

ROMA Connect

Getting Started

Issue 01
Date 2024-04-23



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 Using ROMA Connect for Data Integration.....	1
2 Using ROMA Connect for Service Integration.....	8
3 Using ROMA Connect for Message Integration.....	15
4 Using ROMA Connect for Device Integration.....	22
5 Using ROMA Connect for Composite Applications.....	35
6 Getting Started with Common Practices.....	40

1 Using ROMA Connect for Data Integration

Overview

ROMA Connect supports access of multiple types of data sources and uses data integration tasks to integrate data from source to destination. Both homogeneous and heterogeneous data can be integrated.

This chapter provides an example for configuring data integration from API to MQS to help you get familiar with the data integration process.

The steps for integrating data with ROMA Connect are as follows:

- [Step 1: Prepare Data Sources](#)
- [Step 2: Connect to Data Sources](#)
- [Step 3: Create a Data Integration Task](#)
- [Step 4: Start the Data Integration Task](#)
- [Step 5: View the Data Integration Result](#)

Step 1: Prepare Data Sources

Before using ROMA Connect to integrate data, you need to prepare one API data source and one MQS data source as the source and destination, respectively.

1. Create and publish a function API.

Create, debug, and publish an API in the current ROMA Connect instance as the source. For details, see [Create an API](#), [Debug the API](#), and [Publish the API](#).

For simplicity, the function API does not require request authentication or carry request parameters. When the function API is called, the message "ROMA TEST!" is always returned.

After the function API is created, record the request URL and method of the API.

2. Create a message topic.

Create a message topic on the current ROMA Connect instance as the topic of the MQS data source at the destination. For details, see [Creating a Message Topic](#).

After the topic is created, record the topic name. On the **Application Integrations** page of the ROMA Connect console, click the name of the integration application to which the topic belongs to view and obtain the key and secret.

Step 2: Connect to Data Sources

Before creating a data integration task, separately connect the API and MQS data sources to ROMA Connect to ensure that data can be read from the source and written to the destination.

1. Create an integration application.
 - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** of an instance.
 - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
 - c. In the dialog box displayed, set **Name** and click **OK**.
2. Connect to the API data source.
 - a. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
 - b. On the **Default** tab page, select **API** and click **Next**.
 - c. Enter the configuration information about the API data source.

Table 1-1 API data source configuration

Parameter	Description
Name	Enter an API data source name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in 1 .
Description	Enter a brief description of the API data source.
Address	Enter the API request URL recorded in Step 1: Prepare Data Sources .
Request Method	Select the API request method recorded in Step 1: Prepare Data Sources .
Authentication Mode	Select None , indicating that no authentication is required for the API data source.

- d. After setting the parameters for the API data source, click **Check Connectivity** to test the data source connection.
 - If the test result is **Data source connected successfully**, click **Create**.
 - If the test result is **Failed to connect to the data source**, check and modify the connection parameters, and click **Recheck** until the connection is successful.

- e. Click **Create**.
3. Connect to the MQS data source.
 - a. On the **Data Sources** page, click **Access Data Source** in the upper right corner.
 - b. On the **Default** tab page, select **MQS** and click **Next**.
 - c. Enter the configuration information about the MQS data source.

Table 1-2 MQS data source configuration

Parameter	Description
Name	Enter an MQS data source name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in 1 .
Description	Enter a brief description of the MQS data source.
Connection Address	Select the MQS intranet address of the current ROMA Connect instance. You can view the MQS intranet address on the Instance Information page of the ROMA Connect console.
Enable SSL	Set this parameter to Yes if MQS SASL_SSL is enabled and Intra-VPC Plaintext Access is disabled for the ROMA Connect instance. In other cases, set this parameter to No .
SSL Username/ Application Key	This parameter is mandatory only if Enable SSL is set to Yes . Set this parameter to the key recorded in Step 1: Prepare Data Sources .
SSL Password/ Application Secret	This parameter is mandatory only if Enable SSL is set to Yes . Set this parameter to the secret recorded in Step 1: Prepare Data Sources .

- d. After setting the parameters for the MQS data source, click **Check Connectivity** to test the data source connection.
 - If the test result is **Data source connected successfully**, click **Create**.
 - If the test result is **Failed to connect to the data source**, check and modify the connection parameters, and click **Recheck** until the connection is successful.
- e. Click **Create**.

Step 3: Create a Data Integration Task

After configuring the API and MQS data sources, create a data integration task from the source to the destination.

1. In the navigation pane on the left, choose **Fast Data Integration > Task Management** and click **Create Common Task**.
2. On the **Create Task** page, set the parameters of the data integration task.
 - a. Configure basic information about the task.

Table 1-3 Basic task information

Parameter	Description
Task Name	Enter a task name. It is recommended that you enter a name based on naming rules to facilitate search.
Description	Enter a brief description of the task.
Integration Mode	Select Scheduled as the mode used for data integration. When API is used as the data source at the source, only scheduled tasks are supported.
Tag	Add a tag to classify tasks for quick search. Skip this parameter.
Enterprise Project	Select the enterprise project to which the task belongs. In this example, retain the default value default .

- b. Configure a schedule for the task.
For simplicity, retain the default settings for the task schedule.
 - c. Configure the API data source at the source.

Figure 1-1 Configuring source information

The screenshot displays the 'Source Information' configuration interface. It includes the following elements:

- Instance:** A dropdown menu with 'roma-pnla' selected.
- Integration Application:** A dropdown menu with 'app-test' selected.
- Data Source Type:** A dropdown menu with 'API' selected.
- Data Source Name:** A dropdown menu with 'Select a data source.' and a 'Create Data Source' link.
- Paging:** A toggle switch that is turned off, with a note: 'If Paging is disabled, ROMA Connect obtains all data that meets the conditions through one API request.'
- Incremental Migration:** A toggle switch that is turned off.
- Request Parameters:** A section with tabs for 'Params', 'Headers', and 'Body'. Below the tabs is a table with 'Key' and 'Value' columns and an 'Add' button.
- Parse:** A dropdown menu with 'No' selected.

