Cloud OBS objects only. Example: obs://hwbucket/hwinfo/hw.png , where obs:// is OBS, hwbucket is the bucket name, and hwinfo/hw.png is the full path name of the object.
image_watermark	Storage address of the extracted image watermark in the format of src_file .

Table 4-72 Extracting the dark text watermark from an image (image address)

Parameter	Description
project_id	Project ID.
region_id	Region ID.
region_id	ID of the region where the project is located. Example: xx-xx-1 .
src_file	Address of the image that the dark text watermark needs to be extracted from, in the obs://bucket/object format. bucket : name of the OBS bucket in the same region as the current project; object : full path name of the object. Huawei Cloud OBS objects only. Example: obs://hwbucket/hwinfo/hw.png , where obs:// is OBS, hwbucket is the bucket name, and hwinfo/hw.png is the full path name of the object.
mark_len	Length of the watermark to extract. Range: 0–32. This parameter improves watermark extraction performance.

Table 4-73 Injecting an invisible watermark into an image (image addresses)

Parameter	Description
project_id	Project ID.
region_id	Region ID.
region_id	ID of the region where the project is located. Example: xx-xx-1 .
src_file	Address of the image to add the dark watermark to, in the obs://bucket/object format. bucket : name of the OBS bucket in the same region as the current project; object : full path name of the object. Huawei Cloud OBS files only. Example: obs://hwbucket/hwinfo/hw.png , where obs:// is OBS, hwbucket is the bucket name, and hwinfo/hw.png is the full path name of the object.
blind_watermark	Content of the invisible text watermark. Max.: 32 characters. Digits and letters only. Set either this parameter or image_watermark .
image_watermark	Address of the invisible image watermark in the format of src_file . Set either this parameter or blind_watermark . If both are set, image_watermark takes effect.
dst_file	Storage address of the watermarked image in the format of src_file . If not specified, the value of src_file is used (the original file is overwritten after the watermark is added).

Table 4-74 Extracting an invisible watermark from a document (document address)

Parameter	Description
project_id	Project ID.
region_id	Region ID.
region_id	ID of the region where the project is located. Example: xx-xx-1 .
doc_type	Type of the document a watermark needs to be extracted from. Options: <ul style="list-style-type: none"> • WORD • EXCEL • PDF • PPT
src_file	Address of the document that the dark text watermark needs to be extracted from, in the obs://bucket/object format. bucket : name of the OBS bucket in the same region as the current project; object : full path name of the object. Huawei Cloud OBS objects only. Example: obs://hwbucket/hwinfo/hw.doc , where obs:// is OBS, hwbucket is the bucket name, and hwinfo/hw.doc is the full path name of the object.
file_password	Password for opening a file. Max.: 256 characters. If an Office Word document requires a password for read or domain control, you need to enter the password to open the file.

Table 4-75 Injecting a watermark into a document (document address)

Parameter	Description
project_id	Project ID.
region_id	Region ID.
region_id	ID of the region where the project is located. Example: xx-xx-1 .
src_file	Address of the document to add the watermark to. Format: obs://bucket/object , where bucket is the name of the OBS bucket in the same region as the current project and object is the full path name of the object. Huawei Cloud OBS objects only. Example: obs://hwbucket/hwinfo/hw.png , where obs:// is OBS, hwbucket is the bucket name, and hwinfo/hw.png is the full path name of the object.

Parameter	Description
doc_type	Type of the document to watermark. Options: <ul style="list-style-type: none"> • WORD • EXCEL • PDF • PPT
dst_file	Storage address of the watermarked document in the format of src_file . If not specified, the value of src_file is used (the original file is overwritten after the watermark is added).
blind_watermark	Content of the invisible text watermark. Either this parameter or visible_watermark must be set.
visible_watermark	Content of the visible text watermark. Either this parameter or blind_watermark must be set.
image_mark	Address of document to inject with a visible image watermark. The field is in the format of src_file . The image must be a PNG or JPG file (max.: 1 MB).
visible_type	Whether to inject visible text or image watermarks. TEXT (default): visible_watermark is set to visible text watermark; IMAGE : image_watermark is set to an image watermark address without the visible_watermark , font_size , rotation , and opacity fields. Options: <ul style="list-style-type: none"> • TEXT • IMAGE
file_password	Password for opening a file. Max.: 256 characters. If an Office Word document requires a password for read or domain control, you need to enter the password to open the file.
marked_file_password	Password for a watermarked document. Max.: 256 characters. Default: no password.
readonly_password	Read-only password for a watermarked document. Max.: 256 characters. Default: no read-only password.
front	Font size of a visible watermark. Range: 1–100. Default: 50.
rotation	Font angle of a visible watermark (anticlockwise). Range: 0–90. Default: 45.
opacity	Transparency of a visible watermark. Range: 0–1. Default: 0.3.

Table 4-76 Masking sensitive data

Parameter	Description
project_id	Project ID.
region_id	Region ID.
mask_strategies	List of masking policies. Each policy corresponds to a field. Max.: 100 policies.
name	Name of the field containing sensitive data to mask. Max.: 256 characters.
algorithm	Data masking algorithm name. Options: <ul style="list-style-type: none"> • SHA256 • SHA512 • AES • PRESNM • MASKNM • PRESXY • MASKXY • SYMBOL • KEYWORD • NUMERIC
parameters	Data masking algorithm parameters.
data	Data list.

Table 4-77 Extracting a data watermark

Parameter	Description
project_id	Project ID.
region_id	Region ID.
watermark_key	Watermark key.
columns	List of field types. Max.: 100 types. Must include all fields with primary_key as true and at least one field with primary_key as false to extract watermarks.
name	Field name. Max.: 256 characters.

Parameter	Description
type	Field type. Options: <ul style="list-style-type: none"> • INTEGER • STRING • DOUBLE
primary_key	Whether the field is a primary key. true : primary key, for locating the watermark. false : non-primary key, for embedding or extracting the watermark content in this column. The field type list can contain multiple fields with primary_key as true or false .
data	Watermark data. Max.: 30,000 records.

Table 4-78 Injecting a data watermark

Parameter	Description
project_id	Project ID.
region_id	Region ID.
watermark_content	Watermark content.
watermark_key	Watermark key.
columns	List of field types. Max.: 100 types. Include at least two fields: one primary_key as true (for the primary key) and one primary_key as false (for watermark injection).
name	Field name. Max.: 256 characters.
type	Field type. Options: <ul style="list-style-type: none"> • INTEGER • STRING • DOUBLE
primary_key	Whether the field is a primary key. true : primary key, for locating the watermark. false : non-primary key, for embedding or extracting the watermark content in this column. The field type list can contain multiple fields with primary_key as true or false .
data	Content of a field, which can contain a maximum of 2,000 characters.

4.4.6 Image Recognition

The Image Recognition connector allows accurate image recognition via the Huawei Cloud Image Recognition service .

Prerequisites

Huawei Cloud Image Recognition has been enabled.

Creating an Image Recognition Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Choose **Huawei Cloud Services > Image Recognition**
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Authenticated By	Select an authentication mode for the connector. <ul style="list-style-type: none"> • Cloud service agency. For details, see Cloud Service Delegation. • AK/SK
Access Key	Mandatory when Authentication Mode is set to AK/SK . Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Mandatory when Authentication Mode is set to AK/SK . Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- **Celebrity recognition:** Identifying movie stars and internet celebrities in images.
- **Tag recognition:** Identifying the name, category, and confidence of an object in an image.

Configuring Parameters

Table 4-79 Celebrity recognition

Parameter	Description
Region	Region where the Image Recognition service resides.
Project	Project to which the Image Recognition service belongs.
Obtain Image from	Method to obtain the image. Options are API form parameters, Base64 encoding, and URL path.
API form parameters	If Obtain Image from is set to API form parameters , the trigger of this task flow must be OpenAPI, and images must be uploaded using form parameters of the form-data type. Set this parameter to the name of the form parameters.
Base64 encoding	If Obtain Image from is set to Base64 encoding , set this parameter to the Base64-encoded character string of the image. The size cannot exceed 10 MB.
URL path	If Obtain Image from is set to URL path , set this parameter to the URL of the image. The value can be: <ul style="list-style-type: none"> • HTTP/HTTPS URLs on the public network • Huawei Cloud OBS URLs. To use OBS data, authorization is required, including service authorization, temporary authorization, and anonymous public authorization. For details, see Configuring the Access Permission of OBS.
Confidence	Threshold (0 to 1) of the confidence score. Tags with a confidence score lower than the threshold will not be returned.
Tags	Maximum number of tags that can be returned. The default value is 30 .

Parameter	Description
Tag Language	Language of the returned tags. <ul style="list-style-type: none"> • zh: Chinese • en: English The default value is zh .

Table 4-80 Tag recognition

Parameter	Description
Region	Region where the Image Recognition service resides.
Project	Project to which the Image Recognition service belongs.
Obtain Image from	Method to obtain the image, including API form parameters, Base64 encoding, and URL path.
API form parameters	If Obtain Image from is set to API form parameters , the trigger of this task flow must be Open API, and images must be uploaded using form parameters of the form-data type. Set this parameter to the name of the form parameters.
Base64 encoding	If Obtain Image from is set to Base64 encoding , set this parameter to the Base64-encoded character string of the image. The size cannot exceed 10 MB.
URL path	If Obtain Image from is set to URL path , set this parameter to the URL of the image. The value can be: <ul style="list-style-type: none"> • HTTP/HTTPS URLs on the public network • Huawei Cloud OBS URLs. To use OBS data, authorization is required, including service authorization, temporary authorization, and anonymous public authorization. For details, see Configuring the Access Permission of OBS.

Parameter	Description
Confidence	Threshold (0 to 1) of the confidence score. Tags with a confidence score lower than the threshold will not be returned.

4.4.7 OCR

The OCR connector allows accurate text recognition via Huawei Cloud Optical Character Recognition (OCR).

Prerequisites

Huawei Cloud OCR is enabled.

Creating an OCR Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Choose **Cloud Service > OCR**
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- General table OCR
- General text OCR
- Web image OCR
- Auto classification OCR
- Handwritten text OCR
- ID card OCR

- Vehicle license OCR
- Driving license OCR
- Passport OCR
- Bank card OCR
- Business license OCR
- Transportation license OCR
- License plate OCR
- Business card OCR
- VIN OCR
- VAT Fapiao OCR
- Fapiao verification
- Motor vehicle sales Fapiao OCR
- Taxi Fapiao OCR
- Train ticket OCR
- Fixed-amount Fapiao OCR
- Toll Fapiao OCR
- Flight itinerary OCR
- E-Waybill OCR
- Insurance policy OCR
- Financial statement OCR
- Transportation qualification certificate OCR
- Custom OCR

Configuring Parameters

For details about the parameters, see [API Reference](#) of Optical Character Recognition.

4.4.8 FRS

The FRS connector provides identity verification services via Huawei Cloud Facial Recognition Service (FRS), which uses a computer to process, analyze, and understand face images based on human facial features.

Prerequisites

Huawei Cloud FRS is enabled.

Creating an FRS Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Choose **Cloud Service > FRS**
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Description	Enter the description of the connector to identify it.

Action

- Face Recognition
- Face Matching
- Action LiveDetect
- Silent LiveDetect
- Face Retrieval
- Create a Face Library
- Query All Facial Image Libraries
- Query Face Libraries
- Delete a Face Library
- Add a Face
- Query a Face
- Update a Face
- Delete a Face
- Delete Faces in Batches

Configuring Parameters

For details about the parameters, see [FRS API Reference](#).

4.4.9 CodeArts Artifact

The CodeArts Artifact connector interconnects with Huawei Cloud CodeArts Artifact.

CodeArts Artifact is designed for software release and management. Its secure software repositories allow you to manage software packages and their metadata, download release packages, and release software, achieving continuous delivery.

Prerequisites

You have subscribed to CodeArts Artifact.

Creating a CodeArts Artifact Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Artifact connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

Listing file versions in a project

Configuring Parameters

Table 4-81 File version parameters

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
file_name	File name. Fuzzy search is supported.
limit	Number of items displayed per page.
offset	Offset, which is the position where the query starts. The value must be no less than 0.

4.4.10 CodeArts Build

The CodeArts Build connector interconnects with Huawei Cloud CodeArts Build.

CodeArts Build provides an easy-to-use platform that supports multi-language parallel builds on the cloud. It helps enterprises improve build efficiency while lowering overall costs even at large scale.

Prerequisites

You have subscribed to CodeArts Build.

Creating a CodeArts Build Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Build connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Querying the latest successful build of a specified repository
- Querying details of a build record
- Executing a build task
- Querying success rate of a build task in a specified period
- Checking task running status
- Viewing the build task list in a project
- Downloading Keystore files
- Viewing the build history of a task based on start time and end time
- Viewing the build history of build tasks

Configuring Parameters

Table 4-82 Querying the latest successful build of a specified repository

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.

Parameter	Description
repository_name	Name of the code repository.

Table 4-83 Querying details of a build record

Parameter	Description
job_id	Build task ID, which contains the 32 digits and letters at the end of the browser URL when the build task is edited.
build_number	Build number of the task, which starts from 1 and increases by 1 for each build.
region_id	Region ID.

Table 4-84 Executing a build task

Parameter	Description
region_id	Region ID.
body	Enter the request body.

Table 4-85 Querying success rate of a build task in a specified period

Parameter	Description
job_id	Build task ID, which contains the 32 digits and letters at the end of the browser URL when the build task is edited.
region_id	Region ID.
start_time	Start time, which is in the YYYY-MM-DD format.
end_time	End time, which is in the YYYY-MM-DD format.

Table 4-86 Checking task running status

Parameter	Description
job_id	Build task ID, which contains the 32 digits and letters at the end of the browser URL when the build task is edited.
region_id	Region ID.

Table 4-87 Viewing the build task list in a project

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
page_index	Page number, indicating that the query starts from this page. The value is no less than 0 .
page_size	Number of items displayed on each page. The value is no more than 100 .

Table 4-88 Downloading Keystore files

Parameter	Description
region_id	Region ID.
file_name	Name of the file to download.
domain_id	Tenant ID, which is a string of 32 digits and lowercase letters.

Table 4-89 Viewing the build history of a task based on start time and end time

Parameter	Description
job_id	Build task ID, which contains the 32 digits and letters at the end of the browser URL when the build task is edited.
region_id	Region ID.
offset	Page number, indicating that the query starts from this page. The value is no less than 0 .
limit	Number of items displayed on each page. The value is no more than 100 .
start_time	Start time, which is in the YYYY-MM-DD format. The time range cannot exceed 30 days.
end_time	End time, which is in the YYYY-MM-DD format. The time range cannot exceed 30 days.

Table 4-90 Viewing the build history of build tasks

Parameter	Description
job_id	Build task ID, which contains the 32 digits and letters at the end of the browser URL when the build task is edited.

Parameter	Description
region_id	Region ID.
offset	Page number, indicating that the query starts from this page. The value is no less than 0 .
limit	Number of items displayed on each page. The value is no more than 100 .
interval	The time interval from the current date. The value is no more than 30 .

4.4.11 CodeArts Check

The CodeArts Check connector interconnects with Huawei Cloud CodeArts Check.

CodeArts Check is an in-house code check service. It is distilled from Huawei's years of experience in automatic static check and enterprise-level applications. CodeArts Check provides rich check capabilities for code styles, common quality, and cyber security risks. Comprehensive quality reports and convenient bug handling help enterprises effectively control code quality and achieve success.

Prerequisites

You have subscribed to CodeArts Check.

Creating a CodeArts Check Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Check connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Modifying a task rule set
- Configuring advanced settings for a task
- Querying check failure logs
- Shielding directories of a task.
- Query advanced settings of a task
- Creating a check task
- Querying check parameters of a rule set
- Configuring check parameters of a task
- Setting a default rule set for each language in a project
- Customizing a rule set
- Querying the rule set list
- Viewing the rule list of a rule set
- Querying selected rule sets of a task
- Deleting a custom rule set
- Obtaining rules
- Stopping a check task
- Querying issue details statistics
- Modifying the issue status
- Querying the task status
- Querying historical scan results
- Querying issue summary
- Querying the CMetrics issue summary
- Querying issue details
- Querying the task list
- Executing a check task
- Deleting a check task

Configuring Parameters

For details about the parameters, see the [API Reference](#) of CodeArts Check.

4.4.12 CodeArts Deploy

The CodeArts Deploy connector interconnects with Huawei Cloud CodeArts Deploy.

CodeArts Deploy provides visualized, one-click deployment. It supports deployment on VMs or containers by using Tomcat, Spring Boot, and other templates or by flexibly orchestrating atomic actions. It also supports parallel deployment and seamless integration with CodeArts Pipeline, providing standard deployment environments and automatic deployment.

Prerequisites

You have subscribed to CodeArts Deploy.

Creating a CodeArts Deploy Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Deploy connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Creating a host group
- Querying the deployment history of an application in a project by start time and end time
- Obtaining the success rate of deployed applications in a specified application
- Obtaining the application deployment success rate of a specified project
- Deleting a host group
- Obtaining application details
- Querying host details
- Modifying a host
- Querying a host group
- Deleting a host
- Modifying a host group
- Deleting an application
- Creating a host
- Querying the host group list
- Deploying an application
- Creating an application using a template
- Querying the host list
- Obtaining applications

Configuring Parameters

For details about the parameters, see the [API Reference](#) of CodeArts Deploy.

4.4.13 CodeArts Req

The CodeArts Req connector interconnects with Huawei Cloud CodeArts Req.

CodeArts Req provides highly-efficient requirement management services for software R&D teams to foster collaboration. You can manage requirements via Scrum projects, track bugs, host documents, gain insights from statistics, and manage workloads.

Prerequisites

You have subscribed to CodeArts Req.

Creating a CodeArts Req Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Req connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Querying work item status statistics in a sprint (by handler)
- Querying the project status list
- Updating the domain of a project
- Querying the domain list of a project
- Deleting a project module
- Disassociating a domain from a project
- Querying the module list of a project
- Creating a domain for a project

- Checking sprint history
- Querying associated cases
- Querying code commit or branch creation records associated with the current work item
- Querying wikis associated with the current work item
- Querying work items associated with the current work item
- Querying work item transitions of a scrum project
- Obtaining project statistics overview
- Obtaining requirement statistics
- Querying the history of all work items in a project
- Querying bug statistics
- Updating a work item
- Querying workload by user (multi-project)
- Deleting work items in batches
- Deleting a work item
- Obtaining work item history
- Creating a work item
- Querying project work items
- Finding work items using advanced query
- Querying work item details
- Querying child work items
- Querying workloads by user (single-project)
- Querying the current user information
- Checking sprint details
- Querying sprints of a specified project
- Creating a project
- Updating Scrum project sprints
- Deleting project sprints
- Deleting sprints of projects in batches
- Creating a Scrum project sprint
- Checking whether a project name exists
- Obtaining comments on a specified work item
- Obtaining work item completion rate
- Obtaining the member list of a specified project
- Adding a project member
- Querying the project list
- Querying bug density
- Querying custom fields of a Scrum work item
- Creating work items with fine-grained permissions
- Querying average bugs per user

- Creating a project module
- Updating a project module
- Updating member roles in a project
- Adding project members in batches
- Deleting project members in batches
- Querying on-time requirement completion rate
- Querying project details
- Updating a project
- Deleting a project

Configuring Parameters

For details about the parameters, see the [API Reference](#) of CodeArts Req.

4.4.14 CodeArts Pipeline

The CodeArts Pipeline connector interconnects with Huawei Cloud CodeArts Pipeline.

CodeArts Pipeline allows you to visualize and customize pipelines to shorten the delivery period and improve quality.

Prerequisites

You have subscribed to CodeArts Pipeline.

Creating a CodeArts Pipeline Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Pipeline connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Obtaining the pipeline execution status of a project
- Obtaining the pipeline status
- Obtaining the pipeline list
- Deleting a pipeline
- Obtaining pipeline status in batches
- Stopping a pipeline
- Starting a pipeline
- Checking the pipeline creation status
- Querying template details
- Querying the template list

Configuring Parameters

Table 4-91 Obtaining the pipeline execution status of a project

Parameter	Description
region_id	Region ID.
project_id	CodeArts project ID.
start_date	The start date is included in the query period. The interval between the start date and end date cannot exceed one month.
end_date	The end date is included in the query period. The interval between the start date and end date cannot exceed one month.
offset	Offset, which is the position where the query starts. The value must be no less than 0.
limit	Number of records of each query.

Table 4-92 Obtaining the pipeline status

Parameter	Description
pipeline_id	ID of the pipeline whose status is to be queried.
region_id	Region ID.
build_id	Run ID of the pipeline whose status is to be queried.

Table 4-93 Obtaining the pipeline list

Parameter	Description
region_id	Region ID.

Parameter	Description
pipeline_name	Pipeline name. If the parameter exists, fuzzy match is performed.
project_ids	CodeArts project IDs. If this parameter is specified, the pipeline list of the corresponding project is returned. Separated project IDs with commas (.). The number of IDs range from 0 to 10. If not, the pipeline list of the tenant to which the invoker belongs is returned.
creator_ids	Creator ID. Separated IDs with commas (.). The number of IDs range from 0 to 10.
executor_ids	Executor ID. Separated IDs with commas (.). The number of IDs range from 0 to 10.
status	Pipeline execution status. Value options: waiting, running, verifying, handling, suspending, and completed.
outcome	Pipeline result, which is used to mark the pipeline. Value options: error, success, aborted.
sort_key	Sorting field. Value options: pipeline_name, create_time, start_time.
sort_dir	Sorting type. asc : Ascending; desc : Descending.
git_url	Link format: git@codehub.XXX.git (CodeHub); git@gitee.com.XXX.git (Gitee); git@github.com.XXX.git (GitHub).
offset	Offset, which is the position where the query starts. Min.: 0 (default).
limit	Number of records of each query. Value range: 10 (default) to 50.

Table 4-94 Deleting a pipeline

Parameter	Description
pipeline_id	ID of the pipeline to be deleted.
region_id	Region ID.

Table 4-95 Obtaining pipeline status in batches

Parameter	Description
region_id	Region ID.
pipeline_ids	Pipeline IDs for status inquiry.

Table 4-96 Stopping a pipeline

Parameter	Description
pipeline_id	Pipeline ID.
region_id	Region ID.
build_id	Pipeline execution ID.

Table 4-97 Starting a pipeline

Parameter	Description
pipeline_id	Pipeline ID.
region_id	Region ID.
name	Name of the build parameter.
value	Value of the build parameter. Max. length: 8,192 characters.

Table 4-98 Checking the pipeline creation status

Parameter	Description
task_id	Instance ID.
region_id	Region ID.

Table 4-99 Querying template details

Parameter	Description
template_id	Template ID.
region_id	Region ID.
template_type	Template type.
source	API caller.

Table 4-100 Querying the template list

Parameter	Description
region_id	Region ID.

Parameter	Description
template_type	Template type.
is_build_in	Built-in template or not.
offset	Offset, which is the position where the query starts. The value must be no less than 0.
limit	Number of items displayed per page.
name	Template name, which is fuzzily matched.
sort	Sorting field.
asc	Ascending order or not.

4.4.15 CodeArts Repo

The CodeArts Repo connector interconnects with Huawei Cloud CodeArts Repo.

CodeArts Repo is Huawei's proprietary code hosting service that provides Git-based distributed code management and collaborative development capabilities, including member management, permission control, code hosting, check, review, and tracing, and continuous integration. It helps enterprises of different scales improve R&D quality and efficiency.

Prerequisites

You have subscribed to CodeArts Repo.

Creating a CodeArts Repo Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts Repo connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Parameter	Description
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

Obtaining files in a branch directory; obtaining repository statistics; modifying the status of a repository referenced by a pipeline; checking whether a user has the repository administrator permission; setting a repository to public or private; obtaining the list of repositories that can be set to public under a project; obtaining a public template list; obtaining information about all repositories in a project; verifying a repository name of a specified project; creating a repository; obtaining a repository short ID based on the repository or repository group name to combine the url of the commit details page corresponding to the commit ID; querying a commit of a branch by repository or repository group name; querying a webhook of a specified repository; adding a hook to a specified repository; deleting the hook of a specified repository; querying branches of a specified repository; querying a commit of a repository branch by the repository ID; adding a deploy key; deleting a repository deploy key; obtaining the last commit statistics of a repository; repository statistics; querying a repository under a project; deleting a repository; obtaining all member records in a repository; adding a repository member; setting a role for a member in the repository; deleting a repository member; viewing the repository creation status; verifying HTTPS account and password; obtaining an SSH key list; adding an SSH Key; checking whether the private key has the permission to pull code; deleting a user's public key; query all repositories of a user; creating a project and repository; creating a project and forking a repository; querying all repositories in a project; querying a tag list of a repository; creating a tag; querying details of a repository; querying the commit message of a repository; creating a submit; querying a specific commit message of a repository; querying the commit difference information of a repository; querying a files in a repository; obtaining the number of committed code lines; downloading a repository; creating a protected branch; obtaining the repository branch list; associating branches with work items; obtaining an MR list of a repository; obtaining MR details of a repository

Configuring Parameters

For details about the parameters, see the [API Reference](#) of CodeArts Repo.

4.4.16 CodeArts TestPlan

The CodeArts TestPlan connector interconnects with Huawei Cloud CodeArts TestPlan.

CodeArts TestPlan is a one-stop cloud testing platform provided for software developers. It covers function and API tests and integrates the DevOps agile testing concepts, helping you improve management efficiency and deliver high-quality products.

Prerequisites

You have subscribed to CodeArts TestPlan.

Creating a CodeArts TestPlan Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts TestPlan connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Calculating a custom report in real time
- Querying the requirement tree of a test plan
- Querying the operation history of a test plan
- Obtaining the environment parameter list
- Generating an API test suite by importing files from a repository
- Adding test cases in batches to a plan
- Querying the test plan list in a project (V2)
- Creating a plan in a project
- Querying the test plan list in a project
- Obtaining test case details by case ID
- Updating test case results in batches
- Executing test cases in batches
- Updating a test case in a custom test
- Obtaining test case details
- Deleting test cases in batches in a custom test
- Creating a test case in a custom test
- Deleting a registered service

- Updating a registered service
- Registering a new type of testing service
- Obtaining the information about registered services of the user

Configuring Parameters

Table 4-101 Calculating a custom report in real time

Parameter	Description
project_id	CodeArts project ID.
plan_id	Branch/Plan ID.
region_id	Region ID.
name	Report name.
workpiece_type	Data type (test case: case; test suite: suite).
analysis_dimension_row	Analysis dimension.
compare_dimension_column	Comparison dimension.
creatorIds	Creator.
ownerIds	Owner.
ranks	Level.
releaseIds	Release ID.
status	Status.
moduleIds	Module ID.
results	Results.
labelIds	Tag.
startTime	Start time.
endTime	End time.
isAssociateIssue	Whether the test case is associated with a requirement.

Table 4-102 Querying the requirement tree of a test plan

Parameter	Description
project_id	CodeArts project ID.

Parameter	Description
plan_id	Test plan ID.
region_id	Region ID.
offset	Start offset.
limit	Number of items displayed on each page. A maximum of 200 items are supported.

Table 4-103 Querying the operation history of a test plan

Parameter	Description
project_id	CodeArts project ID.
plan_id	Test plan ID.
region_id	Region ID.
offset	Start offset.
limit	Number of items displayed on each page. A maximum of 200 items are supported.

Table 4-104 Obtaining the environment parameter list

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
offset	Start offset.
limit	Number of items displayed on each page. A maximum of 200 items are supported.

Table 4-105 Generating an API test suite by importing files from a repository

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
testsuite_name	Name of the test suite to be generated.
repository_id	Repository ID.

Parameter	Description
repository_branch	Repository branch.
file_path	Relative path of the YAML or JSON file in the repository. Only YAML and JSON files of Swagger 2.0 are supported.

Table 4-106 Adding test cases in batches to a plan

Parameter	Description
project_id	CodeArts project ID.
plan_id	Plan ID.
region_id	Region ID.
service_id	Service ID returned by the API for registering a test service.
testcase_id_list	Number of test cases contained in the plan. The array length is less than 50. Only one test type can be contained.

Table 4-107 Querying the test plan list in a project (V2)

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
offset	Start offset.
limit	Number of items displayed on each page. A maximum of 200 items are supported.
name	Test plan name for fuzzy search.
current_stage	Test plan stage (create, design, execute, or report).

Table 4-108 Creating a plan in a project

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
name	Plan name.
assigned_id	Processor ID.

Parameter	Description
service_id_list	Test case types contained in the plan. The array length is less than 10.
start_date	Plan start time, in UTC. Example: 2020-03-04
end_date	Plan end time, in UTC. Example: 2020-03-04

Table 4-109 Querying the test plan list in a project

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
name	Test plan name for fuzzy search.
current_stage	Test plan stage (create, design, execute, or report).
offset	Page number. The value ranges from 1 to 20000.
limit	Number of items displayed on each page. The value ranges from 1 to 200.

Table 4-110 Obtaining test case details by case ID

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
testcase_number	Test case ID.

Table 4-111 Updating test case results in batches

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
service_id	Service ID returned by the API for registering a test service.
testcase_id	Test case ID.
execute_id	Execution ID of the registered service.

Parameter	Description
result_id	Test case result. 0 : successful, 1 : failed, 5 : executing, and 6 : stopped.
end_time	Timestamp when case execution ends (mandatory when execution ends).
duration	Case execution duration, in milliseconds (mandatory when the status is updated).
description	Execution result remarks.

Table 4-112 Executing test cases in batches

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
service_id	Service ID returned by the API for registering a test service.
plan_id	Test plan ID.
testcase_id	Test case ID.
execute_id	Execution ID of the registered service.
result_id	Test case result. 0 : successful, 1 : failed, 5 : executing, and 6 : stopped.
start_time	Timestamp when case execution starts (mandatory when case execution starts).

Table 4-113 Updating a test case in a custom test

Parameter	Description
project_id	CodeArts project ID.
testcase_id	Test case ID.
region_id	Region ID.
name	Case name displayed on the page.
service_id	Service ID returned by the API for registering a test service.
rank_id	Test case level. Value options: 0 , 1 , 2 (default), 3 , and 4 .
testcase_number	Test case ID. If this parameter is left blank, the system automatically generates a case ID.

Parameter	Description
description	Test case description.
preparation	Prerequisites that must be met for executing the test case.
expect_result	Expected result of a test case.
test_step	Test step description.
label_list	Tag name list.
module_id	Module ID.
test_version_id	Test version number.
fix_version_id	Sprint number.
assigned_id	Processor ID.
issue_id	ID of the requirement associated with the case.
status_id	Test case status. 0 : new; 5 : designing; 6 : testing; 7 : completed.
defect_id_list	Bug IDs.

Table 4-114 Obtaining test case details

Parameter	Description
project_id	CodeArts project ID.
testcase_id	Test case ID.
region_id	Region ID.

Table 4-115 Deleting test cases in batches in a custom test

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
service_id	Service ID returned by the API for registering a test service.
test_case_id_list	Test case ID.

Table 4-116 Creating a test case in a custom test

Parameter	Description
project_id	CodeArts project ID.
region_id	Region ID.
name	Case name displayed on the page.
service_id	The value is returned by the registration API.
rank_id	Test case level. Value options: 0 , 1 , 2 (default), 3 , and 4 .
testcase_number	Test case ID. If this parameter is left blank, the system automatically generates a case ID.
description	Test case description.
preparation	Prerequisites that must be met for executing the test case.
expect_result	Expected result of a test case.
test_step	Test step description.
label_list	Tag name list.
module_id	Module ID.
test_version_id	Test version number.
fix_version_id	Sprint number.
assigned_id	Processor ID.
issue_id	ID of the requirement associated with the case.

Table 4-117 Deleting a registered service

Parameter	Description
service_id	Unique ID of a registered service. The value is returned by the registration API.
region_id	Region ID.

Table 4-118 Updating a registered service

Parameter	Description
service_id	Unique ID of a registered service. The value is returned by the registration API.
region_id	Region ID.

Parameter	Description
service_name	Test type name displayed on the GUI. The current reserved name cannot be used.
server_host	Domain name, which is used to call APIs. It must start with https or http and contain a maximum of 128 characters. TestHub uses the API under this domain name to keep the consistency between service data and user system data.

Table 4-119 Registering a new type of testing service

Parameter	Description
region_id	Region ID.
service_name	Test type name displayed on the GUI. The current reserved name cannot be used.
server_host	Domain name, which is used to call APIs. It must start with https or http and contain a maximum of 128 characters. TestHub uses the API under this domain name to keep the consistency between service data and user system data.

Table 4-120 Obtaining the information about registered services of the user

Parameter	Description
region_id	Region ID.

4.4.17 CodeArts IDE Online

The CodeArts IDE Online connector interconnects with Huawei Cloud CodeArts IDE Online.

CodeArts IDE Online is a cloud development environment service that enables you to quickly obtain and access the environment using a browser to complete coding, building, debugging, running, code preview, repo access, and command execution, along with a broad range of extensions.

Prerequisites

You have subscribed to CodeArts IDE Online.

Creating a CodeArts IDE Online Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.

3. Select the CodeArts IDE Online connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Listing IDE instances
- Creating an IDE instance
- Listing IDE instances of a tenant
- Stopping an IDE instance
- Starting an IDE instance
- Deleting an IDE instance
- Modifying an IDE instance
- Querying an IDE instance
- Checking whether the IDE instance name is duplicate

Configuring Parameters

Table 4-121 Listing IDE instances

Parameter	Description
region_id	Region ID.
limit	Number of records displayed on each page. Value range: 1-100
offset	Offset where the query starts.
is_temporary	Whether to display on the page (depending on the tag configuration).
label	Tag.
search	Query by keyword (fuzzy query by instance name and description).
sort_dir	Sorting mode. Value options: asc and desc .

Parameter	Description
sort_key	Sorting field. Value options: display_name (instance name), status , pvc_quantity (storage capacity), created_time , and stack_id .

Table 4-122 Creating an IDE instance

Parameter	Description
org_id	Tenant ID (corresponding to domainId of the Huawei Cloud account).
region_id	Region ID.
agent_id	Dealer ID, which is used in the tutorial activity scenario.
arch	CPU architecture. Value options: x86 and Arm .
cpu_memory	CPU specifications (Arm: 4 vCPUs and 8 GB memory; x86: 1 vCPU and 1 GB memory, 2 vCPUs and 4 GB memory, and 2 vCPUs and 8 GB memory), which correspond to the specifications configured for the stack and can be obtained by calling the ListStacks API. If the tag is not empty, refer to the stack specifications configured for the tag. Quantum stack: 2 vCPUs and 8 GB memory (x86). Other stacks: 1 vCPU and 1 GB memory or 2 vCPUs and 4 GB memory (x86).
description	Description.
display_name	Instance name, which can contain digits, letters, underscores (_), periods (.), and hyphens (-). Value length: 3–100 characters
is_temporary	Instance tag. Third-party systems need to set the tag together with CodeArts IDE Online. If this parameter is not transferred, the default value is used.
label_tag	Sorting mode. Value options: asc and desc .
plugin_enable_list	Pre-set plug-ins.
plugin_vars	Pre-set plug-in parameters. Encrypt sensitive information if needed.
port_id	(Optional) Port ID of the cloud server, required when the ECS is accessed via the private network.
private_ip	(Optional) IP address of the cloud server, required when the ECS is accessed via the private network.
pvc_quantity	PVC specifications, for example, 5 GB, 10 GB, 20 GB.

Parameter	Description
refresh_interval	Auto hibernation time. Arm: 30 or 60 . x86: 30 , 60 , 240 , 1440 , or -1 . The unit of values except -1 (no hibernation) is minute. If no operation is performed on an instance for a period longer than the specified duration, the instance will be suspended (all saved data will not be deleted).
repository_id	(Optional) JointForce repository ID, required in the JointForce scenario.
stack_id	Stack ID, obtained via the ListStacks API.
task_type	Task type, which is used in the tutorial activity scenario.
token	(Optional) JointForce token, required in the JointForce scenario.
vpc_id	(Optional) VPC ID of the cloud server, required when the ECS is accessed via the private network.
instance_domain_id	Tenant ID of the authorized user of the instance.
instance_user_id	ID of the authorized user of the instance.

Table 4-123 Listing IDE instances of a tenant

Parameter	Description
org_id	Tenant ID (corresponding to domainId of the Huawei Cloud account).
region_id	Region ID.
is_temporary	Instance tag. Third-party systems need to set the tag together with CodeArts IDE Online. If this parameter is not transferred, the default value is used.
limit	Number of records displayed on each page. Value options: 10 , 15 , and 30 .
offset	Offset where the query starts.
search	Query by keyword (fuzzy query by instance name and description).

Table 4-124 Stopping an IDE instance

Parameter	Description
instance_id	Instance ID.

Parameter	Description
region_id	Region ID.

Table 4-125 Starting an IDE instance

Parameter	Description
instance_id	Instance ID.
region_id	Region ID.
plugin_enable_list	Plug-ins.
plugin_vars	Plug-in parameters. Encrypt sensitive information if needed.

Table 4-126 Deleting an IDE instance

Parameter	Description
instance_id	Instance ID.
region_id	Region ID.

Table 4-127 Modifying an IDE instance

Parameter	Description
instance_id	Instance ID.
region_id	Region ID.
description	Description.
display_name	Instance name, which can contain digits, letters, underscores (_), periods (.), and hyphens (-). Value length: 3–100 characters
refresh_interval	Auto hibernation time. Arm: 30 or 60 . x86: 30, 60, 240, 1440 , or -1 . The unit of all values except -1 (no hibernation) is minute. If no operation is performed on an instance for a period longer than the specified duration, the instance will be suspended (all saved data will not be deleted).

Table 4-128 Querying an IDE instance

Parameter	Description
instance_id	Instance ID.
region_id	Region ID.