Table 1-4 Configuration information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application created in Step 2: Connect to Data Sources .
Source Data Type	Select API .
Data Source Name	Select the API data source that you configured in Step 2: Connect to Data Sources .
Paging	Skip this parameter.
Incremental Migration	Skip this parameter.
Request Parameters	The API data source does not have request parameters. Therefore, you do not need to set this parameter.
Parse	Set this parameter to No .

- d. Configure the MQS data source at the destination.

Figure 1-2 Configuring destination information

The screenshot shows a configuration form titled "Destination Information". It contains several dropdown menus and a text input field, each preceded by a red asterisk indicating a required field. The fields are: Instance (romafdi_roma-lz5a), Integration Application (app-doc), Data Source Type (MQS), Data Source Name (MQS-doc), Topic Name (Topic-doc), and Key (Enter a key.). A blue "Create Data Source" button is located to the right of the Data Source Name dropdown.

Table 1-5 Configuration information at the destination

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application created in Step 2: Connect to Data Sources .
Data Source Type	Select MQS .
Data Source Name	Select the MQS data source that you configured in Step 2: Connect to Data Sources .
Topic Name	Select the topic recorded in Step 1: Prepare Data Sources .
Key	Skip this parameter.

- e. Retain the default settings for **Post-Integration Operation**.
3. Click **Save**, and then click **Back to Task List**. The created data integration task is displayed in the task list.

Step 4: Start the Data Integration Task

After a data integration task is created, **Task Status** will be displayed as **Stopped**.

You can start the task by using either of the following methods:

- **Creating schedule:** The task is periodically executed according to the schedule created during task creation.
Select the task you want to start, click **Start** above the task list to start the task schedule, and wait until the task is executed as scheduled.

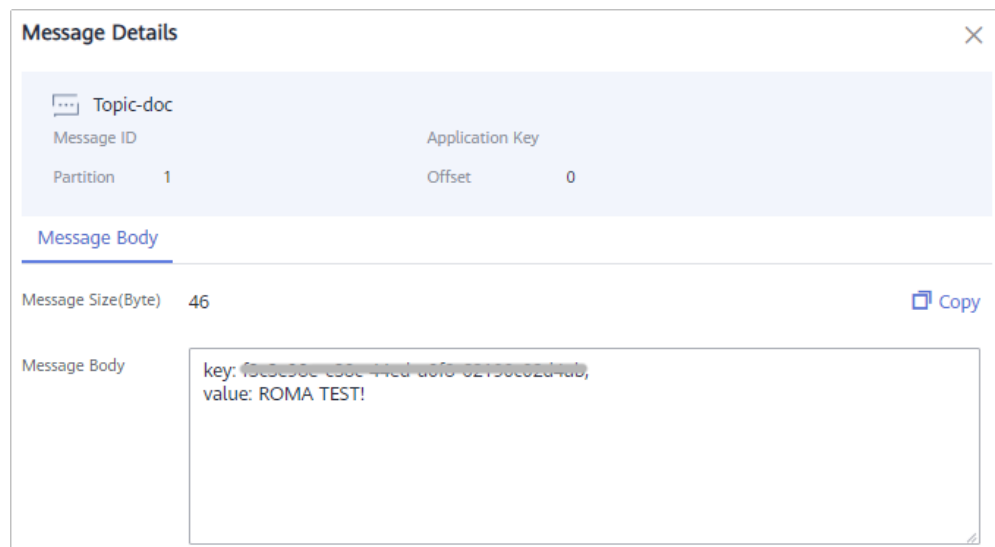
- Manual scheduling:** The task is manually executed only once. Before manually executing a task, ensure that **Task Status** is **Stopped**.
 Click **Start Scheduling** on the right of the data integration task you want to start. When **Running Status** of the task is displayed as **Successful**, the task is successfully executed.

To execute the API-to-MQS data integration task immediately and view the task result, manual scheduling is selected here.

Step 5: View the Data Integration Result

- In the navigation pane on the left, choose **Message Queue Service > Message Query**.
- Click **Advanced Search** to expand advanced search options.
- Enter the search criteria and click **Search** to query the messages sent by the client.
 - Select the message topic created in [Step 1: Prepare Data Sources](#) for **Topic Name**.
 - Select **Creation time** for **Search By** and select the time segment for the client to send messages to ROMA Connect.
- Click **Message Body** next to the message record and check whether the message content is the same as the response message of the API.

Figure 1-3 Viewing the message content



2 Using ROMA Connect for Service Integration

Overview

ROMA Connect encapsulates APIs, data sources, and custom functions into RESTful APIs and opens them to external systems, manages the whole API lifecycle, and protects APIs in multiple layers.

This chapter provides a configuration example to help you get familiar with the service integration process.

The steps for integrating services with ROMA Connect are as follows:

- [Step 1: Create an API Group](#)
- [Step 2: Create an API](#)
- [Step 3: Debug the API](#)
- [Step 4: Publish the API](#)
- [Step 5: Call the API](#)

Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance, and the network where the local PC is located can communicate with the EIP network.
- The [Postman](#) tool has been downloaded and installed on your local PC.

Step 1: Create an API Group

Each open API belongs to an API group. Before creating an API, create an API group.

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** of an instance.
2. Create an integration application.
 - a. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.

- b. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
3. In the navigation pane on the left, choose **API Connect > API Groups**. In the upper right corner of the page, click **Create API Group** and select **Create Directly**.
4. In the **Create API Group** dialog box, enter the group-related parameters and click **OK**.

Figure 2-1 API group configuration

The screenshot shows a configuration dialog for an API group. It includes the following fields and options:

- Name:** A text input field containing 'APIGroup_test'. Below it, a note states: 'Enter a string of 3 to 255 characters starting with a letter. Only letters, digits, hyphens (-), underscores (_), periods (.), slash (/), colons (:), and parentheses (()) are allowed.'
- Scope:** Two radio buttons: 'Current application' (selected) and 'All'.
- Integration Application:** A dropdown menu showing 'app-test'. To its right is a link that says 'Create Integration Application'.
- Description:** A large text area with the placeholder text 'Enter a description.' and a character count '0/255' in the bottom right corner.