Table 4-129 Checking whether the IDE instance name is duplicate

Parameter	Description
region_id	Region ID.
display_name	Instance name, which can contain digits, letters, underscores (_), dots (.), and dashes (-). Value length: 3–100 characters

4.4.18 CodeArts PerfTest

The CodeArts PerfTest connector interconnects with Huawei Cloud CodeArts PerfTest.

CodeArts PerfTest provides performance test services for cloud applications that are built based on HTTP, HTTPS, TCP, UDP, HLS, RTMP, WebSocket, or HTTP-FLV.

Prerequisites

You have subscribed to CodeArts PerfTest.

Creating a CodeArts PerfTest Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the CodeArts PerfTest connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Description	Enter the description of the connector to identify it.
Access Key	Access key ID (AK) of the current account. Obtain the AK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).
Secret Access Key	Secret access key (SK) of the current account. Obtain the SK by referring to Access Keys . If an AK/SK pair has been generated, find the downloaded AK/SK file (such as credentials.csv).

Action

- Querying a report
- Querying a task
- Modifying a task
- Deleting a task
- Modifying a project
- Querying a global variable
- Deleting a project
- Querying a task set
- Querying a project
- Creating a project
- Querying a project set

Configuring Parameters

For details about the parameters, see the [API Reference](#) of CodeArts PerfTest.

4.5 SaaS Applications

4.5.1 GitHub

The GitHub connector is used to connect to the GitHub platform.

GitHub is a hosting platform for open-source and private software projects. It is mainly used for version control and collaborative development. GitHub provides code hosting services based on its version control system, and other tools such as wiki, issue tracker, and code review.

Creating a GitHub Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the GitHub connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Token	GitHub account authorization token, created in Settings on the GitHub official website.
Description	Enter the description of the connector to identify it.

Action

- Adding a comment under a pull request
- Closing a pull request
- Creating an issue

Configuration Parameters

Table 4-130 Adding a comment under a pull request

Parameter	Description
Registry Name	Enter the name of the GitHub code repository.
Repository Owner	Enter the owner of the GitHub code repository.
Pull Request No.	Enter the ID of the pull request to which a comment will be added?
Content	Enter the content of the comment.

Table 4-131 Closing a pull request

Parameter	Description
Registry Name	Enter the name of the GitHub code repository.
Repository Owner	Enter the owner of the GitHub code repository.
Pull Request No.	Enter the ID of the pull request to be deleted.

Table 4-132 Creating an issue

Parameter	Description
Registry Name	Enter the name of the GitHub code repository.
Repository Owner	Enter the owner of the GitHub code repository.
Title	Enter the issue title.
Content	Enter the content of the new issue.

4.5.2 WordPress

The WordPress connector can push articles to websites and delete pushed articles based on requirements.

Creating a WordPress Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the WordPress connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
IP Address	Enter the WordPress address. Example: http://myblog.com .
Username	Enter the WordPress username.
Password	Enter the WordPress password.
Description	Enter the description of the connector to identify it.

Action

- Publishing an article
- Deleting an article

Configuration Parameters

Table 4-133 Publishing an article

Parameter	Description
Title	Title of the article to be published.
Parameter	Content of the article to be published.

Table 4-134 Deleting an article

Parameter	Description
ID	ID of the article to be deleted.

4.5.3 SAP

The SAP connector is used to interconnect with the SAP system.

The SAP system is a set of modern, information-based, and intelligent enterprise application software for enterprise resource planning (ERP). It provides reference for enterprise management and systematic planning for enterprise development. The SAP system is popular among enterprises for business modernization and informatization with streamlined process and less resources.

Creating an SAP connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the SAP connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Login Language	Select a language. <ul style="list-style-type: none"> • Chinese • English
SAP Server IP Address	Enter the IP address of the SAP server.
System ID	Enter the system ID.
SAP Group	Enter the SAP group name.
SAP Username	Enter the username for connecting to SAP.
Password	Enter the password for connecting to SAP.
Description	Enter the description of the connector to identify it.

Action

Accessing SAP functions

Configuration Parameters

Parameter	Mandatory	Description
SAP Function	Yes	Enter the name of the SAP function to be accessed.

Parameter	Mandatory	Description
Parameter	No	Enter the parameter name and value for accessing the function.

4.5.4 Jira Software

Jira is a popular and powerful project and issue tracking tool widely used in teams and organizations, including IT, operations, sales, marketing, and human resources. The main functions of Jira are as follows: project management, issue tracking, workflow management, reporting and analysis, and integration and expansion. Jira facilitates team management and collaboration and improves work efficiency and quality.

Creating a Jira Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Jira Software connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Type	<p>Select the type of the Jira connector.</p> <ul style="list-style-type: none"> ● Cloud: The Jira instance is deployed on the cloud. Users can use a browser to access it from anywhere. ● On-Premise: The Jira instance is installed locally. Jira is installed and configured on your own server or local PC.
Connection Address	<p>Enter the connection address of Jira.</p> <ul style="list-style-type: none"> ● When Type is set to Cloud, the format of the address is <code>https://your-domain.atlassian.net</code>. <i>your-domain</i> indicates the domain name of an organization or team. A valid Jira Cloud subscription is required to access the address. If you are not sure about your Jira Cloud address, obtain it from your Atlassian account or administrator. ● When Type is set to On-Premise, enter the IP address or host name of the server where Jira is located. Example: <code>http://192.168.0.1:8080</code> or <code>http://Jira.example.com:8080</code>

Parameter	Description
Authentication	Select the Jira authentication type. The options are Basic and API Token .
Username	Required only when Authentication is set to Basic Username for connecting to Jira
Password	Required only when Authentication is set to Basic Password for connecting to Jira
Email	Required only when Authentication is set to API Token Jira email address
API Token	Required only when Authentication is set to API Token Generate an API token in the personal settings in Jira. The API token is used for identity authentication and allows Jira users to access their accounts and related data via APIs.
Description	Enter the description of the connector to identify it.

Action

- Creating a project
- Updating a project
- Querying project details
- Deleting a project
- Creating an issue
- Modifying an issue
- Changing issue status
- Querying issues
- Querying issue details
- Searching for available users to assign issues
- Adding a comment
- Modifying a comment
- Querying comments

Configuration Parameters

Table 4-135 Creating a project

Parameter	Description
Project Name	Name of the project.
Project Key	Unique ID of the project.

Parameter	Description
ProjectTemplateType	Project template.
Project description	Description of the project.

Table 4-136 Updating a project

Parameter	Description
update by Project Key	Unique ID of the project.
Project Name	Name of the project.
Project description	Description of the project.

Table 4-137 Querying project details

Parameter	Description
Project Key	Unique ID of the project.

Table 4-138 Deleting a project

Parameter	Description
delete by Project Key	Unique ID of the project.

Table 4-139 Creating an issue

Parameter	Description
Project Type	Project type.
Project Key	Unique ID of the project.
Issue Type	Issue type.
Priority	Issue priority.
Summary	Issue summary.
Raised By	Enter a username.

Parameter	Description
Assign To	Enter a username.
Description	Description of the issue.

Table 4-140 Modifying an issue

Parameter	Description
Project Type	Project type.
Project Key	Unique ID of the project.
Name	Issue name or ID.
Issue Type	Issue type.
Priority	Issue priority.
Summary	Issue summary.
Raised By	Enter a username.
Assign To	Enter a username.
Description	Description of the issue.

Table 4-141 Changing issue status

Parameter	Description
Issue Name/ID	Issue name or ID.
Target Status	Select a status supported by the issue.

Table 4-142 Querying issues

Parameter	Description
Jql	Jira Query Language (JQL) expression.
Offset	Pagination offset. The default value is 0 .
Max. Records Returned	Pagination limit.

Table 4-143 Querying issue details

Parameter	Description
Issue Name/ID	Enter an issue name or ID.

Table 4-144 Searching for available users to assign issues

Parameter	Description
Project Key	Unique ID of the project (mandatory for new issues).
Issue Name/ID	Enter an issue name or ID (mandatory when Project Key is not specified).
Jira Username	Used for exact search.

Table 4-145 Adding a comment

Parameter	Description
Issue Name/ID	Enter an issue name or ID.
Visibility	Select a visible scope.
Description	Comment content.

Table 4-146 Modifying a comment

Parameter	Description
Issue Name/ID	Enter an issue name or ID.
Comment ID	ID of the comment.
Description	Comment content.
Visibility	Select a visible scope.

Table 4-147 Querying comments

Parameter	Description
Issue Name/ID	Enter an issue name or ID.
Offset	Pagination offset. The default value is 0 .
Max. Records Returned	Pagination limit.

4.5.5 WeCom

The WeCom connector allows you to use WeCom functions in a workflow after a few configurations.

WeCom is a communication and office tool designed for enterprises by Tencent. WeCom provides the same communication experience as WeChat, along with rich OA applications and capabilities to connect to the WeChat ecosystem, helping enterprises connect to internal systems, ecosystem partners, and consumers. WeCom features professional collaboration, security management, and people-as-a-service.

Creating a WeCom Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the WeCom connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Type	Select the connection type. <ul style="list-style-type: none"> • Chatbot: Message pushing by chatbot. • Application: Other functions.
Enterprise ID	Mandatory when the connection type is set to Application . Each enterprise has a unique corpId . For details, see "Basic Concepts" in the documentation for WeCom enterprise internal development server APIs.
AppSecret	Mandatory when the connection type is set to Application . Each application has an independent access secret. For details, see "Basic Concepts" in the documentation for WeCom enterprise internal development server APIs.
Chatbot Key	Mandatory when the connection type is set to Chatbot . Value of the request parameter key in the robot webhook.
Description	Enter the description of the connector to identify it.

Action

- Obtaining department IDs
- Creating a department
- Updating a department

- Deleting a department
- Obtaining member IDs
- Creating a member
- Updating a member
- Deleting a member
- Obtaining a group chat
- Creating a group chat
- Modifying a group chat
- Pushing a text message
- Pushing a markdown message
- Pushing a file message
- Pushing an image message
- Pushing an image and text message
- Pushing a text card message
- Withdrawing an application message
- Pushing a text message to a group chat
- Pushing a markdown message to a group chat
- Pushing a file message to a group chat
- Pushing an image message to a group chat
- Pushing a text and image message to a group chat
- Pushing a text card message to a group chat
- Pushing a text message by chatbot
- Pushing a markdown message by chatbot
- Pushing a file message by chatbot
- Pushing an image message by chatbot
- Pushing an image and text message by chatbot

Configuration Parameters

Table 4-148 Obtaining department IDs

Parameter	Description
Department ID	Obtains a specified department and its sub-departments (and sub-departments of sub-departments, recursively). If left empty, the full organizational structure is obtained by default.

Table 4-149 Creating a department

Parameter	Description
Dept Name	Department names at the same level must be unique. The value contains 1 to 32 UTF-8 characters and cannot contain the following characters: *?"<> :
English Name	Department names at the same level must be unique. This takes effect only after the multi-language function is enabled on the management backend. The value contains 1-64 characters, excluding the following characters: *?"<> :
Parent Dept ID	Parent department ID, which is a 32-digit integer.
Order	Sequence number in the parent department in descending order. Range: [0, 2 ³²).
Dept ID	If specified, the value must be a 32-digit integer greater than 1. If not specified, a default ID is generated.

Table 4-150 Updating a department

Parameter	Description
Dept ID	Department ID.
Dept Name	The value contains 1 to 32 UTF-8 characters and cannot contain the following characters: *?"<> :
English Name	This takes effect only after the multi-language function is enabled on the management backend. The value contains 1-64 characters, excluding the following characters: *?"<> :
Parent Dept ID	Parent department ID.
Order	Sequence number in the parent department in descending order. Range: [0, 2 ³²).

Table 4-151 Deleting a department

Parameter	Description
Dept ID	Department ID.

Table 4-152 Obtaining member IDs

Parameter	Description
Pagination Cursor	Cursor used for pagination query. The value is a character string and is returned by the last invoking. Leave this empty for the first invoking.
Total Data	Expected requested data volume. Range: 1 to 10000.

Table 4-153 Creating a member

Parameter	Description
Member ID	Member user ID. Unique management account in an enterprise. Max.: 64 characters. Start with a digit or letter and use only digits, letters, underscores (_), hyphens (-), at signs (@), and periods (.). This is case-insensitive.
Member Name	Max.: 64 UTF-8 characters
Alias	Max.: 64 UTF-8 characters
Gender	1: male; 2: female.
Enable Member	1: enabled; 0: disabled.
Major Dept ID	Major department ID.
Title	Position information. Max.: 128 characters.
Dept IDs	ID list of departments to which the member belongs. Max.: 100 IDs. Separate IDs with commas (,).
Sorting Members in Dept	The default value is 0. Members are sorted by earliest creation time. The number must match that of departments. Values are listed in descending order. Range: [0, 2^32). Separate values with commas (,).
Department Leaders?	The number must match that of departments. The value indicates whether the user is the leader of the department. 1 : leader; 0 : non-leader. It can be used to identify the upper-level approver in self-built or third-party applications. Separate values with commas (,).
Immediate Superior IDs	User IDs of enterprise members. Max.: 5 IDs. Separate IDs with commas (,).
Fixed-line Phone	The value consists of digits, hyphens (-), plus signs (+), or commas (,). Max.: 32 bytes.
Mobile Number	The value must be unique in an enterprise. Enter either the mobile number or email address, or both.

Parameter	Description
Email Address	An email address of 6 to 64 bytes. The value must be unique in an enterprise. Enter either the mobile number or email address, or both.
Enterprise Mailbox	Only for enterprises that have enabled the enterprise email function. An email address of 6 to 64 bytes, and must be unique in an enterprise. If not specified, the system generates a default enterprise email address for the user. (The system-generated email address can be changed once only.)
Address	Max.: 128 characters.

Table 4-154 Updating a member

Parameter	Description
Member ID	Member user ID. Unique management account in an enterprise. The value is case-insensitive and contains 1–64 bytes.
Member Name	Max.: 64 UTF-8 characters
Alias	Max.: 64 UTF-8 characters
Gender	1: male; 2: female.
Enable Member	1: enabled; 0: disabled.
Major Dept ID	Major department ID.
Job Title	Max.: 128 UTF-8 characters.
Dept IDs	IDs of departments to which the member belongs. Max.: 100 IDs. Separate IDs with commas (,).
Sorting Members in Dept	The default value is 0 . This parameter is valid when Dept IDs is transferred. The number must match that of departments. Values are listed in descending order. Range: [0, 2 ³²). Separate values with commas (,).
Department Leaders?	The number must match that of departments. The value indicates whether the member is the department leader. 1 : leader; 0 : non-leader. Separate values with commas (,).
Immediate Superior IDs	Enterprise members. Max.: 5. Separate IDs with commas (,).
Fixed-line Phone	The value consists of digits, hyphens (-), plus signs (+), or commas (,). Length: 1–32 bytes.

Parameter	Description
Mobile Number	Phone number, must be unique in an enterprise. If a member has enabled the WeCom account, they need to modify this parameter by themselves. (In this case, this parameter is ignored, and no error is reported.)
Email Address	An email address of up to 64 bytes. It must be unique in an enterprise. If the WeCom account is bound to a Tencent XMail address, this parameter can only be modified in XMail. (In this case, this parameter is ignored, and no error is reported.)
Enterprise Mailbox	This parameter is available only for enterprises that have enabled the enterprise email function. The value is an email address of 6 to 64 bytes, and must be unique in an enterprise. This can be modified once by members whose enterprise email address was system-generated.
Address	Max.: 128 characters.

Table 4-155 Deleting a member

Parameter	Description
Member ID	Member user ID.

Table 4-156 Obtaining a group chat

Parameter	Description
Group Chat ID	Group chat ID.

Table 4-157 Creating a group chat

Parameter	Description
Group Chat Name	Up to 50 UTF-8 characters. Any excess will be truncated.
Group Owner ID	ID of the specified group owner. If not specified, the system randomly selects a user from the user list as the group owner.
Group Member IDs	2 to 2000 members. Separate IDs with commas (,).
Group Chat ID	Unique ID of a group chat. The value is a string of up to 32 characters. Use only digits and uppercase or lowercase letters. If not specified, the system randomly generates a value.

Table 4-158 Modifying a group chat

Parameter	Description
Group Chat ID	Group chat ID.
Group Chat Name	New group chat name. If no update is required, ignore this parameter. Up to 50 UTF-8 characters. Any excess will be truncated.
Group Owner ID	ID of the new group owner. If no update is required, ignore this parameter. The group owner of a course group chat must be in the group owner list.
Member IDs to Add	IDs of members to be added.
Member IDs to Remove	IDs of members to be removed.

Table 4-159 Pushing a text message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If specified as @all , the message is sent to all members of the enterprise application.
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
Content	Message content. Max.: 2048 bytes. Any excess will be truncated. (ID translation is supported.)
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-160 Pushing a markdown message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If this parameter is set to @all , the message is sent to all members of the enterprise application.

Parameter	Description
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
Content	Content of the markdown message. The value is UTF-8 encoded and up to 2048 bytes.

Table 4-161 Pushing a file message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If this parameter is set to @all , the message is sent to all members of the enterprise application.
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
File ID	Can be obtained by invoking the API for uploading temporary materials.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-162 Pushing an image message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If this parameter is set to @all , the message is sent to all members of the enterprise application.
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
Image Media File ID	Can be obtained by invoking the API for uploading temporary materials.

Parameter	Description
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-163 Pushing an image and text message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If this parameter is set to @all , the message is sent to all members of the enterprise application.
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
Subject	Max.: 128 bytes. Any excess will be truncated.
Message Thumbnail	Thumbnail ID of an image and text message. Obtained via the material management API (value of media_id returned by the upload API).
Content	HTML tags are supported. Max.: 666 KB.
Message Author	Max.: 64 bytes.
Source Link	Page link of Read More .
Description	Max.: 512 bytes. Any excess will be truncated.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked; 2 : public within the enterprise.

Table 4-164 Pushing a text card message

Parameter	Description
Receiver ID	Member IDs for receiving the message. Separate multiple recipients by vertical bars (). Max.: 1,000 IDs. Special case: If this parameter is set to @all , the message is sent to all members of the enterprise application.
Recipient Dept ID	Department IDs for receiving the message. Separate recipients by vertical bars (). Max.: 100 IDs. Skip this parameter when the receiver is set to @all .

Parameter	Description
Enterprise App ID	Enterprise application ID. The value is an integer. It can be viewed on the settings page of the application.
Subject	Max.: 128 bytes. Any excess will be truncated.
Description	Max.: 512 bytes. Any excess will be truncated.
Redirection Link	Max.: 2,048 bytes (HTTP/HTTPS included).
Button Text	The default value is More . Max.: 4 characters. Any excess will be truncated.

Table 4-165 Withdrawing an application message

Parameter	Description
Message ID	Obtained from the response of the message sending API of the application.

Table 4-166 Pushing a text message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
Content	Max.: 2048 bytes. Any excess will be truncated.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-167 Pushing a markdown message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
Content	Content of the markdown message. The value is UTF-8 encoded and up to 2048 bytes.

Table 4-168 Pushing a file message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
File ID	Obtained by invoking the API for uploading temporary materials.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-169 Pushing an image message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
Image Media File ID	Can be obtained by invoking the API for uploading temporary materials.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked.

Table 4-170 Pushing a text and image message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
Subject	Max.: 128 bytes. Any excess will be truncated.
Message Thumbnail	Thumbnail ID of an image and text message. Obtained via the material management API (value of media_id returned by the upload API).
Content	Message content. HTML tags are supported. Max.: 666 KB.
Message Author	Max.: 64 bytes.
Source Link	Page link of Read More .
Description	Max.: 512 bytes. Any excess will be truncated.
Confidentiality	Whether the message is confidential. 0 (default): public; 1 : private and watermarked; 2 : public within the enterprise.

Table 4-171 Pushing a text card message to a group chat

Parameter	Description
Group Chat ID	The group specified by the chat ID must be created by the connected application.
Subject	Max.: 128 bytes. Any excess will be truncated.
Description	Max.: 512 bytes. Any excess will be truncated.
Redirection Link	Max.: 2,048 bytes (HTTP/HTTPS included).
Button Text	The default value is More . Max.: 4 characters. Any excess will be truncated.

Table 4-172 Pushing a text message by chatbot

Parameter	Description
Content	Max.: 2048 bytes. The excess part will be truncated.
User IDs to Mention	<i>@a member</i> : mention a specified member in the group; @all : mention all members. Separate IDs with commas (,).
Mobile Numbers to Remind	<i>@a member</i> : mention a specified member corresponding to the phone number; @all : mention all members. Separate numbers with commas (,).

Table 4-173 Pushing a markdown message by chatbot

Parameter	Description
Content	Content of the markdown message. The value is UTF-8 encoded and up to 4096 bytes.

Table 4-174 Pushing a file message by chatbot

Parameter	Description
File ID	Obtained via the group chatbot file upload API.

Table 4-175 Pushing an image message by chatbot

Parameter	Description
Base64-encoded Image Content	Base64-encoded image content.
MD5 Value	MD5 value of the image content before Base64 encoding.

Table 4-176 Pushing an image and text message by chatbot

Parameter	Description
Subject	Max.: 128 bytes. Any excess will be truncated.
Description	Max.: 512 bytes. Any excess will be truncated.
Redirection URL	Click to go to the redirect link.
Image URL	Image link of the image and text message. Available: JPG and PNG formats. Use 1068 x 455 and 150 x 150 sizes for better experience.

4.5.6 DingTalk

The DingTalk connector allows you to use DingTalk functions in a workflow after a few configurations.

DingTalk is an enterprise-level intelligent mobile office platform developed by Alibaba. It features office collaboration and application development in the digital era.

Creating a DingTalk Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the DingTalk connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
AppKey	Client ID in "application development" > "internal applications" > "application details" > "application information" in DingTalk.

Parameter	Description
AppSecret	Client Secret in "application development" > "internal applications" > "application details" > "application information" in DingTalk.
RobotCode	RobotCode in "application development" > "internal applications" > "application details" > "robot and message push" in DingTalk. Mandatory for the chatbot function.
Description	Enter the description of the connector to identify it.

Action

- Sending a link message by chatbot
- Sending a Markdown message by chatbot
- Sending an image message by chatbot
- Sending a file message by chatbot
- Sending an ActionCard message by chatbot
- Sending a text message by chatbot

Configuration Parameters

Table 4-177 Sending a link message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .
Subject	Message subject.
Content	Message content.
Image URL	Link to the preview image.
Redirection URL	Redirection link opened after clicking the message.

Table 4-178 Sending a Markdown message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .

Parameter	Description
Subject	Message subject.
Content	Message content.

Table 4-179 Sending an image message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .
Image URL	Link to the preview image.

Table 4-180 Sending a file message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .
Media File ID	Value of media_id of the API for uploading media files.
File Name	Name of the file to be sent.
File Type	xlsx, pdf, zip, rar, doc and docx files are supported.

Table 4-181 Sending an ActionCard message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .
Subject	Message subject.
Content	Message content.
Redirect Button	Text displayed on the redirect button.
Redirection URL	Redirection link opened after clicking the message.

Table 4-182 Sending a text message by chatbot

Parameter	Description
Group Chat ID	Obtain the value from the DingTalk API for creating an enterprise internal group or other groups. For an existing group, obtain the value by calling the JSAPI biz.chat.chooseConversationByCorpld .
Content	Message content.

4.5.7 Tencent Questionnaire

The Tencent Questionnaire connector allows you to use Tencent Questionnaire functions in a workflow after a few configurations.

Tencent Questionnaire offers end-to-end professional online survey services, including questionnaire design, data collection, and statistical analysis.

Creating a Tencent Questionnaire Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Tencent Questionnaire connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
ApplicationID	Enter the application ID of Tencent Questionnaire.
AppSecret	Enter the application secret of Tencent Questionnaire.
Description	Enter the description of the connector to identify it.

Action

- Querying answer details
- Querying questionnaire details
- Querying questionnaire statistics

Configuring Parameters

Table 4-183 Querying answer details

Parameter	Description
Questionnaire ID	Enter the questionnaire ID.
Answer ID	Enter the answer ID.

Table 4-184 Querying questionnaire details

Parameter	Description
Questionnaire ID	Enter the questionnaire ID.

Table 4-185 Querying questionnaire statistics

Parameter	Description
Questionnaire ID	Enter the questionnaire ID.

4.5.8 WJX Questionnaire

The WJX connector allows you to use WJX Questionnaire functions in a workflow after a few configurations.

WJX is a professional online platform for questionnaires, exams, assessments, and voting. It offers users intuitive tools for designing surveys, collecting data, customizing reports, and analyzing results.

Creating a WJX Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the WJX connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
AppID	Enter the AppID of WJX.

Parameter	Description
AppKey	Enter the AppKey of WJX.
Description	Enter the description of the connector to identify it.

Action

- Update the questionnaire status
- Listing questionnaires
- Querying questionnaire details

Configuring Parameters

Table 4-186 Update the questionnaire status

Parameter	Description
Questionnaire No.	Enter the questionnaire number.
Unix Timestamp	Used to check the validity period of a request, which is limited to 30 seconds after it is sent. Enter the Unix timestamp when the request is called. The Unix timestamp is the total number of seconds from 1970-01-01 00:00:00 GMT to the time when the request is called.
Whether to use the cache	Whether to use the cache for query APIs. The default value is Yes . <ul style="list-style-type: none"> • Yes • No
Questionnaire Status	Select a questionnaire status. <ul style="list-style-type: none"> • Publish or run • Paused • Removed

Table 4-187 Listing questionnaires

Parameter	Description
Unix Timestamp	Used to check the validity period of a request, which is limited to 30 seconds after it is sent. Enter the Unix timestamp when the request is called. The Unix timestamp is the total number of seconds from 1970-01-01 00:00:00 GMT to the time when the request is called.

Parameter	Description
Whether to use the cache	Whether to use the cache for query APIs. <ul style="list-style-type: none"> • Yes (default) • No
Questionnaire Publisher	If this parameter is left blank, the questionnaire published by the main account is obtained. The WeCom subaccount must be prefixed with qw\$.
Questionnaire No.	Enter the questionnaire number.
Folder Name	Get questionnaires in the folder.
Keyword for fuzzy query of a questionnaire name	The value must not contain Chinese or special characters and cannot exceed 10 letters. For questionnaires with names that include fuzzy keywords, an exact match is required.
Query Start Timestamp	Queries questionnaires created on and after the specified timestamp (in Unix format, representing the total milliseconds since January 1, 1970, 00:00:00 GMT).
Query End Timestamp	Queries questionnaires created before the specified timestamp (in Unix format, representing the total milliseconds since January 1, 1970, 00:00:00 GMT).
Paging page number	The default value is 1 , indicating that the first page is returned.
Page Size	Default: 10 ; maximum: 300 . If the value exceeds the maximum, the default value will be used.
Obtain all questionnaires of the enterprise	Yes No (default): Only questionnaires of the main account will be obtained. This parameter is valid only when the creator is the main account.
Only obtain the starred questionnaire	Yes No (default): Both star-mark questionnaires and non-star-mark questionnaires will be obtained.
Query by time	Select a query type. <ul style="list-style-type: none"> • Query not by time • By questionnaire start time • By questionnaire creation time

Parameter	Description
Sorting Rules	Select a sorting rule. <ul style="list-style-type: none"> • Ascending by Questionnaire No. • Descending by Questionnaire No. • Ascending by questionnaire start time • Descending by questionnaire start time • Ascending by questionnaire creation time • Descending by questionnaire creation time
Questionnaire Status	Get questionnaires in a specified state. <ul style="list-style-type: none"> • Not released • Released • Paused • Deleted • Complete Delete • Audited
Questionnaire Review Status	Get questionnaires in the specified review state. <ul style="list-style-type: none"> • Passed • Under review • Not Passed • Waiting for real name

Table 4-188 Querying questionnaire details

Parameter	Description
Questionnaire No.	Enter the questionnaire number.
Unix Timestamp	Used to check the validity period of a request, which is limited to 30 seconds after it is sent. Enter the Unix timestamp when the request is called. The Unix timestamp is the total number of seconds from 1970-01-01 00:00:00 GMT to the time when the request is called.
Whether to use the cache	<ul style="list-style-type: none"> • Yes (default) • No
Obtain the question information	<ul style="list-style-type: none"> • Yes (default) • No
Obtain the question option information	<ul style="list-style-type: none"> • Yes (default) • No

Parameter	Description
Obtain the question and answer option list	<ul style="list-style-type: none"> • Yes • No (default)
Obtain the question setting information	<ul style="list-style-type: none"> • Yes • No (default)
Obtain the pagination information	<ul style="list-style-type: none"> • Yes • No (default)
Simple data is returned	<p>Whether to return simple data</p> <ul style="list-style-type: none"> • Yes (Obtain the question information, Obtain the question option information, Obtain the question and answer option list, Obtain the question setting information, Obtain the pagination information do not take effect) • No (default)
Whether the data is in JSON format	<p>This parameter is valid only when Whether to return simple data is set to Yes.</p> <p>Whether to return data in JSON format.</p> <ul style="list-style-type: none"> • Yes • No (default)
Return the questionnaire title	<p>This parameter is valid only when Whether the data is in JSON format is set to Yes.</p> <ul style="list-style-type: none"> • Yes • No (default)

4.5.9 Polarion

The Polarion connector allows you to use Polarion functions in a workflow after a few configurations.

Polarion is a full lifecycle management platform for software development and product management. It provides tools and functions for requirement management, defect tracking, version control, test management, and project management. Polarion accelerates development and improves product quality by facilitating team collaboration and management throughout the development process.

Creating a Polarion Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.

3. Select the Polarion connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Connection Address	Enter the connection address. Example: http://endpoint:port/polarion .
API Token	Enter the API token.
Description	Enter the description of the connector to identify it.

Action

- Listing software requirements
- Updating software requirement status
- Listing test cases
- Listing software test cases
- Listing associations between test cases and requirements
- Adding a software test report
- Updating a software test report
- Listing issues

Configuration Parameters

Table 4-189 Listing software requirements

Parameter	Description
Project ID	Enter the project ID of the requirement.

Table 4-190 Updating software requirement status

Parameter	Description
Project ID	Enter the project ID of the requirement.
Work Item ID	Enter the work item ID.
Status	Select a requirement status. <ul style="list-style-type: none"> • Draft • Reviewed • Approved • Rejected

Parameter	Description
Resolution	Select a rejection reason. <ul style="list-style-type: none"> • Obsolete • Duplicate • Invalid • Later

Table 4-191 Listing test cases

Parameter	Description
Project ID	Enter the project ID of the requirement.

Table 4-192 Listing software test cases

Parameter	Description
Project ID	Enter the project ID of the requirement.

Table 4-193 Listing associations between test cases and requirements

Parameter	Description
Project ID	Enter the project ID of the requirement.
Test Case ID	Enter the test case ID.

Table 4-194 Adding a software test report

Parameter	Description
Project ID	Enter the project ID of the requirement.
Space ID	Enter the space ID.
Title	Enter the report title.
Name (ID)	Enter the name (ID).
Type	Select a work item type. <ul style="list-style-type: none"> • Software Test Case • est Case

Parameter	Description
Link Role	Select a role. <ul style="list-style-type: none"> • has parent • duplicates • relates to
Content Type	Select a content type. <ul style="list-style-type: none"> • text/html • text/plain
Content	Enter the report content.

Table 4-195 Updating software test report

Parameter	Description
Project ID	Enter the project ID of the requirement.
Space ID	Enter the space ID.
Title	Enter the report title.
Name (ID)	Enter the name (ID).
Type	Select a work item type. <ul style="list-style-type: none"> • Software Test Case • est Case
Content Type	Select a content type. <ul style="list-style-type: none"> • text/html • text/plain
Content	Enter the report content.

Table 4-196 Listing issues

Parameter	Description
Project ID	Enter the project ID of the requirement.

4.6 Emails

4.6.1 QQ Mail

The QQ Mail connector is used to interconnect with QQ Mail to send emails.

Tencent's QQ Mail is a popular free email service in China. Users can use QQ Mail to send and receive emails with attachments, manage schedules and memos, and set security protection measures such as email and spam filtering.

Prerequisites

You have obtained the QQ Mail authorization code. For details, see [Configuring a QQ Mail Authorization Code](#).

Creating a QQ Mail Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the QQ Mail connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Email Address	QQ Mail address of the sender.
Authorization Code	Authorization code of the QQ Mail sender, which is obtained from the settings in the QQ Mail personal center. For details, see Configuring a QQ Mail Authorization Code .
Description	Enter the description of the connector to identify it.

Action

Sending an email

Configuration Parameters

Parameter	Description
To	Enter the recipients' email addresses. Use commas (,) to separate.
Copy To	Enter the CC email addresses. Use commas (,) to separate.
Bcc	Enter the BCC email addresses. Use commas (,) to separate.
Subject	Enter the email subject.
Content	Enter the email content.

Configuring a QQ Mail Authorization Code


1. Log in to QQ Mail (<https://en.mail.qq.com/>) in a browser and go to the QQ Mail home page.
2. On the home page, go to **Settings > General**.
3. Find **IMAP/SMTP**, and click **Generate Authorization Code**.

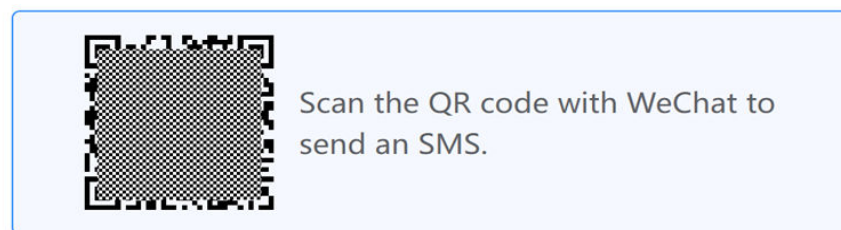
Third-party Services (It may be risky to log in to QQ mailbox from a third party. Login via [QQ Mail App](#) will be more secure.)

IMAP/SMTP : Enabled [Generate Authorization Code](#) [Disable Service](#)

4. Perform SMS verification as required, and click **SMS Sent**.

SMS Verification

Use your phone  to send an SMS and click SMS Sent for verification.



The QR code doesn't work? Try to [Manually send SMS](#) (SMS fees are charged by the mobile network operator.)

Select another verification method 

SMS Sent

5. The authorization code is displayed.

4.6.2 163 Mail

The 163 Mail connector is used to interconnect with 163 Mail to send emails.

NetEase's 163 Mail is one of the first free email services in China. Users can send and receive emails through web pages, mobile apps, and POP3/SMTP. 163 Mail features a simplified interface, high usability, and strong security, along with large storage and versatile functions, such as mail filtering, spam blocking, and mail archiving. In addition, 163 Mail provides value-added services such as the enterprise mailbox and VIP mailbox for different user requirements.

Prerequisites

You have obtained the 163 Mail authorization code. For details, see [Configuring a 163 Mail Authorization Code](#).

Creating a 163 Mail Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the 163 Mail connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Email Address	163 Mail address of the sender.
Authorization Code	Authorization code of the 163 Mail sender, which is obtained from the settings in the 163 Mail personal center. For details, see Configuring a 163 Mail Authorization Code .
Description	Enter the description of the connector to identify it.

Action

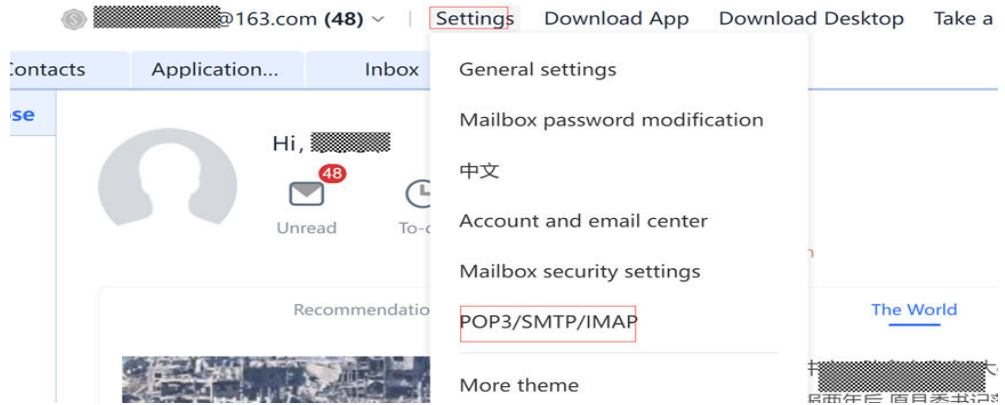
Sending emails

Configuration Parameters

Parameter	Description
To	Enter the recipients' email addresses. Separate addresses with commas (,).
Copy To	Enter the CC email addresses. Separate addresses with commas (,).
Bcc	Enter the BCC email addresses. Separate addresses with commas (,).
Subject	Enter the email subject.
Content	Enter the email content.