Table 2-1 Parameters for creating an API group

Parameter	Description
Name	Enter an API group name. Using naming rules facilitates future search.
Scope	Specify whether the API group is available on the current application or on all applications. Select Current application .
Integration Application	Select the integration application created in 2. If none is available, click Create Integration Application on the right to create one.
Description	Enter a brief description of the API group.

Step 2: Create an API

Create an API on ROMA Connect and associate it with the backend service.

1. In the left navigation pane on the instance console, choose **API Connect > APIs**. In the upper right corner, click **Create API**.
2. On the page displayed, configure the frontend definition of the API.

Figure 2-2 Frontend configuration

The screenshot shows a 'Frontend Definition' form with the following fields and values:

- API Name:** API_test
- Integration Application:** app-test
- Group:** Group-test
- URL:**
 - Method: GET
 - Protocol: HTTPS
 - Subdomain Name: 774d8...9c87.apic...
 - Path: /test
- Matching:** Exact match (selected), Prefix match
- Tags:** Select or enter tags.
- Description:** Enter a description.

Table 2-2 Parameters for defining an API frontend

Parameter	Description
API Name	Enter an API name. Using naming rules facilitates future search.
Integration Application	Select the integration application created in Step 1: Create an API Group . If none is available, click Create Integration Application on the right to create one.
Group	Select the API group created in Step 1: Create an API Group .
URL	Configure the API access address. <ul style="list-style-type: none"> • Method: request method of the API. Set this parameter to GET here. • Protocol: request protocol of the API. Set this parameter to HTTPS here. • Path: request path of the API, in /getUserInfo/{userId} format. Set this parameter to /test here.
Matching	Matching mode of the API request path. Set this parameter to Exact match here.
Tags	Skip this parameter.
Description	Skip this parameter.

3. Configure API security information.

Figure 2-3 Security configuration

The screenshot shows the 'Security Configuration' panel. At the top, 'Visibility' is set to 'Public' (with a help icon). Below it, 'Authentication Mode' is set to 'None' (with options for App, IAM, Custom, and None). A 'Security Level' slider is positioned at the far left, with the text 'All users will be granted access. (Not recommended)'. At the bottom, there is a checkbox for 'CORS' with the description: 'Enable this option to allow restricted resources on a web page to be requested from other domains.'

Table 2-3 Security parameters

Parameter	Description
Visibility	Specifies whether the API can be released to the marketplace. Retain the default value Public , which indicates that the API can be released to the marketplace.
Authentication Mode	Security authentication mode used by the API. In this example, select None .
CORS	Specifies whether CORS is supported. You do not need to enable this option.

4. Configure the request parameters of the API. You can skip this step and click **Next**.
5. Set the backend type. In this example, select **Mock** to return specified response.
6. Configure API backend information.

Figure 2-4 Backend configuration

The screenshot shows the 'Basic Information' section of the backend configuration. 'Status Code' is a dropdown menu set to '200'. 'Response' is a text area containing 'ROMA TEST!' with a character count of '10/2,048'. Below this, there is a checkbox for 'Backend Authentication' with the label 'Use custom authorizer for authentication'. At the bottom, there is a table header for 'Constant Parameter Name' and 'Parameter Value', and a '+ Add Header' button.

Table 2-4 Parameters for defining an API backend

Parameter	Description
Status Code	Select the HTTP status code returned by the API. Use 200 here.
Response	Enter the response result of the API. In this example, set this parameter to ROMA TEST! . Once you call the API, ROMA TEST! is returned.
Backend Authentication	Specify whether to use a custom authorizer for authentication. Skip this parameter.
Add Header	Customize the header parameters of the API response. Skip this parameter.

7. Configure response examples to help API callers understand the responses to an API request.

Figure 2-5 Response configuration

Table 2-5 Response parameters

Parameter	Description
Example Success Response	Example of a system response when an API is called successfully. In this example, set this parameter to ROMA TEST! .
Example Failure Response	Example of a system response when an API cannot be called. Skip this parameter.

8. Click **Finish**. After the API is created, the API details page is displayed.

Step 3: Debug the API

After you create an API, debug it to ensure that the API functions properly.

1. On the details page of the API created in [Step 2: Create an API](#), click **Debug** in the upper right corner.

2. Click **Debug** next to the API's access address. Request parameters are not required in this example because none has been set in previous steps.
3. View the request and response for API calling.
If the call is successful, a success response body is returned. In this example, **ROMA TEST!** is returned.

Figure 2-6 API debugging

Request	Response	Status Code	200	Duration	9 ms
1 GET /test HTTP/1.1	1 HTTP/1.1 200 OK				
2 Host: cb785e47317d4abfb0c1996f504f15bd.apic. .apis.com	2 Transfer-Encoding: chunked				
3 User-Agent: APIGatewayDebugClient/1.0	3 Connection: keep-alive				
4 X-Api-Authz: SDK-HMAC-SHA256 Access =a78580f5c32d47358a39cde3de41e684, SignedHeaders=user-agent ;x-apig-mode;x-sdk-date, Signature =24dc4ceb5b6921bf894416531159e4f5210f0a090690076f58dfe933419ee214	4 Content-Type: application/json				
5 X-Api-Mode: debug	5 Date: Tue, 14 Mar 2023 03:07:47 GMT				
6 X-Sdk-Date: 20230314T030747Z	6 Server: api-gateway				
7	7 X-Api-Latency: 0				
8	8 X-Api-Ratelimit-Api: remain:99,limit:100,time:1 minute				
	9 X-Api-Ratelimit-Api-Allenv: remain:5999,limit:6000,time:1 second				
	10 X-Request-Id: 006b59d9f76734e9dfa2a6ad84d437d6				
	11				
	12 ROMA TEST!				

4. Click **X** on the right of **Debug** to return to the APIs details page after debugging.

Step 4: Publish the API

Once the API is tested normal, publish it to an environment so that it can be called by other users.

1. On the API details page, click **Publish** in the upper right corner.
2. On the page displayed, set **Environment** to **RELEASE** and click **Publish**.