Configuring a 163 Mail Authorization Code

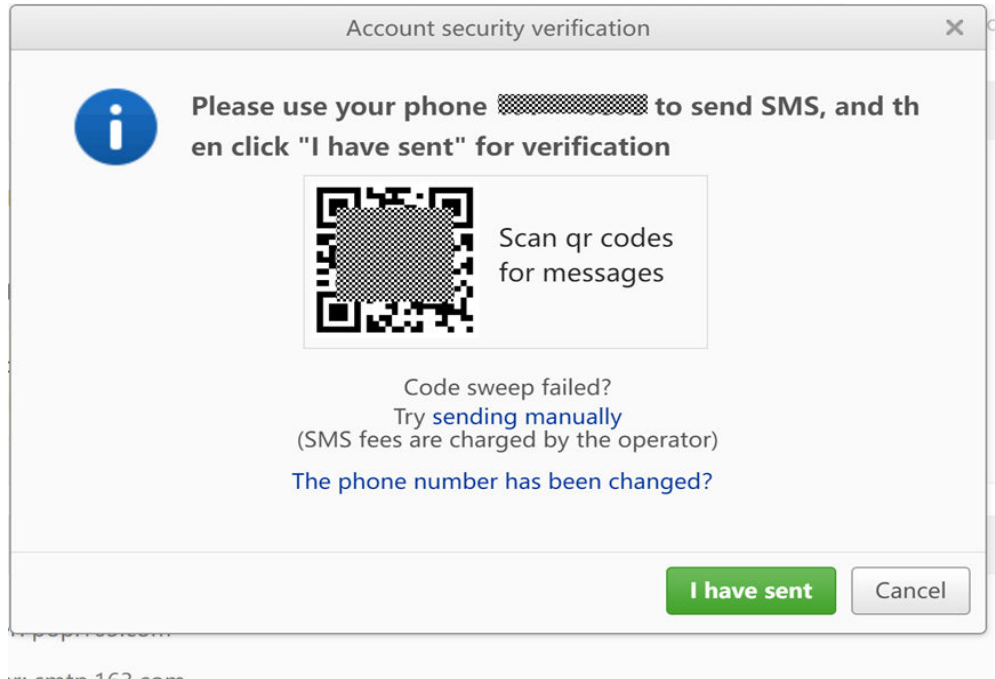
1. Log in to 163 Mail (<https://hw.mail.163.com/>) in a browser and go to the 163 Mail home page.
2. Choose **Settings** > **POP3/SMTP/IMAP**.



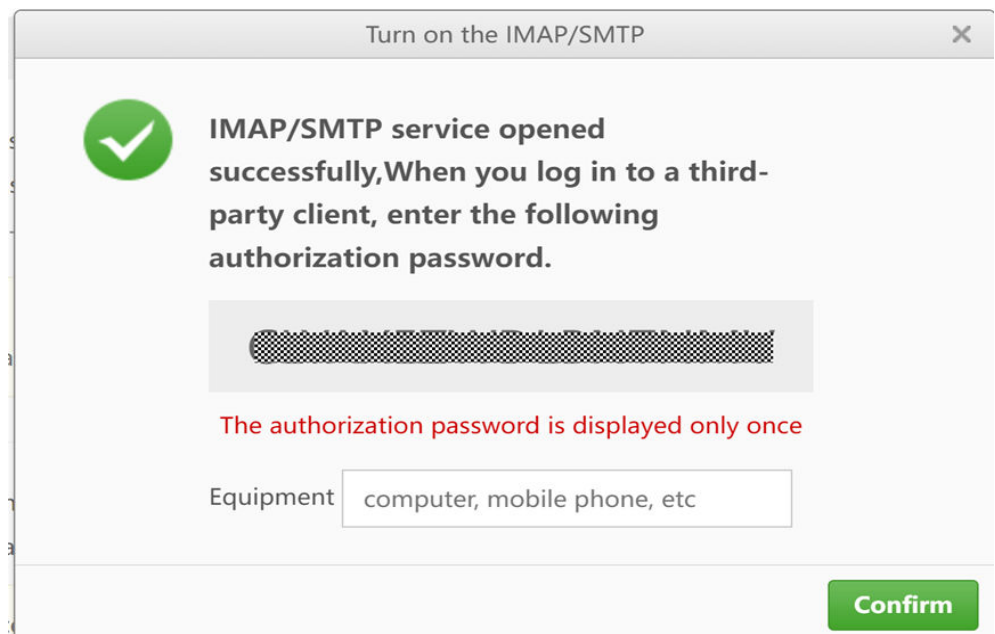
3. Enable IMAP/SMTP service.

A composite image showing a security notice and the POP3/SMTP/IMAP settings page. The top part is a banner with a smartphone icon, text: 'Please use the official NetEase email client to ensure accounts security', and a QR code. Below is the 'POP3/SMTP/IMAP' settings page. It shows 'Enable service:' with 'IMAP/SMTP service' and 'POP3/SMTP service' both set to 'Disabled | Enable' (the 'Enable' link is highlighted). A tip box states: 'Tips: if you log in the netease mailbox at a third party, there may be a risk of email leakage and even harm the security of Apple ID.'

4. Complete the account security verification as required, and click **I have sent**.



5. The 163 Mail authorization code is displayed.



4.7 Others

4.7.1 FTP

The FTP connector is a channel for file transfer based on the FTP/SFTP protocol. Through the FTP connector, files can be exchanged with an external FTP server.

- The File Transfer Protocol (FTP) is in the TCP/IP protocol group. It consists of two parts: FTP server and FTP client. The FTP server is used to store files. Users can use the FTP client to access resources on the FTP server.
- The SSH File Transfer Protocol (SFTP) is a secure version of FTP and a part of the SSH protocol. SFTP enables easy data transmission and data access through the data flow shell. SFTP, also known as the SSH file transfer protocol, provides a secure connection to transfer files and traverses file systems on local and remote systems.

By default, FTP uses TCP ports 20 and 21. Port 20 is used to transmit data, and port 21 is used to transmit control information (commands). Whether to use port 20 as the port for data transmission depends on the FTP transmission mode. In the active mode, the data transmission port number is 20. In the passive mode, the port is determined after the negotiation between the server and the client.

Creating an FTP Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the FTP connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Protocol	Protocol used by the FTP connector. Options are FTP and SFTP .
Host IP	IP address of the FTP server.
Port	By default, FTP uses TCP ports 20 and 21. Port 20 is used to transmit data, and port 21 is used to transmit control information (commands).
Username	FTP username.
Password	FTP user password.
Description	Enter the description of the connector to identify it.

Action

Writing data to a file

Configuration Parameters

Parameter	Description
Path	Enter the relative path of the file on the FTP server.
File Name	Enter the name of the target file.

Parameter	Description
Content	Enter the content to be written.
Write Policy	<p>Select a policy.</p> <ul style="list-style-type: none"> • Overwrite: Overwrite the content of the file with the same name. • Append: Add content to the end of the file with the same name. • Backup: Back up the file with the same name. The suffix is the execution time.

4.8 Message Systems

4.8.1 ActiveMQ

ActiveMQ is an open-source message middleware based on the Java Message Service (JMS). ActiveMQ provides a reliable, efficient, and scalable message transfer mechanism and supports multiple message and transmission protocols, such as TCP, UDP, SSL, and NIO. ActiveMQ supports multiple queue and topic modes and can be used in scenarios such as asynchronous message transfer, publishing/subscription, and message routing. It provides a visualized management interface, facilitating configuration and monitoring with high reliability and flexibility. ActiveMQ can be used to build distributed systems. It supports multiple programming languages and platforms and provides various APIs.

Creating an ActiveMQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the ActiveMQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Brokers	List brokers of the ActiveMQ.
Username	Username for connecting to ActiveMQ.
Password	Password for connecting to ActiveMQ.
SSL Authentication	Whether SSL authentication is used for connecting ActiveMQ.
Description	Enter the description of the connector to identify it.

Action

Sending messages

Configuration Parameters

Parameter	Description
Destination Object Type	Type of the destination object to which a message is sent. <ul style="list-style-type: none"> • Topic • Queue
Destination Object Name	Enter the name of the destination object, that is, the name of the topic or queue.
Content	Enter the content of the message to be sent.

4.8.2 ArtemisMQ

ArtemisMQ is an open-source high-performance message queue system built on Apache ActiveMQ and Apache Artemis. ArtemisMQ provides a reliable asynchronous message transfer mechanism that allows different applications to communicate with each other through messages. It uses the message-based middleware mode, allowing producers (senders) to send messages to queues or topics, while consumers (receivers) can receive and process these messages from queues or topics.

Creating an ArtemisMQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the ArtemisMQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
IP Address	Enter the IP address of ArtemisMQ.
Port	Port number for connecting to ArtemisMQ
Username	Username for connecting to ArtemisMQ
Password	Password for connecting to ArtemisMQ
SSL Authentication	Whether SSL authentication is used for connecting to ArtemisMQ.
Description	Enter the description of the connector to identify it.

Action

None

Configuration Parameters

None

4.8.3 IBM MQ

IBM MQ is a message queue software for asynchronous communication in a distributed system to secure reliable data transmission between applications, platforms, and systems. It serves as a middleware and prevents message loss, duplication, or disorder during transmission, and converts and formats data between different systems.

Creating an IBM MQ Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the IBM MQ connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
IP Address	Enter the IP address of the IBM MQ host.
Port	Enter the port number of the IBM MQ host.
Queue Manager	A system management program that provides queue services, APIs, and queue management services.
CCSID	Character set used to encode character strings in messages.
Channel Name	Channel for transferring messages between queue managers in the MQ system.
Username	User name for connecting to the IBM MQ.
Password	User password for connecting to the IBM MQ.
SSL Enable	Whether to use SSL authentication when connecting to the IBM MQ. <ul style="list-style-type: none"> • True • False
Cipher Suite	Mandatory when SSL Enable is set to True . Enter the cipher suite.

Parameter	Description
Trust Store	Mandatory when SSL Enable is set to True . Enter the Base64-encoded trust store file.
Trust Store Password	Mandatory when SSL Enable is set to True . Enter the password of the trust store.
Description	Enter the description of the connector to identify it.

Action

Sending messages

Configuration Parameters

Table 4-197 Sending messages

Parameter	Description
Destination Object Type	Select the destination object type. <ul style="list-style-type: none"> • Topic • Queue
Destination Object Name	Enter the topic/queue name of the destination object.
Content	Enter the message content.

4.8.4 Kafka

Kafka is a distributed streaming platform developed by LinkedIn. It features high throughput and low latency for massive amounts of real-time data streaming. Kafka consists of producers, consumers, and brokers. Producers send data to the Kafka cluster; consumers subscribe to the cluster and process the data; brokers are the core component of a Kafka cluster as they are responsible for storing and forwarding messages. Featuring scalability, high throughput, low latency, reliability, and durability, Kafka is widely used for big data processing, real-time data streaming, and log collection.

Creating a Kafka Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click **New Connection**.
3. Select the Kafka connector.
4. In the dialog box displayed, configure the connector and click **OK**.

Parameter	Description
Name	Enter the connector instance name.
Broker Address	Enter the Kafka broker address.
Authentication Mode	Select the authentication mode of Kafka. <ul style="list-style-type: none"> • SSL • None
SASL	Mandatory when Authentication Mode is set to SSL . Select the SASL mechanism. <ul style="list-style-type: none"> • PLAIN • SCRAM-SHA-512
SSL Username	Mandatory when Authentication Mode is set to SSL . Enter the SSL username or application key.
SSL Password	Mandatory when Authentication Mode is set to SSL . Enter the SSL password.
SSL Certificate Format	Mandatory when Authentication Mode is set to SSL . Select the SSL certificate format. <ul style="list-style-type: none"> • PEM • JKS
SSL Certificate	Mandatory when Authentication Mode is set to SSL . Enter the Base64-encoded certificate content.
SSL Certificate Password	Mandatory when Authentication Mode is set to SSL . Enter the SSL certificate password.
Description	Enter the description of the connector to identify it.

Action

Writing messages

Configuration Parameters

Table 4-198 Sending messages

Parameter	Description
Topic	Enter the topic to be listened.
Partition	Partition to which the message will be written.
Key	Key value of the message.
Message	Message content.

Parameter	Description
Acks	<p>Producer acknowledgement.</p> <ul style="list-style-type: none"> • all • -1 • 0 • 1
KeySerializer	<p>Method of turning the key into bytes. The default value is org.apache.kafka.common.serialization.StringSerializer.</p>
ValueSerializer	<p>Method of turning the value into bytes. The default value is org.apache.kafka.common.serialization.StringSerializer.</p>
BufferMemorySize	<p>Total bytes of memory the producer can use to buffer records waiting to be sent to the server. The default value is 33554432.</p>
Retries	<p>Number of retries.</p>

4.9 Custom Connectors

4.9.1 OpenAPI

You can create a custom OpenAPI connector and import OpenAPI files in JSON or YAML format. Currently, OpenAPI 2.0 and 3.0 are supported.

The OpenAPI connector supports multiple authentication modes, connection types, and custom actions and action parameters.

Creating an OpenAPI Connector

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click the **Custom Connector** tab.
3. Click **New Connector**. The **New OpenAPI Connector** dialog box is displayed.
4. Select a file to be imported.

Table 4-199 Description for file importing

Parameter	Description
Select an OpenAPI file.	<ul style="list-style-type: none"> • Select the OpenAPI version. You can select 2.0 or 3.0. • Click Download Template to download the OpenAPI template file of the corresponding version. • Modify the template file based on service requirements and upload the file.

Parameter	Description
Authentication Type	<p>Select the authentication type required when the connector is used to create a connection. The authentication type is extracted from the security mode in the OpenAPI specification.</p> <p>The template file for version 3.0 is used as an example. The following authentication types are available:</p> <ul style="list-style-type: none"> • api_key: API key authentication. Authentication parameters can be modified in this case. • x-bfs-iam-token: IAM user token authentication. • x-bfs-iam-aksk: IAM user AK/SK authentication. • Basic: Basic authorization. • None: No authentication is required.
Action	<p>Select the action to be configured. The action is extracted from the API in the OpenAPI specification. After a connector is generated, you can add a trigger.</p>

5. Configure basic information.

Table 4-200 Parameter description

Parameter	Description
Icon	<p>Icon for the custom connector.</p> <p>Click Upload Icon to upload an icon from the local host. If this parameter is not uploaded, a default icon is generated based on the initial letter of the connector name.</p>
Name	Enter the connector name.
Tag	Enter a tag for the connector.
Type	<p>Select a connector type.</p> <ul style="list-style-type: none"> • HTTP • Database • Huawei Cloud service • SaaS application • Email • Other • Message system
Description	Enter the description of the connector to identify it.

6. Click **Create and Configure Details**. The action configuration page is displayed.




- Click . In the pop-up box, configure action information and click **OK**.

Table 4-201 Parameter description

Parameter	Description
Name	Enter the action name.
Type	Select an action type. <ul style="list-style-type: none"> • GET • POST • PUT • DELETE
Path	Enter a path, for example, /history .
Description	Enter the action description.

- After the action is added, configure action parameters.

Table 4-202 Parameter description

Parameter	Description
Input Parameter	Options are Query , Header , and Path . <ul style="list-style-type: none"> • Name: Enter the parameter name. • Display Name: Enter the parameter display name. • Type: Select the parameter type. Options are string, integer, and number. • Mandatory: Whether a parameter is mandatory. • Description: Enter the parameter description. Click  in the Operation column to add a parameter.
Body	Configure request body parameters. When the action type is POST , PUT , or DELETE , request body parameters are required. <ul style="list-style-type: none"> • Name: Enter the parameter name. • Display Name: Enter the parameter display name. • Type: Select the parameter type. Options are object, string, integer, and number. • Mandatory: Whether a parameter is mandatory. • Description: Enter the parameter description. Click  in the Operation column to add a parameter. <p>NOTE By default, the name of the first parameter is a root node, and the parameter type is object. Both of them cannot be modified.</p>

Parameter	Description
Return Value	<p>Configure return value parameters. Click + in the Return Value column and add different return values in the drop-down list.</p> <ul style="list-style-type: none"> • Name: Enter the parameter name. • Display Name: Enter the parameter display name. • Type: Select the parameter type. Options are object, array, string, and integer. • Mandatory: Whether a parameter is mandatory. • Description: Enter the parameter description. <p>Click + in the Operation column to add a parameter.</p> <p>NOTE By default, the name of the first parameter is a root node, and the parameter type is object. Both of them cannot be modified.</p>

9. Click **Save**.

Creating an OpenAPI Connection

1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Connector**. On the page displayed, click the **Custom Connector** tab.
3. Click **New Connection** in an OpenAPI connector.
4. In the pop-up box displayed, configure the information.

Table 4-203 Parameter description (Basic authentication)

Parameter	Description
Name	Enter the connection name.
URL	Enter the connection address.
Username	Enter a username.
Password	Enter the password.
Description	Enter a description.

NOTE

In this example, the OpenAPI connector uses the basic authentication. OpenAPI connectors using other types of authentication require corresponding authentication parameters when creating a connection.

If a custom connector uses no authentication, no connection is required.

5. Click **OK**.

5 Processors

- [5.1 Conditions](#)
- [5.2 Parallel Processor](#)
- [5.3 Delay](#)
- [5.4 EDI Processor](#)
- [5.5 Variable Assignment](#)
- [5.6 Variable Updater](#)
- [5.7 Sorter](#)
- [5.8 Data Conversion](#)
- [5.9 Splitter](#)
- [5.10 Filter](#)
- [5.11 Script](#)
- [5.12 Message Logger](#)
- [5.13 Data Mapper](#)
- [5.14 Iterator](#)
- [5.15 Error Monitoring](#)
- [5.16 End](#)
- [5.17 Symmetric Encrypt/Decrypt](#)
- [5.18 Asymmetric Encrypt/Decrypt](#)
- [5.19 XML Verification](#)

5.1 Conditions

The conditioning processor is used to create condition branches to determine the execution logic of subsequent nodes. You need to configure the conditions of each

branch on the connection line between the **Conditions** node and subsequent nodes.

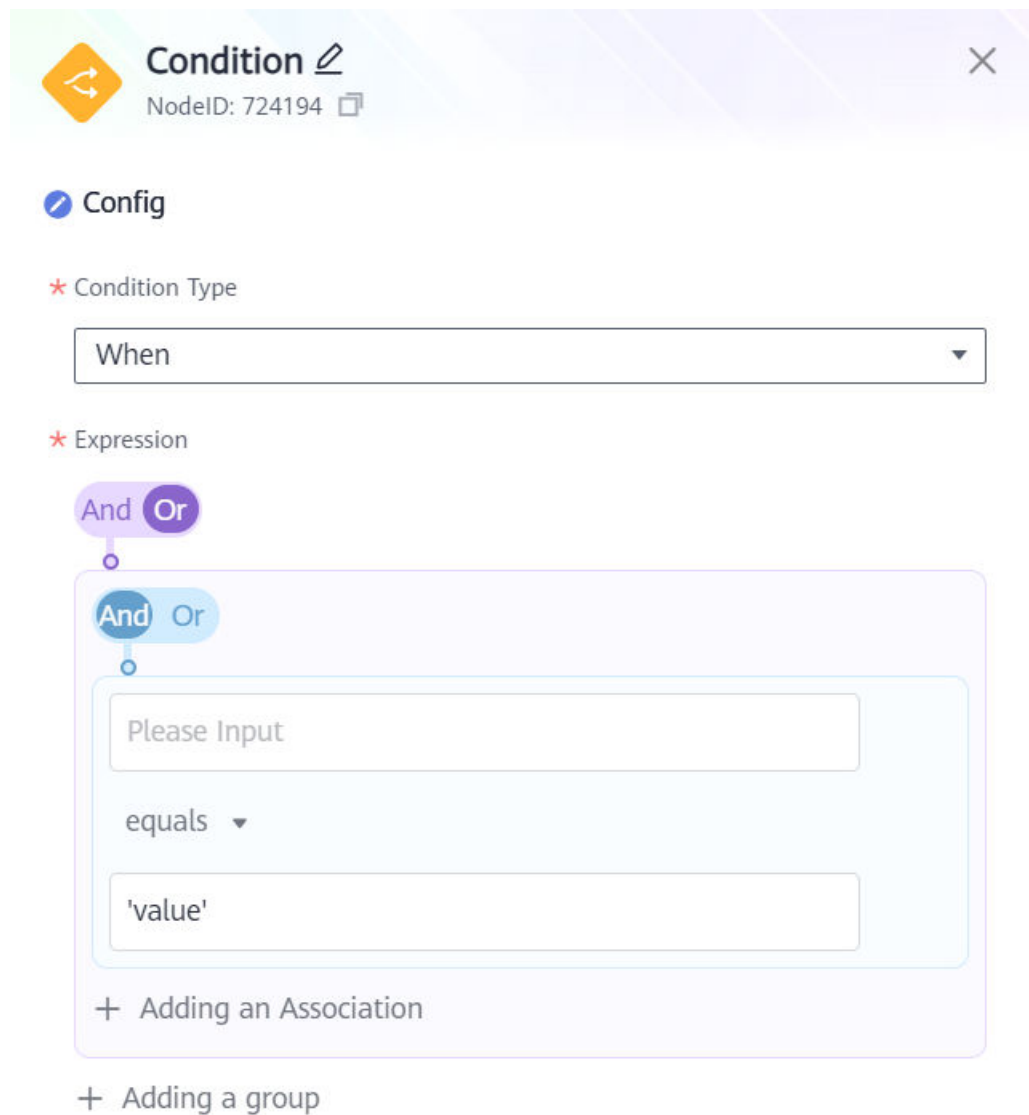
Constraints

- In a **Conditions** node, if there are multiple expressions of the same type that meet the **When** condition, the first created branch is executed.
- In a **Conditions** node, only one branch of the **Otherwise** condition type is allowed.

Configuration Parameters

Parameter	Description
Condition Type	<p>There are two condition types: When and Otherwise.</p> <ul style="list-style-type: none"> • When: If the current condition is met, the branch is executed when the data transferred to the Conditions node meets the specified expression. • Otherwise: If other conditions are not met, the branch is executed when the data transferred to the Conditions node does not meet the conditions of all other branches.
Expression	<p>Expression of the current condition branch.</p> <p>As shown in Figure 5-1, the AND and OR logical operators in box 1 indicate the association between outer logical groups. Click Add Group in box 5 to add an outer logical group.</p> <p>The logical operator in box 2 indicates the association relationship between inner logic groups. Click Add Associations in box 4 to add an inner logic group.</p> <p>The drop-down list in box 3 indicates the logic of the value in the two text boxes. Options are equals, not equals, less than, not less than, larger than, and not larger than.</p>

Figure 5-1 Branching conditions



5.2 Parallel Processor



The parallel processor is used to create multiple parallel processing branches so that multiple branch tasks can be executed at the same time. The merged branch execution result can be referenced using the ROMA expression **$\${body}$** in subsequent steps.


Configuration Parameters


Parameter	Description
Merge Output Format	<p>Format of the merge result. Currently, only JSON is supported, indicating that the merge result is output in JSON character string format.</p> <p>You can use either of the following methods to obtain the result of a branch:</p> <ul style="list-style-type: none"> If the subsequent node is Script, use JSON.parse('\${body}') to convert the merge result into a JSON object, and then obtain the branch result based on the branch name. Example: <pre>var result=JSON.parse('\${body}');</pre> <pre>var branch_result=result.<i>branch name</i></pre> or <pre>var branch_result=result['<i>branch name</i>']</pre> If the subsequent node is Variable Assignment, Conditions, or Data Conversion, use \$.<i>branch name</i> to obtain the execution result of the branch.
Stop On Failure	<p>Whether to stop the task when an error occurs during parallel processing.</p> <ul style="list-style-type: none"> True: When an error occurs in any parallel processor branch, the entire task stops and an error message is returned. False: When an error occurs in the parallel processor branch, the entire task continues to be executed on subsequent nodes.
Timeout (ms)	<p>Maximum execution time of the parallel processing, within which if a branch task is not completed, the branch task will be discarded and excluded in the final merge result.</p> <p>If this parameter is set to 0, the system waits till all branch tasks are complete.</p>

NOTE


The branch name of a **Parallel Processor** node needs to be configured on the connection line between adjacent nodes.

 **Parallel Processor** 



NodeID: 158426 

 **Config**

Merge Output Format

Json 

Stop On Failure

False 

* Timeout (Milliseconds) 




0


5.3 Delay



The delayer processor is used to create a delay of a specified time. A typical application scenario is as follows: If a step is a time-consuming task and the subsequent step depend on the execution result of the step, you can add the delayer processor between the two steps to ensure that the subsequent step is executed properly.

Configuration Parameters

Parameter	Description
Latency (ms)	Delay duration, in milliseconds.

 **Delay** 


NodeID: 485313 

 Connector >  **Config**

* Delay Expression (ms)

1000

5.4 EDI Processor

The EDI processor exchanges and automatically processes data using a standard format between two computer systems.

Configuration Parameters

Parameter	Description
Operation	Select an operation for the component. <ul style="list-style-type: none"> • Parse: parses received EDI files of a specified standard into JSON files. • Generate: converts JSON files into EDI files of a specified standard.
Standard	Supported message standards are: X12 and EDIFACT
EDIFACT Version	This parameter is required for Standard set to EDIFACT . Select the EDIFACT version. EDIFACT is mainly used in Europe and Asia. The default version is 40100.
X12 Version	This parameter is required for Standard set to X12 . Select the X12 version. X12 is mainly used in North America. The default version is 005010.
Stop On Violation	Whether to stop the entire task upon EDI processing failure.
Input Type	Component deployment mode. <ul style="list-style-type: none"> • API Form-Data Param: submits target file information using an API form. • Text: submits target file information through a file transfer server, for example, an FTP server.
API Form-Data Param	Required when Input Type is set to API Form-Data Param . Enter the parameter name used for uploading the file in the API request body table to obtain the file information transferred by the previous node. You can use variables to reference the data in previous nodes. For details, see 3.6 Referencing Variables .
Content	Required when Input Type is set to Text . You can reference data in the previous node to obtain the file information. For details, see 3.6 Referencing Variables .

EDI Processor NodeID: 158426

Connector > Config

Operation
Parse

Standard
X12

X12 Version
005010

Stop On Violation ?
No

Input Type
API Form-Data Param

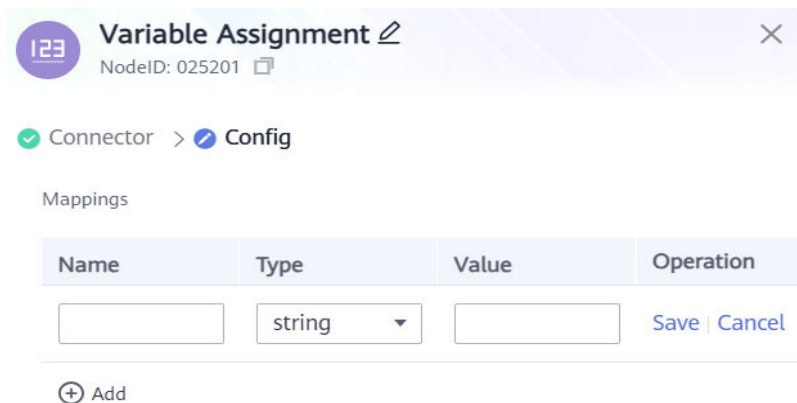
* API Form-Data Param ?

5.5 Variable Assignment

The variable assignment processor is used to create variables, which can be used in subsequent steps. Multiple variables can be created in a **Variable Assignment** node.

Configuration Parameters

Parameter	Description
Name	Enter the name of the variable.
Type	Type of the variable. Supported types are: string, integer, long integer, decimal, boolean, and object.
Value	Enter the value of the variable. The value can be a constant, for example, hello world . You can also use an expression to reference variables from previous nodes, for example, \${a} and \${payload} .

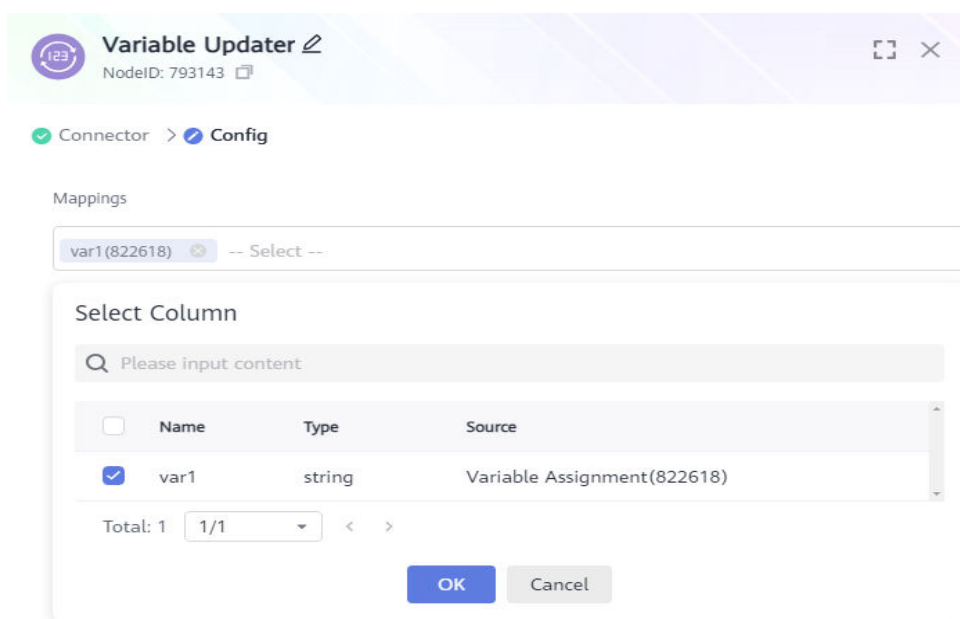


5.6 Variable Updater

The variable updater is used to modify the values of variables from a preceding **Variable Assignment** node. Multiple variables can be modified in a **Variable Updater** node.

Configuration Parameters

Parameter	Description
Variable	Select the variable created in a preceding value assignment processor.
Value	Enter the new variable value. The value can be a constant, for example, hello world . You can also use an expression to reference variables from previous nodes, for example, \${a} and \${payload} .



5.7 Sorter

The sorter processor is used to sort data.

Configuration Parameters

Parameter	Description
Dataset	Data to be sorted, which can be the data transferred from the previous node or custom data. You can reference data from previous nodes. For details, see 3.6 Referencing Variables .
Path	Parameter used for sorting. You can reference data from previous nodes. For details, see 3.6 Referencing Variables .
Sort By	Select a sorting mode. Options are Ascendingly and Descendingly .

The screenshot shows the configuration interface for the Sorter connector. At the top, there is a header with the connector name 'Sorter' and 'NodeID: 025201'. Below this, a breadcrumb indicates the current view is 'Config'. The configuration is organized into sections: 'Data' (required, marked with a red asterisk) is an empty text input field; 'Sort Path' (required, marked with a red asterisk and a help icon) contains the path '\$.data[*].id'; and 'Order' is a dropdown menu currently set to an empty state.

5.8 Data Conversion

The data conversion connector is used to convert the format of data streams. Currently, XML-to-JSON and JSON-to-XML are supported.

Configuration Parameters

Parameter	Description
Conversion Mode	Select the data conversion mode. Options are XML TO JSON and JSON TO XML .

Parameter	Description
Content	Use a script to compile the data to be converted. You can reference data from previous nodes. For details, see 3.6 Referencing Variables .



5.9 Splitter

The splitter processor splits the data transferred to this node based on the specified expression and transfers the split data to subsequent nodes through [node variables](#). Generally, a splitter component is followed by a [filter](#) component.

Splitter is an asynchronous processing component. If a task contains a **Splitter** node, the OpenAPI will return the data transferred to the **Splitter** node.

Configuration Parameters

Parameter	Description
Expression	Expression for splitting data. An expression is used to split data into data blocks for subsequent processes in parallel. For example, if the payload data is <code>{"data":[{"id":1},{"id":2}]}</code> , you can use <code>`\${payload.data}`</code> to split the data into <code>{"id":1}</code> and <code>{"id":2}</code> .



5.10 Filter

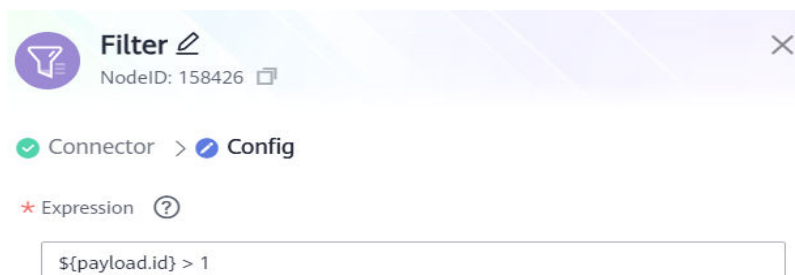
The filter component filters the data transferred to this component based on the specified expression and transfers the filtered data to subsequent nodes through **node variables**. Generally, a filter component is the subsequent node of a **splitter** component.

Configuration Parameters

Parameter	Description
Expression	Expression for filtering data. Use an expression to filter data and transfer the filtered data to subsequent nodes. For example, if multiple data blocks are <code>{"id":1}</code> and <code>{"id":2}</code> , you can use expression <code>\$ {payload.id} > 1</code> to filter out the data block <code>{"id":2}</code> .

NOTE

If no data matches the filtering expression, subsequent nodes of the **Filter** node will not be executed.

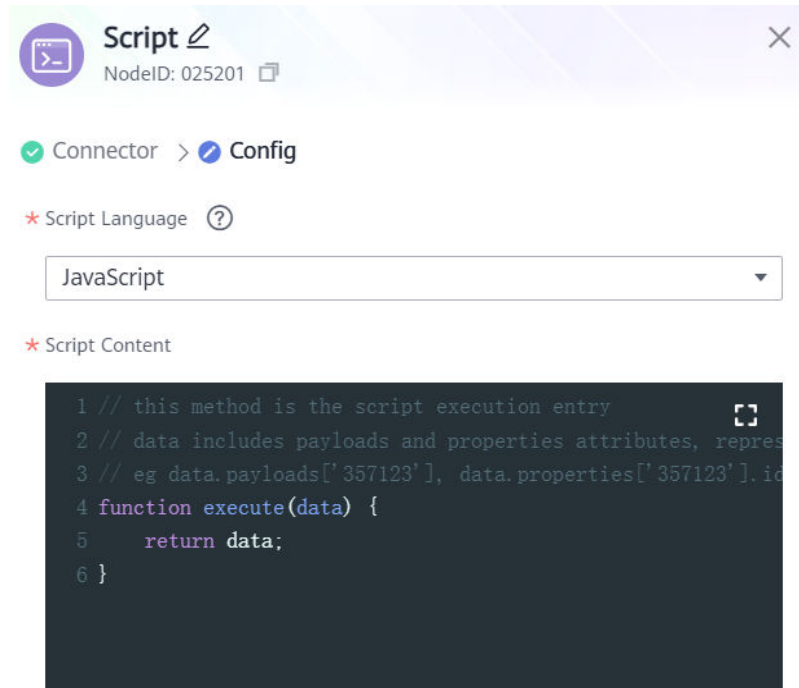


5.11 Script

The script processing component is used to process data. You can use a specified programming language to compile a script, process the data transferred to the script processing component, and generate new data for subsequent steps.

Configuration Parameters

Parameter	Description
Script Language	Programming language used for writing scripts. Currently, only JavaScript of the ES5 version is supported.
Content	<p>Compile the script content.</p> <p>You can reference data from previous nodes. For details, see 3.6 Referencing Variables.</p> <p>To parse content data in JSON format, use JSON.parse('\${body}') to convert the content data into a JSON object and then parse the data.</p> <p>The script processing component can generate an output result and transfer it to subsequent nodes. You need to specify the name of the variable to be output at the end of the script. For example:</p> <pre>function getUser(){ return {"id": \${property.id},"name": "\${property.name}"}; } var user=getUser(); user;</pre> <p>NOTE</p> <p>If you need to write a function or logic code block, use braces ({} to surround the code block.</p>

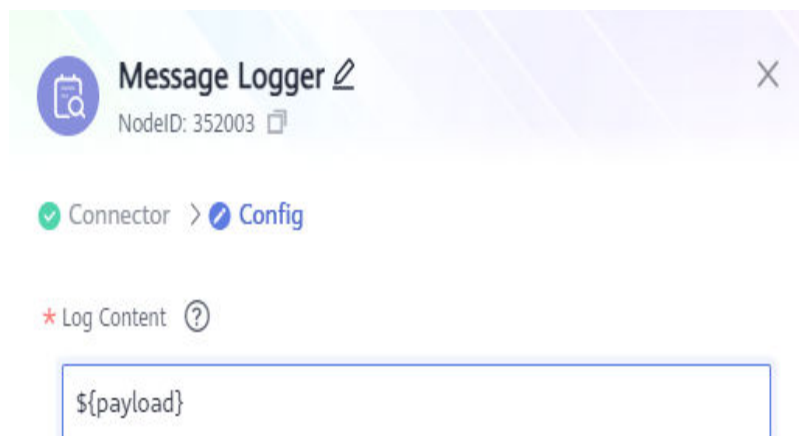


5.12 Message Logger

The **Message Logger** component is used to customize log information (log level: debug), which will be recorded in the runtime log of the current task.

Configuration Parameters

Parameter	Description
Log Content	Constants, or data in preceding steps by referencing variables. For details, see 3.6 Referencing Variables .

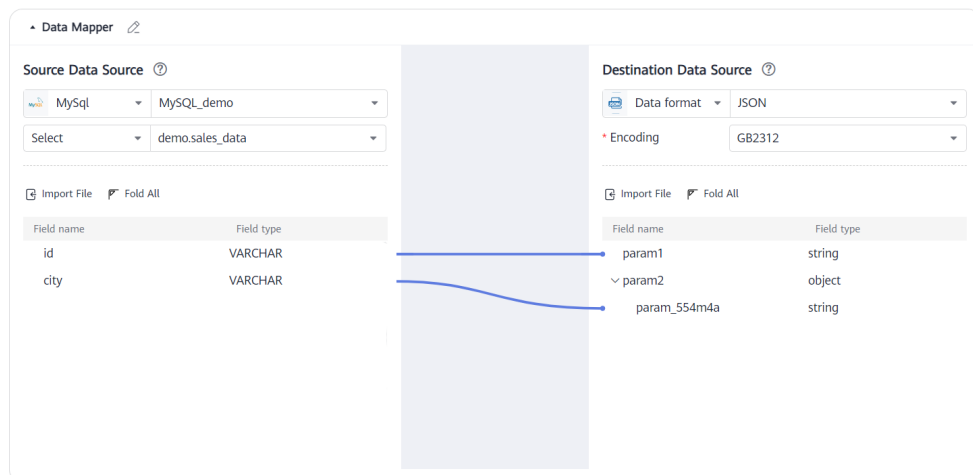


5.13 Data Mapper

The data mapper processor is used to convert the data format. Subsequent nodes can reference the output data from a data mapper using **`${payload}`**.