Figure 2-7 Publishing the API

API Name APL_test

* Environment [Create Environment](#)

Description

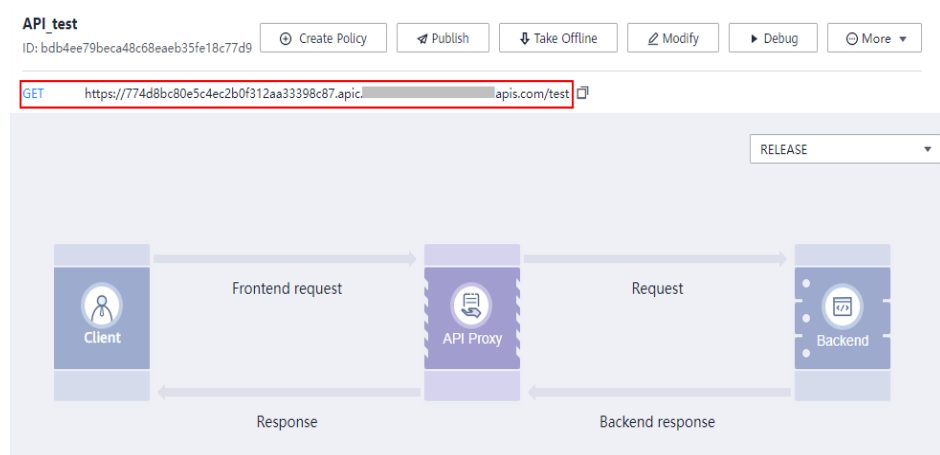
0/255

Step 5: Call the API

In this example, Postman is used to call the API.

1. Obtain request information of the API.
On the API details page, view the API calling information, including the request method and access address.

Figure 2-8 API request information

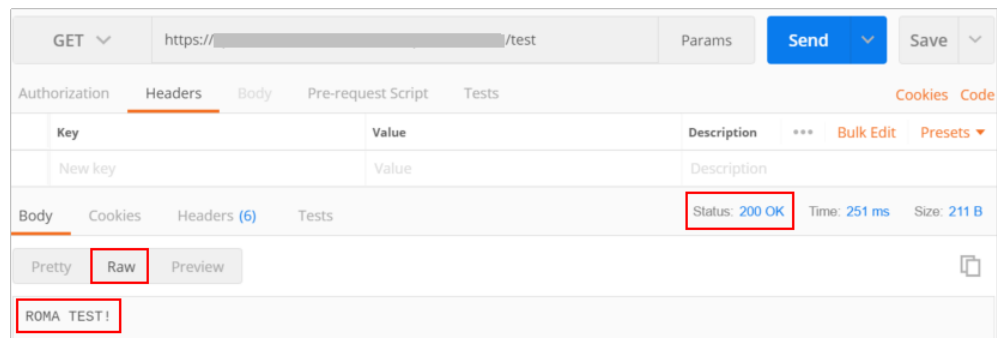


2. Call the API.

Use Postman to call the API obtained in 1.

After the API is successfully called, check whether the response is the same as the test result in **Step 3: Debug the API**.

Figure 2-9 API calling response



3 Using ROMA Connect for Message Integration

Overview

ROMA Connect provides secure and standard message channels to implement integration with different message systems.

This chapter provides a configuration example of interworking with ROMA Connect using Kafka command-line tool to help you get familiar with the message integration process.

The steps for integrating messages with ROMA Connect are as follows:

- [Step 1: Create a Message Topic](#)
- [Step 2: Send and Receive Messages to and from a Topic](#)

Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance, and the network where the local PC is located can communicate with the EIP network.
- The [Java JDK](#) has been downloaded and installed on your local PC, and related environment variables have been configured.
- The open-source Kafka command-line tool of the corresponding version has been downloaded based on the Kafka version of the ROMA Connect instance. You can view the Kafka version in the **MQS Information** area on the **Instance Information** page of the ROMA Connect console.
 - [Kafka command-line tool 1.1.0](#)
 - [Kafka command line tool 2.7.2](#)

NOTE

The Kafka server version of ROMA Connect can be 1.1.0 or 2.7. Use the Kafka command-line tool of the same version as the server to prevent unpredictable problems.

- If **MQS SASL_SSL** has been enabled for the ROMA Connect instance, ensure that the [client certificate](#) has been downloaded. To download the client

certificate, log in to the ROMA Connect console, choose **Message Queue Service > Topic Management**, and click **Download SSL Certificate**.

Step 1: Create a Message Topic

A topic is a channel for message transmission between the message client and ROMA Connect. The client sends messages to and receives messages from ROMA Connect through the topic.

1. Create an integration application.
 - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** of an instance.
 - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
 - c. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
2. Create a message topic.
 - a. In the navigation pane on the left, choose **Message Queue Service > Topic Management**. On the page displayed, click **Create Topic** in the upper right corner.
 - b. In the **Create Topic** dialog box, set topic parameters and click **OK**.

Figure 3-1 Creating a topic

The screenshot displays a configuration form for creating a topic. The fields and their values are as follows:

- Topic Name:** Topic-test
- Integration Application:** (Empty dropdown)
- Permission:** Publish/Subscribe (selected), Publish, Subscribe
- Partitions:** 3 (Value range: 1 to 100)
- Replicas:** 3 (Value range: 1 to 3)
- Aging Time (h):** 72 (Value range: 1 to 168)
- Synchronous Replication:** Off (Note: The acks parameter must be set to -1 on the producer client so that multiple replicas can be synchronously replicated.)
- Synchronous Flushing:** Off
- Tag:** Add (Note: In common service scenarios, leave this field blank.)
- Filter:** Add (Note: (Optional) Blocks the display of messages containing these values.)
- Description:** (Edit icon)

Table 3-1 Topic parameters

Parameter	Description
Topic Name	Enter a topic name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in 1 .
Permission	Select the topic permission for the integration application to which the topic belongs. In this example, select Publish/Subscribe , which indicates that the topic can be used to produce and consume messages.
Partitions	The appropriate number of partitions can improve the concurrent performance of message creation and retrieval. Retain the default value 3 .

Parameter	Description
Replicas	ROMA Connect automatically backs up data on each replica. If one replica is faulty, data will still be available. The reliability increases with the number of replicas of a topic. Retain the default value 3 .
Aging Period (h)	After the aging time expires, messages stored in the topic are deleted. Retain the default value 72 .
Synchronous Replication	When a client creates a message to a topic, ROMA Connect determines whether to replicate the message to all replicas and then returns a response to the client. You do not need to enable this option.
Synchronous Flushing	Specifies whether each message created by a client to the topic is immediately written to the disk. You do not need to enable this option.
Tag	A tag can be added for a topic to facilitate topic search. You do not need to add a tag.
Filter	(Optional) Add values to filter out topic messages containing these values.
Description	Enter a brief description of the topic.

Step 2: Send and Receive Messages to and from a Topic

Use the Kafka command-line tool on the local PC to send messages to and receive messages from topics in CLI mode.

The operations for sending and receiving messages to and from a topic vary depending on whether SASL_SSL is enabled for the ROMA Connect instance. If SASL_SSL is enabled, messages sent and received by clients to and from topics are encrypted before transmission.

1. Decompress the Kafka command-line tool and client certificates.

Locate the downloaded Kafka command-line tool and client certificate files on your local PC and decompress them.

The following uses Windows as an example. Assume that the Kafka command-line tool path is **D:\kafka_2.11-1.1.0** and the client certificate file path is **D:\cert**.

2. (Optional) Modify the **kafka-run-class.bat** file in the Kafka command-line tool. This step is required only if the Kafka command-line tool 1.1.0 is used. Otherwise, skip this step.

Find the **kafka-run-class.bat** file in the **D:\kafka_2.11-1.1.0\bin\windows** directory and add double quotation marks (") to **%CLASSPATH%** in the following script line:

```
set COMMAND=%JAVA% %KAFKA_HEAP_OPTS% %KAFKA_JVM_PERFORMANCE_OPTS%  
%KAFKA_JMX_OPTS% %KAFKA_LOG4J_OPTS% -cp "%CLASSPATH%" %KAFKA_OPTS% %*
```

- (Optional) Modify the configuration file of the Kafka command-line tool. This step is performed only if SASL_SSL is enabled for the ROMA Connect instance. Otherwise, skip this step.

Locate the **consumer.properties** and **producer.properties** files in **D:\kafka_2.11-1.1.0\config** and add the following content to the files separately:

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \  
username="*****" \  
password="*****";  
sasl.mechanism=PLAIN  
security.protocol=SASL_SSL  
ssl.truststore.location=D:/cert/client.truststore.jks  
ssl.truststore.password=dms@kafka  
ssl.endpoint.identification.algorithm=
```

In the preceding information:

- The values of **username** and **password** are the key and secret of the integration application to which the topic belongs. You can click the integration application name on the **Integration Applications** page of the ROMA Connect console to view and obtain the key and secret.
- The value of **ssl.truststore.location** is the path for storing the client certificate obtained in **1**.

NOTICE

The certificate path in Windows must contain slashes (/).

- On the command line window, run the following command to go to the directory where the Kafka command-line tool is stored:

```
d:  
cd kafka_2.11-1.1.0\bin\windows
```

- Produce messages to ROMA Connect.

- Run the following command to create a connection with the topic for producing messages:

If SASL_SSL is not enabled for the ROMA Connect instance, run the following command:

```
kafka-console-producer.bat --broker-list IP:9094,IP:9095,IP:9096 --topic TopicName
```

If SASL_SSL is enabled for the ROMA Connect instance, run the following command:

```
kafka-console-producer.bat --broker-list IP:9095,IP:9096,IP:9097 --topic TopicName --  
producer.config ../../config/producer.properties
```

In the preceding information:

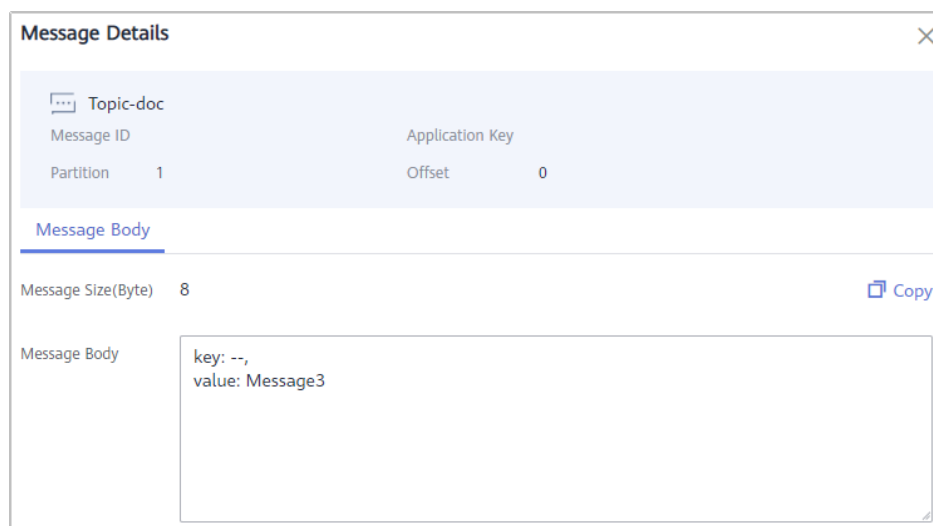
- IP** indicates the MQS connection address of ROMA Connect. You can obtain the address by viewing **EIP** on the **Instance Information** page of the ROMA Connect console.
 - TopicName** indicates the name of the topic created in **Step 1: Create a Message Topic**.
- Enter the message contents and send the messages to the topic.