Configuring Parameters

Parameter	Description
Source Data Source	If the data source is a database, the field information is automatically displayed after a table is selected. In other cases, manually add fields or import a file. Only JSON, XML, and CSV files can be imported.
Destination Data Source	If the data source is a database, the field information is automatically displayed after a table is selected. In other cases, manually add fields or import a file. Only JSON, XML, and CSV files can be imported.

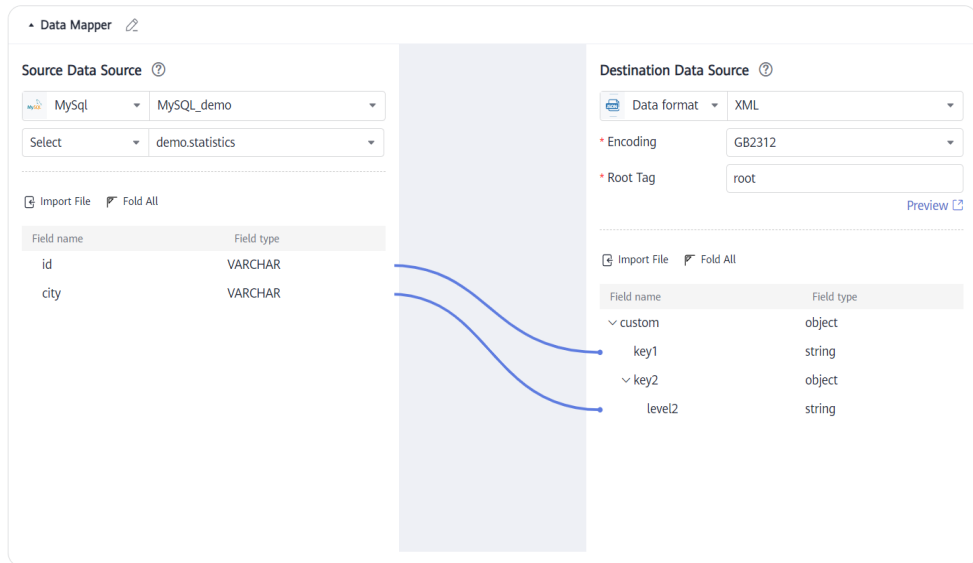


NOTE

Subsequent nodes can reference the output data of a data mapper processor using **`${node ID}{payload.xxx}`**. The node ID is displayed after you click a data mapper node.

Destination - XML Data Example

If destination data source format is XML, **Root Tag** defaults to **root**. You can set this parameter to other values but cannot leave it empty.



Data in the **demo.statistics** table of the source MySQL database:

id	city
uuid1	suzhou
uuid2	shanghai

Mapping result:

```
<root>
  <custom>
    <key1>uuid1</key1>
    <key2>
      <level2>suzhou</level2>
    </key2>
  </custom>
  <custom>
    <key1>uuid2</key1>
    <key2>
      <level2>shanghai</level2>
    </key2>
  </custom>
</root>
```

Destination - CSV Data Example

Header: Whether to include the table header of the CSV file in the output. Options are **true** and **false**.

Delimiter: Data in each file column is separated by cells, which can be represented as tab characters, spaces, commas, or semicolons.

Scenario 1: Set the destination data source format to CSV, **Header** to **true**, and **Delimiter** to **comma**.

When a file is imported, if **Header** is set to **true**, the first line of the CSV file is the field names.

	A	B
1	Product Name	ManufacturerId
2	'product-sub'	'product-sub'
3		
4		
5		

Data Mapper

Source Data Source
 MySQL | MySQL_demo
 Select | demo.sales_data

Field name	Field type
id	VARCHAR
city	VARCHAR

Destination Data Source
 Data format: CSV
 Encoding: GB2312
 Header: true
 Delimiter: comma

Field name	Field type
ProductName	string
ProductSerial	string

Data in the **demo.statistics** table of the source MySQL database:

id	city
uuid1	suzhou
uuid2	shanghai

Mapping result:

```
ManufacturerId, ProductName
uuid1, suzhou
uuid2, shanghai
```

Scenario 2: Set the destination data source format to CSV, **Header** to **false**, and **Delimiter** to **comma**.

When a file is imported, if **Header** is set to **false**, the header field of the imported file is not used. The **filedN** field is used by default.

	A	B
1	ProductName	ManufacturerId
2	'product-sub'	'product-sub'
3		
4		
5		

Data in the **demo.statistics** table of the source MySQL database:

id	city
uuid1	suzhou
uuid2	shanghai

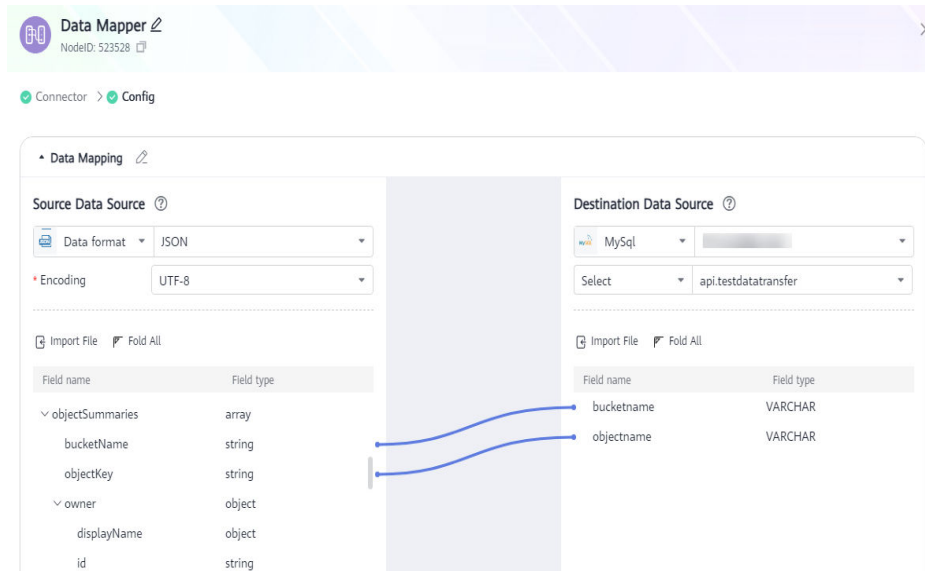
Mapping result:

```
uuid1, suzhou
uuid2, shanghai
```

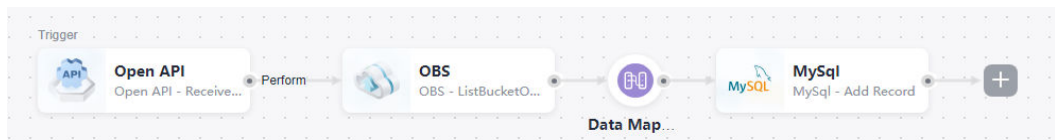
Source - JSON Data Example

If the source data format is JSON, use the output of the source data source as the **import file** of the source, and select the encoding format (**GB2312**, **UTF-8**, or **ASCII**) of the file. The system automatically generates fields accordingly.

Set the destination data format to MySQL, and select a table name. The system automatically generates fields in the database table.



(Example) The following composite application calls the open API to list objects in a specified OBS bucket and map the bucket name and object names to corresponding fields in a specified MySQL table via data mapping, and then stores the bucket name and object names to MySQL. In this application, the OBS data serves as the source of data mapping, while MySQL is the destination. The open API calls the OBS connector to list bucket objects as the import file of the remote data source. Alternatively, you can also manually add fields based on the object list.



Content of the imported file:

```
{
  "responseHeaders": {
    "bucket-location": "cn-north-7",
    "connection": "keep-alive",
    "content-length": "504",
    "content-type": "application/xml",
    "date": "2024-04-24T07:07:43.000+00:00",
    "id-2": "32AAAQAAEAABAAAQAAEAABAAAQAAEAABCSbOSbK
    nLEgZnxffa8vFPuDgVIt7n9",
    "request-id": "0000018F0EEE84734012ACF15A46395E",
    "server": "OBS"
  },
}
```

```
"originalHeaders": {
  "connection": "keep-alive",
  "content-length": "504",
  "content-type": "application/xml",
  "date": "Wed, 24 Apr 2024 07:07:43 GMT",
  "server": "OBS",
  "x-obs-bucket-location": "cn-north-7",
  "x-obs-id-2": "32AAAQAEEAABAAAQAEEAABAAAQAEEAABCSbOSbK
+ncLEgZnxffa8vFPuDgVlt7n9",
  "x-obs-request-id": "0000018F0EEE84734012ACF15A46395E"
},
"statusCode": 200,
"objectSummaries": [
  {
    "bucketName": "test0223",
    "objectKey": "test0424.txt",
    "owner": {
      "displayName": null,
      "id": "0ab78f256080d2890feac0086234d7c0"
    },
    "metadata": {
      "responseHeaders": {},
      "originalHeaders": {},
      "statusCode": 0,
      "lastModified": "2024-04-24T06:51:46.566+00:00",
      "contentLength": 19,
      "contentType": null,
      "contentEncoding": null,
      "contentDisposition": null,
      "cacheControl": null,
      "contentLanguage": null,
      "expires": null,
      "etag": "\"0b26e313ed4a7ca6904b0e9369e5b957\"",

```

```
"contentMd5": null,
"crc64": null,
"storageClass": "STANDARD",
"webSiteRedirectLocation": null,
"nextPosition": -1,
"appendable": false,
"objectStorageClass": "STANDARD",
"allMetadata": {},
"metadata": {},
"requestId": ""
},
"objectContent": null
}
],
"commonPrefixes": [],
"extendCommonPrefixes": [],
"bucketName": "test0223",
"truncated": false,
"prefix": "",
"marker": "",
"maxKeys": 1000,
"delimiter": null,
"nextMarker": null,
"location": "cn-north-7",
"extenedCommonPrefixes": [],
"objects": [
{
"bucketName": "test0223",
"objectKey": "test0424.txt",
"owner": {
"displayName": null,
"id": "0ab78f256080d2890feac0086234d7c0"
},

```

```
"metadata": {  
  "responseHeaders": {},  
  "originalHeaders": {},  
  "statusCode": 0,  
  "lastModified": "2024-04-24T06:51:46.566+00:00",  
  "contentLength": 19,  
  "contentType": null,  
  "contentEncoding": null,  
  "contentDisposition": null,  
  "cacheControl": null,  
  "contentLanguage": null,  
  "expires": null,  
  "etag": "\"0b26e313ed4a7ca6904b0e9369e5b957\"",  
  "contentMd5": null,  
  "crc64": null,  
  "storageClass": "STANDARD",  
  "webSiteRedirectLocation": null,  
  "nextPosition": -1,  
  "appendable": false,  
  "objectStorageClass": "STANDARD",  
  "allMetadata": {},  
  "metadata": {},  
  "requestId": ""  
},  
"objectContent": null  
}  
],  
"requestId": "0000018F0EEE84734012ACF15A46395E"  
}
```

Data mapping result:

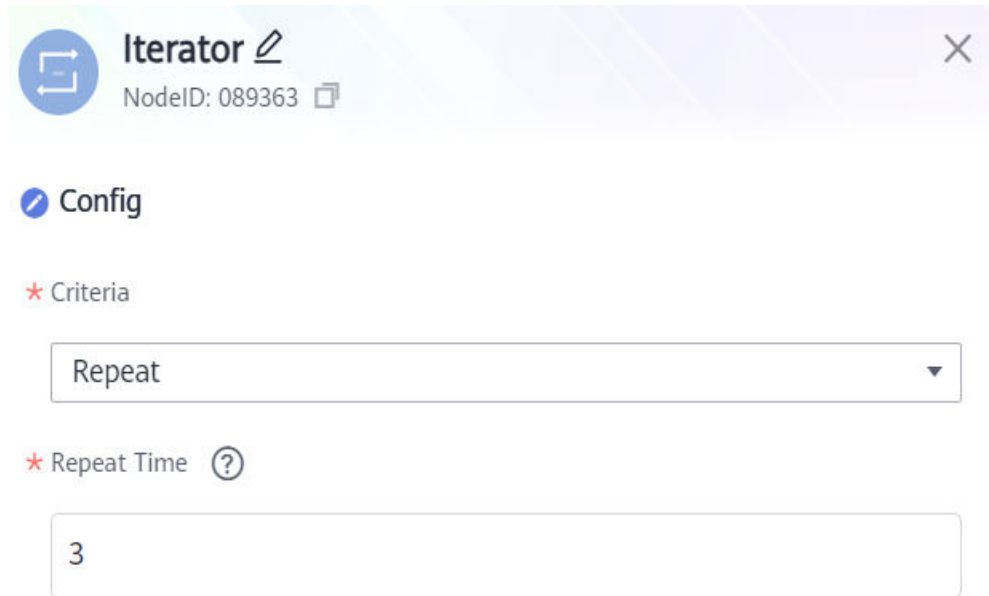
bucketname	objectname
test0223	test0424.txt

5.14 Iterator

The iterator processor is used to process data cyclically. You can specify the number of iterations, conditions, or arrays.

Configuration Parameters

Parameter	Description
Criteria	Specify the iteration criteria. <ul style="list-style-type: none"> • Repeat • Do While • Foreach
Count	When Criteria is set to Repeat , configure the number of iterations. The default value is 3 . A maximum of 100 iterations can be executed. Any access will trigger an error report.
Condition	When Criteria is set to Do While , configure the iteration conditions. When the conditions are met, a maximum of 100 iterations can be executed. Any access will trigger an error report.
Array	When Criteria is set to Foreach , configure the arrays to be iterated, which can be obtained from the preceding node. For details, see 3.6 Referencing Variables . A maximum of 100 elements can be included in an array. Any access will trigger an error report.
Parallel Process	Required when Criteria is set to Foreach . A maximum of 10 parallel processes are supported.

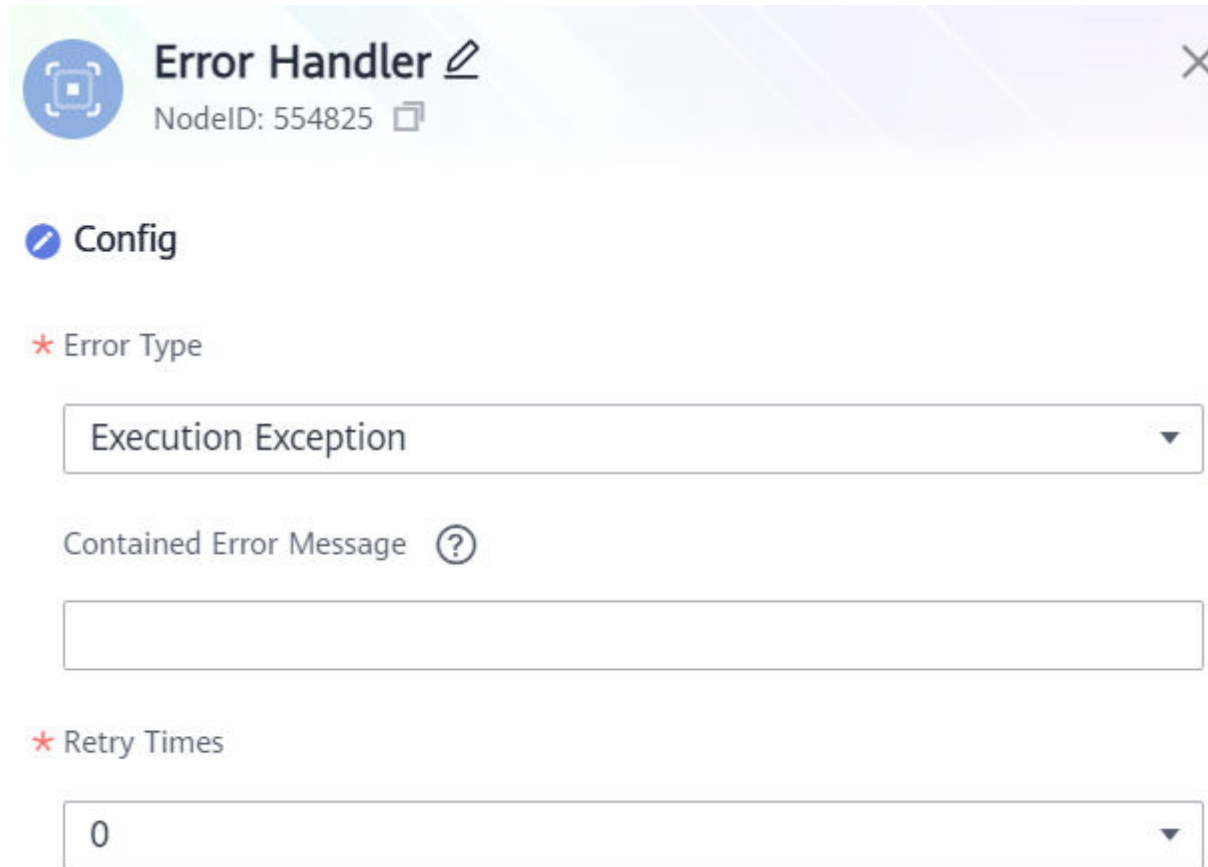


5.15 Error Monitoring

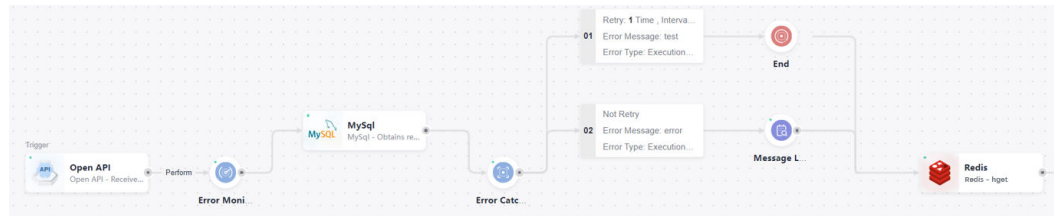
The error monitoring processor is used to monitor a specified process, capture exceptions, and perform retry and exception handling operations based on configured parameters.

Configuration Parameters

Parameter	Description
Error Type	Type of the captured exception. Execution Exception covers all service exceptions.
Error Message	Error message contained in the captured exceptions. The exception handling procedure is performed only when the captured exception contains specified error message.
Retry Times	Maximum number of retry times of monitored processes when an exception occurs.
Retry Delay (Second)	Retry interval of monitored processes when an exception occurs, in seconds.



Configuration Example



In the preceding figure, error monitoring is configured to monitor the MySQL component to obtain records. If an exception occurs and the error message contains **test**, the MySQL operation will be retried in two seconds and then terminated. Subsequent processes will not be performed. If an exception occurs and the error message contains **error**, the MySQL operation will not be retried, and the task proceeds to log collection and the Redis **hget** operation.

5.16 End

The end processor is used to terminate a flow task of a composite application.

This processor is usually placed at the end of a flow.

You do not need to configure parameters for this processor.

5.17 Symmetric Encrypt/Decrypt


The symmetric encrypt/decrypt processor is used to encrypt plaintext and decrypt ciphertext by using the specified symmetric encryption algorithm.