```
>Message1  
>Message2  
>Message3
```

In the preceding information, **Message1**, **Message2**, and **Message3** indicate the actual message contents sent to the topic. Each row indicates a message.

- c. On the ROMA Connect console, choose **Message Queue Service** > **Message Query**.
- d. Click **Advanced Search** to expand advanced search options.
- e. Enter search criteria and click **Search** to query the messages sent by the client.
 - Select the message topic created in **Step 1: Create a Message Topic** for **Topic Name**.
 - Select **Creation time** for **Search By** and select the time segment for the client to send messages to ROMA Connect.
- f. Click **Message** next to the message record and check whether the message content is the same as that sent in **5.b**.

Figure 3-2 Querying messages



6. Consume messages from ROMA Connect.
 - a. Run the following command to create a connection with the topic for reading messages:

If SASL_SSL is not enabled for the ROMA Connect instance, run the following command:

```
kafka-console-consumer.bat --bootstrap-server IP:9094,IP:9095,IP:9096 --topic TopicName --from-beginning
```

If SASL_SSL is enabled for the ROMA Connect instance, run the following command:

```
kafka-console-consumer.bat --bootstrap-server IP:9095,IP:9096,IP:9097 --topic TopicName --from-beginning --consumer.config ../../config/consumer.properties
```

In the preceding information:

- **IP** indicates the MQS connection address of ROMA Connect. You can obtain the address by viewing **EIP** on the **Instance Information** page of the ROMA Connect console.

- **TopicName** indicates the name of the topic created in [Step 1: Create a Message Topic](#).
- b. After the command is executed, the system is continuously connected to the topic and reads messages. To stop reading messages, press **Ctrl+C**, enter **Y**, and press **Enter**.

4 Using ROMA Connect for Device Integration

Overview

ROMA Connect uses the standard Message Queue Telemetry Transport (MQTT) protocol to connect devices and implement quick and easy device management on the cloud.

This chapter provides a configuration example to help you get familiar with the device integration process.

The steps for integrating devices with ROMA Connect are as follows:

- [Step 1: Define a Product](#)
- [Step 2: Register a Device](#)
- [Step 3: Configure Data Forwarding Rules](#)
- [Step 4: Debug the Device](#)

Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance, and the network where the local PC is located can communicate with the EIP network.
- The [MQTTX client](#) has been downloaded and installed locally.
- A topic has been created for message integration in the ROMA Connect instance. For details, see [Creating a Message Topic](#).

Step 1: Define a Product

A product is a device model. You can define a product to determine the function properties of a device. Each device belongs to a product.

1. Create an integration application.
 - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** of an instance.

- b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
 - c. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
2. Create a product.
 - a. In the navigation pane on the left, choose **LINK > Product Management**. On the page displayed, click **Create Product** in the upper right corner.
 - b. In the **Create Product** dialog box, set product parameters and click **OK**.

Table 4-1 Parameters for creating a product

Parameter	Description
Product Type	Select Common as the product type.
Protocol Type	Select MQTT as the protocol type.
Integration Application	Select the integration application created in 1 .
Product Template	You do not need to set this parameter.
Product Name	Enter a product name, which can be user-defined.
Manufacturer Name	Enter a manufacturer name of the device, which can be user-defined.
Manufacturer ID	Enter a manufacturer ID of the device, which can be user-defined.
Product Model	Enter a product model of the device, which can be user-defined.
Device Type	Select Default Type as the type of the access device.
Model Version	Enter a device model version, which can be user-defined.
Description	Enter a brief description of the product.

3. Add a thing model service for the product.
 - a. On the **Product Management** page, click the **Products** tab and click the name of the product created in **2** to access the product details page.
 - b. On the **Thing Model** tab page, click **Create**.
 - c. In the **Create Thing Model Service** dialog box, set thing model service parameters and click **OK**.
Assume that the device is a temperature sensor. Create a temperature service **Temperature** for the product.

Table 4-2 Parameters for creating a thing model service

Parameter	Description
Name	Enter Temperature as the name of the thing model service.
Status	This parameter specifies whether to enable the thing model service. Retain the default value, that is, the thing model service is enabled.
Description	Enter a brief description of the thing model service.

4. Add a property for the thing model service.
 - a. In the left pane of the **Thing Model** tab page, select the thing model service added in 3.
 - b. On the **Attributes** tab page, click **Create**.
 - c. In the **Create Attribute** dialog box, set property parameters and click **OK**. Assume that the device reports only the property **temperature** whose data type is **String**.

Table 4-3 Parameters for creating a property

Parameter	Description
Name	Enter temperature as the property name.
Data Type	Select String as the data type of the property parameter reported by the device.
Mandatory	This parameter specifies whether the property must be reported by a device. Retain the default value, that is, the property must be reported by a device.
Description	Enter a brief description of the property.
Max. Data Length	Enter 32 as the maximum length of the property value.
Enumerated Value	You do not need to set this parameter.

5. Add a command for the thing model service.
 - a. On the **Commands** tab page, click **Create**.
 - b. In the **Create Command** dialog box, set command parameters and click **OK**. Assume that a device has the **SET_STATUS** command for setting the status, the command field is **status**, the data type is **String**, and the enumerated values are **on** and **off**.

Table 4-4 Parameters for creating a command

Parameter	Description
Name	Enter SET_STATUS as the command name.
Description	Enter a brief description of the command.


- c. Locate the created command in the command list and click  on the left of the command name to expand the command fields.
- d. Click **Create Field**.
- e. In the **Create Delivery Command Field** dialog box, set field parameters and click **OK**.

Table 4-5 Parameters for creating a command field

Parameter	Description
Name	Enter status as the command field name.
Data Type	Select String as the data type of the command field.
Mandatory	This parameter specifies whether the field must be carried in a delivered command. Retain the default value.
Description	Enter a brief description of the command field.
Max. Data Length	Enter 32 as the maximum length of the command field value.
Enumerated Value	Set this parameter to on,off , indicating that the device can be enabled or disabled.

Step 2: Register a Device

After a device is registered with ROMA Connect, a unique ID and key are allocated to the device to access ROMA Connect.

1. In the navigation pane on the left, choose **LINK > Device Management**. On the page displayed, click **Create Device** in the upper right corner.
2. In the **Create Device** dialog box, set device parameters and click **OK**.

Table 4-6 Parameters for creating a device

Parameter	Description
Integration Application	Select the integration application created in 1 .
Associated Product	Select the product created in Step 1: Define a Product .

Parameter	Description
Device ID	Enter a device ID, which is the unique identifier of the device and can be user-defined.
Device Name	Enter a device name, which can be user-defined.
Password	Enter the access password of the device. If you do not specify this parameter, the system automatically generates a value.
Confirm Password	Enter the confirm password of the device. If you do not specify this parameter, the system automatically generates a value.
Status	This parameter specifies whether to enable the device. The device can connect to ROMA Connect only after being enabled. Retain the default value, that is, the device is enabled.
Device Tag	You do not need to set this parameter.
Description	Enter a brief description of the device.

3. On the **Device Management** page, click the name of the created device to access the device details page. On the **Topics** tab page, view the topic information about the device.

Step 3: Configure Data Forwarding Rules

ROMA Connect does not directly store data reported by devices. You need to configure data forwarding rules to forward device data to other services for storage. The following section uses ROMA Connect MQS as an example to describe how to forward data to the destination.

1. In the navigation pane on the left, choose **LINK > Rule Engine**. On the page displayed, click **Create Rule** in the upper right corner.
2. In the **Create Rule** dialog box, set rule parameters and click **OK**.

Figure 4-1 Creating a rule

Create Rule

* Rule Name
Enter 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

* Integration application

Description
0/200

Status Enabled

Table 4-7 Rule parameters

Parameter	Description
Name	Enter a rule name, which can be user-defined.
Integration Application	Select the integration application created in 1 .
Description	Enter a brief description of the rule.
Status	This parameter specifies whether to enable the rule. Retain the default value, that is, the rule is enabled.

- In the rule list, click the name of the rule created in [2](#) to access the rule details page.
- In the **Source** area, click **Create Source**, set related parameters and click **Save**.

Figure 4-2 Creating a source

Source After a source is created or deleted, the message forwarding rule takes effect within 1 minute.

Product Name	Device Name	Topic Name	Topic Level	Base64 Encoding	Include Device Data	Operation
<input type="text" value="product-test"/>	<input type="text" value="device-test"/>	<input type="text" value="IJKES2548916/out/de..."/>	Device-level	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

Table 4-8 Source parameters

Parameter	Description
Product Name	Select the product created in Step 1: Define a Product .
Device Name	Select the device created in Step 2: Register a Device .

Parameter	Description
Topic Name	Select the topic to which the device sends messages. The topic name is in the format of <i>{Product ID}/out/{Device ID}</i> .
Topic Level	Select a topic level. The value is automatically adapted based on the value of Device Name . If you do not specify Device Name , Topic Level is Product . If you select a specific device from the Device Name drop-down list box, Topic Level is Device .
Base64 Encoding	Specifies whether Base64 encoding is performed on the forwarded device data. You do not need to enable this option.
Include Device Data	Specifies whether the forwarded data includes device data. To facilitate subsequent identification of the device to which the forwarded data belongs, enable this option.

- In the **Destination** area, click **Create Destination**, set related parameters and click **Save**.

Figure 4-3 Creating a destination

Destination After a destination is created or deleted, the message forwarding rule will take effect within 1 minute.

Destination	Connection Address	Topic Name	Username	Password	Operation
ROMA MQS	105.106.0.163:9093,105...	Topic-test	Enter a username.	Enter a password.	Save Cancel

Table 4-9 Destination parameters

Parameter	Description
Destination	Select ROMA MQS to integrate messages forwarded to ROMA Connect MQS.
Connection Address	Select the MQS intranet address of the current ROMA Connect instance. You can view the address on the Instance Information page of the ROMA Connect console.
Topic Name	Select the topic created for message integration in Prerequisites .
Username	This parameter is available only if MQS SASL_SSL is enabled for the ROMA Connect instance. Enter the key of the integration application to which the topic defined in Topic Name belongs.

Parameter	Description