Configuration Parameters

Parameter	Description
Operation	Options are symmetric encrypt, symmetric decrypt, HMAC signing, and HMAC signature verification.
Encryption Algorithm	Symmetric encrypt or decrypt supports only the AES algorithm. HMAC signing or signature verification supports only HMAC algorithm.
Encryption Mode	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Currently, only GCM is supported.
PBKDF2 Password	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Used to generate the PBKDF2 key.
PBKDF2 Hash Algorithm	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . SHA256 and SHA384 are supported.
PBKDF2 Salt	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Used to generate the PBKDF2 key.
Secret Key Length	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Length of the generated key. Options are 128 , 192 , or 256 .
Plaintext	Plaintext to be encrypted, or signed, or verified.
Ciphertext	Ciphertext to be decrypted.
Content Padding Mode	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Currently, only PKCS5Padding is supported.
PBKDF2 Iteration Count	Mandatory when Operation is set to Symmetric Encrypt or Symmetric Decrypt . Used to generate the PBKDF2 secret key. The default value is 100000 .

Parameter	Description
TAG Length	Mandatory when Operation is set to Symmetric Encrypt . Used to generate a random tag. The default value is 16 .
IV Length	Mandatory when Operation is set to Symmetric Encrypt . Used to generate a random IV. The default value is 12 .
AAD Length	Mandatory when Operation is set to Symmetric Encrypt . Used to generate a random AAD. The default value is 16 .
Secret Key	Mandatory when Operation is set to HMAC Signing or HMAC Signature Verification . HMAC signature key.
Signature	Mandatory only when Operation is set to HMAC Signature Verification . HMAC signing result.
Signature Algorithm	Mandatory when Operation is set to HMAC Signing or HMAC Signature Verification . HMAC signing algorithm. Options are SHA256 and SHA384.

Symmetric En/Decryption


NodeID: 585875 


✓ Connector > ⚙️ Config


* Operation


* Encryption Algorithm

* Encryption Mode

* PBKDF2 Password 

* PBKDF2 Hash Algorithm 

* PBKDF2 Salt 

* Secret Key Length 

* Plaintext


5.18 Asymmetric Encrypt/Decrypt

The asymmetric encrypt/decrypt processor is used to encrypt plaintext and decrypt ciphertext by using the specified asymmetric encryption algorithm.


Configuration Parameters


Parameter	Description
Private Key	Base64-encoded, mandatory for decryption.
Public Key	Base64-encoded, mandatory for encryption.
Operation	Asymmetric encryption or decryption.
Encryption Algorithm	Currently, only RSA is supported.
Plaintext	Mandatory when Operation is set to Asymmetric Encrypt . Plaintext to be encrypted.
Ciphertext	Mandatory when Operation is set to Asymmetric Decrypt . Base64-encoded ciphertext to be decrypted.
Padding Mode	Plaintext padding mode. Currently, only RSA_PKCS1_OAEP_PADDING is supported.
OAEP Hash Algorithm	Mandatory when Padding Mode is set to RSA_PKCS1_OAEP_PADDING . Options are SHA256 and SHA384.

Asymmetric En/Decryption

NodeID: 710841 


Connector > **Config**

Private Key 

Public Key 

* Operation

* Encryption Algorithm

* Ciphertext 

* Padding Mode

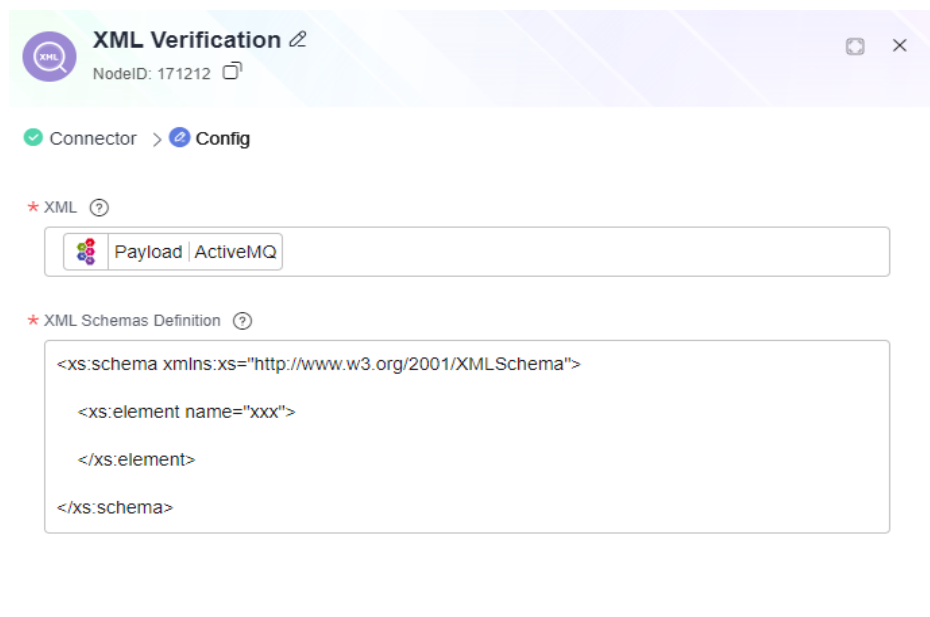
* OAEP Hash Algorithm

5.19 XML Verification

The XML verification processor is used to verify and process XML content. If the verification is successful, no error logs will be generated. Otherwise, error logs will be recorded and the flow task will be stopped.

Configuring Parameters

Parameter	Description
XML	XML data to be verified. The value can contain a maximum of 100,000 characters.
XML Schemas Definition	XSD data to be verified (same as the XML compilation rule). The namespace http://www.w3.org/2001/XMLSchema cannot be changed.



6 Debugging a Task Flow Online

After creating and configuring a composite application and its task flows, you can debug the flows online and view the running information of each node in real time to meet service requirements.

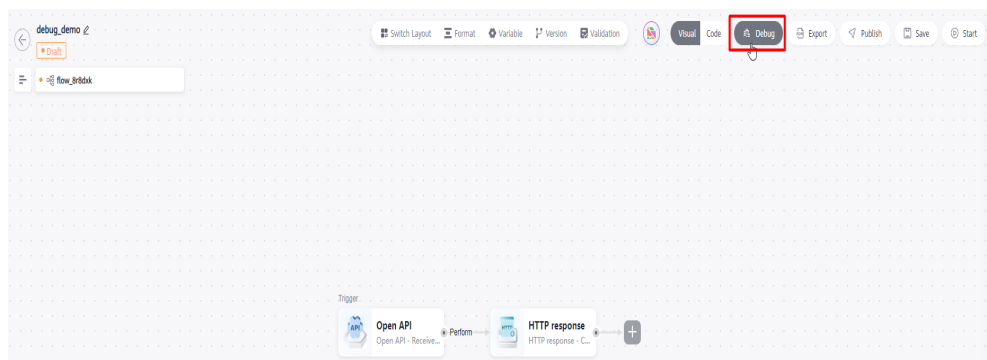
Prerequisites and Constraints

- A ROMA Connect instance is required to serve as the debugging environment. You can purchase an exclusive ROMA Connect instance or use a shared instance.
- Online flow debugging may affect other flows running on the instance. You are advised to use an exclusive instance for debugging.
- A shared instance allows only one flow to be debugged at a time. An exclusive instance allows a maximum of five flows to be debugged at a time.
- The trigger node does not run during online debugging. For example, the OpenAPI node does not publish any API during debugging. Manually configure the trigger node to trigger the flow.
- The debugging process pauses at a node for a maximum of 60 seconds, and then proceeds to the next one.

Procedure

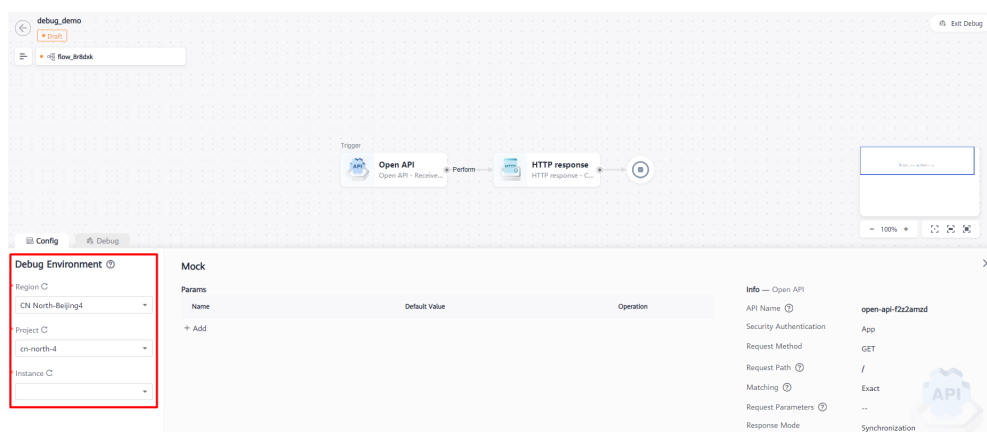
1. Log in to the new ROMA Connect console.
2. In the navigation pane on the left, choose **Application**. On the page displayed, click the target application.
3. Click **Modify** in the upper right to go to the design page.
4. Click **Debug** at the top to enter the debugging mode.

Figure 6-1 Canvas for application designing



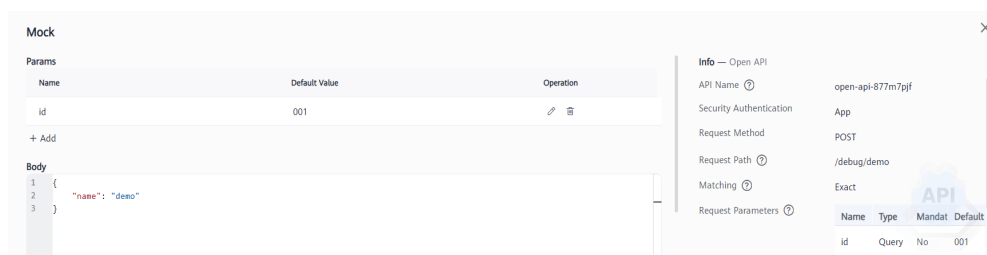
5. Select a debug environment. Set **Region**, **Project**, and **Instance** for the online debugging environment.

Figure 6-2 Configuring debugging environment



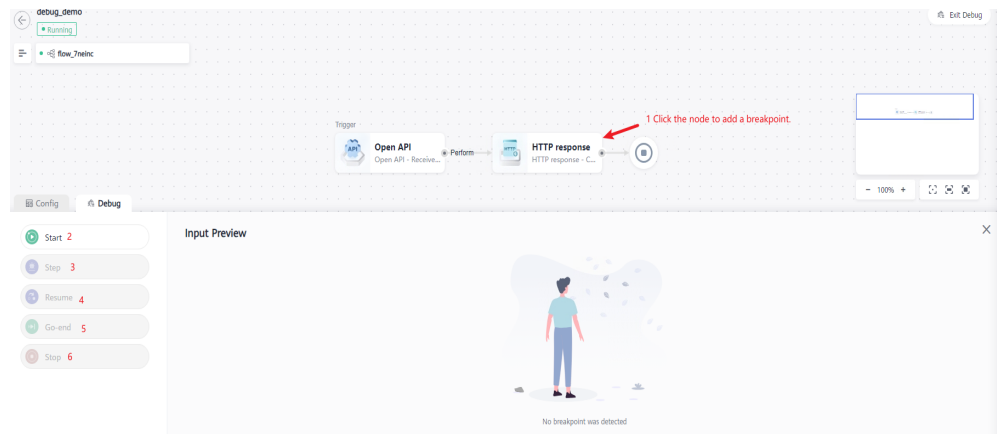
6. Configure the trigger node. Configure trigger parameters based on the trigger type. For example, configure API request parameters and request body for an OpenAPI node.

Figure 6-3 Configuring OpenAPI node parameters



7. Click **Save**.
8. Debug the flow.

Figure 6-4 Debugging a flow



- a. Click a node of the flow to add a breakpoint. The flow may stop at a breakpoint during debugging. To remove the breakpoint, click the node again.

 NOTE

Breakpoints cannot be added to trigger, parallel processor, or conditioning nodes.

- b. **Start:** Click **Start** to start debugging a flow. If a breakpoint exists in the flow, the task stops at the breakpoint and the **properties** and **payload** data that enters the breakpoint is displayed. In other cases, the task automatically runs through the flow, or stops at the error node with the error information.
 - c. **Step:** When the debugging process pauses at a node, click **Step** to proceed to the next node.
 - d. **Resume:** When the debugging process pauses at a node, click **Resume** to proceed to the next breakpoint. If no such breakpoint exists, the task automatically runs through the flow, or stops at the error node with the error information.
 - e. **Go-end:** When the debugging process pauses at a node, click **Go-end** to run through the flow, or the task stops at the error node with the error information.
 - f. **Stop:** When a flow is in the debugging state, click **Stop** to terminate the debugging.
9. After the debugging is complete, click **Exit Debug** in the upper right to exit the debugging mode and return to the application design page.

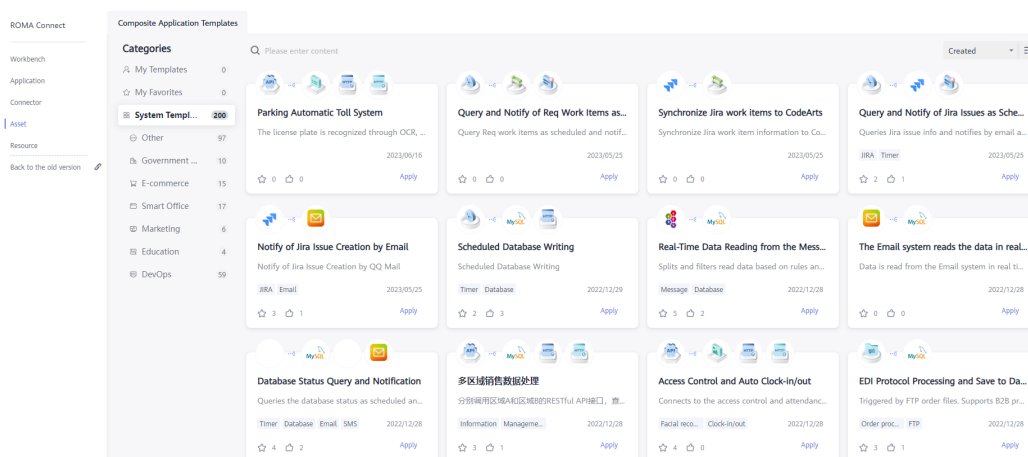
7 Asset Catalog

All composite application templates, including custom, favorite, and system templates are stored in **Asset Catalog**.

You can add a system template to your favorites and click a template to check its details.

You can also click **Apply** on a card to create a composite application using the template.

Figure 7-1 Asset catalog



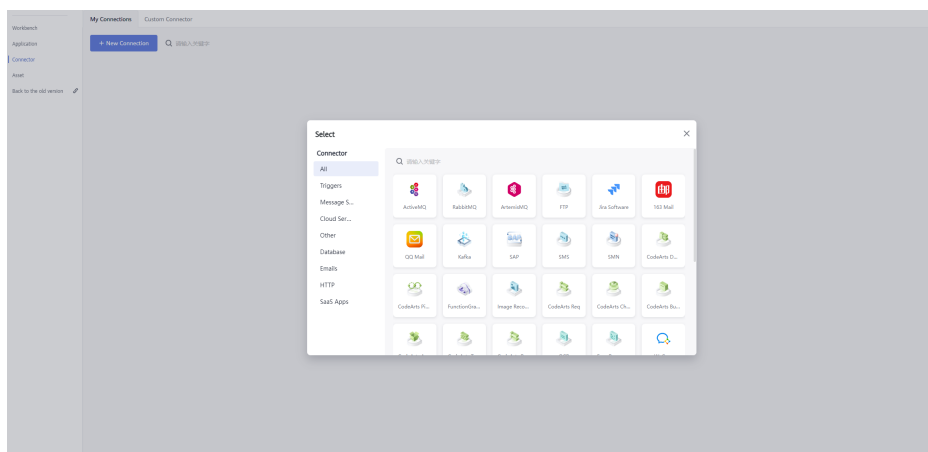
8 Related Concepts

Composite Application

Composite applications are built by integrating multiple existing applications via open APIs and event channels for fast service replication and innovation with tailored integration experience.

Connector

An iPaaS connector is a cloud service that can be used to connect applications, data, and devices to implement various integration solutions. Connectors provide pre-built connection points to transfer data from one application or system to another.



9 Appendix

9.1 Accessing the Internet from an Instance

9.1 Accessing the Internet from an Instance

RCU instances access the Internet via the tenant VPC, and tenants need to enable the NAT gateway in VPC for public access.

Procedure

1. View the VPC and subnet of the RCU instance by referring to [Viewing Details of an Instance](#)
2. [Buy a public NAT gateway](#). Set the VPC and subnet to the ones where the instance resides, and configure other parameters as required.

Figure 9-1 Buying a public NAT gateway

< | Buy Public NAT Gateway ⓘ

* Billing Mode: **Yearly/Monthly** | Pay-per-use

* Region: [Dropdown]

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

* Name: [Input: nat-]

* VPC: [Dropdown: --Select--] | View VPCs

Only VPCs without NAT gateways and default routes can be selected.

* Subnet: [Dropdown: --Select--] | View Subnets

The selected subnet is for the NAT gateway only. To enable communications over the Internet, after the NAT gateway is created, you need to add rules.

* Specifications: **Small** | Medium | Large | Extra-large

Supports up to 10,000 connections. [Learn more](#)
The connections supported by a NAT gateway in a yearly/monthly subscription can always be increased later, but they cannot be decreased.

Advanced Settings ▾

Description	Tag
1	
2	
3	
4	
5	
6	
7	
8	
9 months	
1 year	
2 years	
3 years	

Auto-renew ⓘ

3. Set the NAT gateway rule by referring to [Adding an SNAT Rule](#). Set **Scenario** to **VPC**, and **CIDR Block** to the subnet where the RCU instance resides, and bind a public IP address.

Figure 9-2 Adding an SNAT rule

Add SNAT Rule

- If both an EIP and a NAT gateway are configured for a server, data will be forwarded through the EIP. [View restrictions](#)
- It is not recommended that an SNAT rule and a DNAT rule share the same EIP because there may be service conflicts.
- An SNAT rule cannot share an EIP with a DNAT rule with Port Type set to All ports.

Public NAT Gateway Name: nat-

* Scenario: **VPC** Direct Connect/Cloud Connect

* CIDR Block: **Existing** Custom ⓘ

--Select--

* Public IP Address Type: **EIP**

You can select 20 more EIPs. ⓘ [View EIP](#) ⓘ

<input type="checkbox"/>	EIP	EIP Type	Bandwidth Name	Bandwidth(Mbit/s)	Billing Mode
<input type="checkbox"/>	10	5_g-vm	bandwidth	10	Yearly/Monthly

If multiple EIPs are selected for an SNAT rule, an EIP will be chosen from your selection at random.

Monitoring: Create alarm rules in [Cloud Eye](#) to monitor your SNAT connections.

Description:

Cancel **OK**