Password	This parameter is available only if MQS SASL_SSL is enabled for the ROMA Connect instance. Enter the secret of the integration application to which the topic defined in Topic Name belongs.

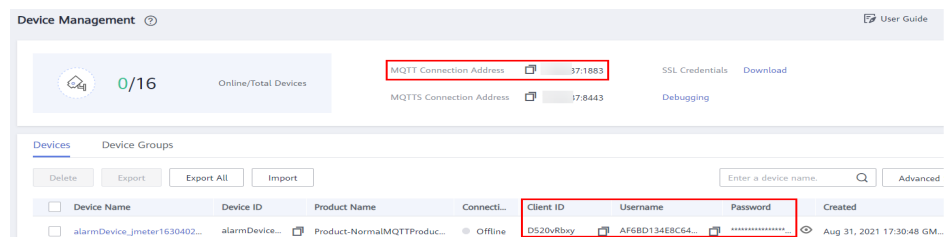
Step 4: Debug the Device

A device can access ROMA Connect by integrating the MQTTX client or SDK. Use the MQTTX client to simulate a device to connect and send messages to ROMA Connect.

1. Obtain device access information.

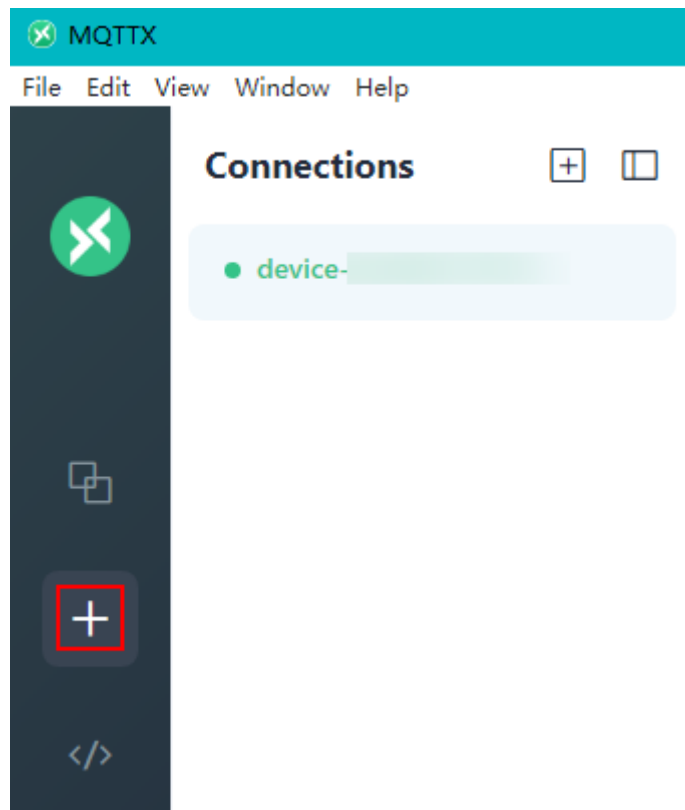
In the device list on the **Device Management** page, view and record the values of **MQTT Connection Address**, **Client ID**, **Username**, and **Password** of the device created in [Step 2: Register a Device](#).

Figure 4-4 Device access information



2. Open the MQTTX client installed on your local PC and click the icon for adding new connections on the client.

Figure 4-5 MQTTX client



3. On the page for editing connection properties, enter the connection property configuration information and click **Connect**.

Figure 4-6 Configuring connection properties



Table 4-10 Connection property configuration

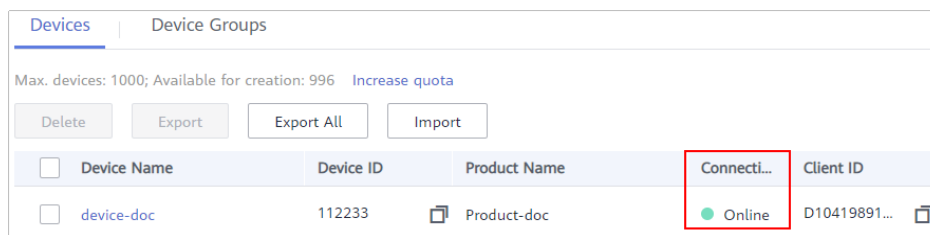
Parameter	Description
Name	Enter a name for the connection configuration.

Parameter	Description
Client ID	Enter the client ID obtained in 1.
Host	Select mqtt:// , and enter the IP address in MQTT Connection Address obtained in 1.
Port	Enter the port number in MQTT Connection Address obtained in 1.
Username	Enter the username obtained in 1.
Password	Enter the password obtained in 1.

4. Click **Connect** to connect to ROMA Connect.

If the connection is successful, the connection status in the upper left corner of the MQTTX client will be displayed in green. On the ROMA Connect console, choose **LINK > Device Management**. On the page displayed, **Status** of the device will be displayed as **Online**.

Figure 4-7 Device online




5. Enable the device to send data to ROMA Connect.
 - a. On the MQTTX client, click the connection created in 3, enter the device topic and message content, and click  to send a message to ROMA Connect.
 - Enter the topic of the message sent by the device in the text box in the format of *{Product ID}/out/{Device ID}*. You can log in to the ROMA Connect instance console, choose **LINK > Device Management**, click a device name to go to the details page, and view topics of the device on the **Topics** tab.
 - Enter the message content in JSON format in the message text box.

Figure 4-8 Enabling a device to send a message



- b. On the ROMA Connect console, choose **Message Queue Service** > **Message Query**.
- c. Click **Advanced Search** to expand advanced search options.
- d. Enter search criteria and click **Search** to query the messages sent by the client.
 - Select the message topic created in **Step 3: Configure Data Forwarding Rules** for **Topic Name**.
 - Select **Creation time** for **Search By** and select the time segment for the device to send messages to ROMA Connect.
- e. Click **Message Body** and check whether the value of the **data** field in the message reported by the device is the same as that sent by the MQTTX client.

Figure 4-9 Viewing the message body



6. Enable the device to receive a command from ROMA Connect.
 - a. On the MQTTX client, click the connection created in **3**, and click **New Subscription**, enter the topic (in the format of **/v1/devices/{Device ID}/command**) of the command delivery request, and click **Confirm** to subscribe to the device command delivery message.
You can log in to the ROMA Connect instance console, choose **LINK** > **Device Management**, click a device name to go to the details page, and view topics of the device on the **Topics** tab.

Figure 4-10 Subscribing to device messages

The 'New Subscription' dialog box includes the following fields and controls:

- Topic:** A text input field containing the path `/v1/devices/d[redacted]5/command`.
- QoS:** A dropdown menu showing '0' and 'At most once'.
- Color:** A text input field showing the hex color `#86B73A`.
- Alias:** An empty text input field.
- Buttons:** 'Cancel' and 'Confirm' buttons at the bottom right.

- b. On the ROMA Connect instance console, choose **LINK > Device Management**. In the **Operation** column of the device, choose **More > Deliver Command**.
- c. In the **Deliver Command** dialog box, set the command to be delivered and click **OK**.

Figure 4-11 Delivering a command

The 'Deliver Command' dialog box includes the following fields and controls:

- Service:** A dropdown menu showing 'Temperature' and a 'Create' button.
- Synchronous Communication:** A toggle switch currently turned off.
- Command:** A dropdown menu showing 'SET_STATUS' and a 'Create' button.
- status:** A text input field containing the value 'on'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

Table 4-11 Parameters for delivering a command

Parameter	Description
Service	Select the thing model service added in 3. In this example, select Temperature .
Synchronous Communication	Retain the default value Disable .
Command	Select the command added in 5. In this example, select SET_STATUS .
status	This parameter varies according to the command field added in 5. In this example, the status field is displayed and set to on , indicating that the device is enabled.

- d. On the **Received** tab page of the MQTTX client, you can view the command delivered by ROMA Connect to the device. Check whether the values of **cmd** and **paras** in the command are the same as those set on the ROMA Connect console.

Figure 4-12 Command received by the device

```

Topic: /v1/devices/d[redacted]6/command QoS: 0
{"mid":1,"serviceId":"Temperature","deviceId":"D:[redacted]","cmd":"SET_STATUS",
"paras":
{"status":"on"},"msgType":"cloudReq"}
    
```

5 Using ROMA Connect for Composite Applications

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

This chapter describes how to expose an API that returns **hello world** via a composite application.

The steps for using composite applications with ROMA Connect are as follows:

Step 1: Creating a Composite Application

Step 2: Configuring a Workflow

Step 3: Publishing an Application

Step 4: Viewing Results

Prerequisites

Before performing this operation, complete the following preparations:

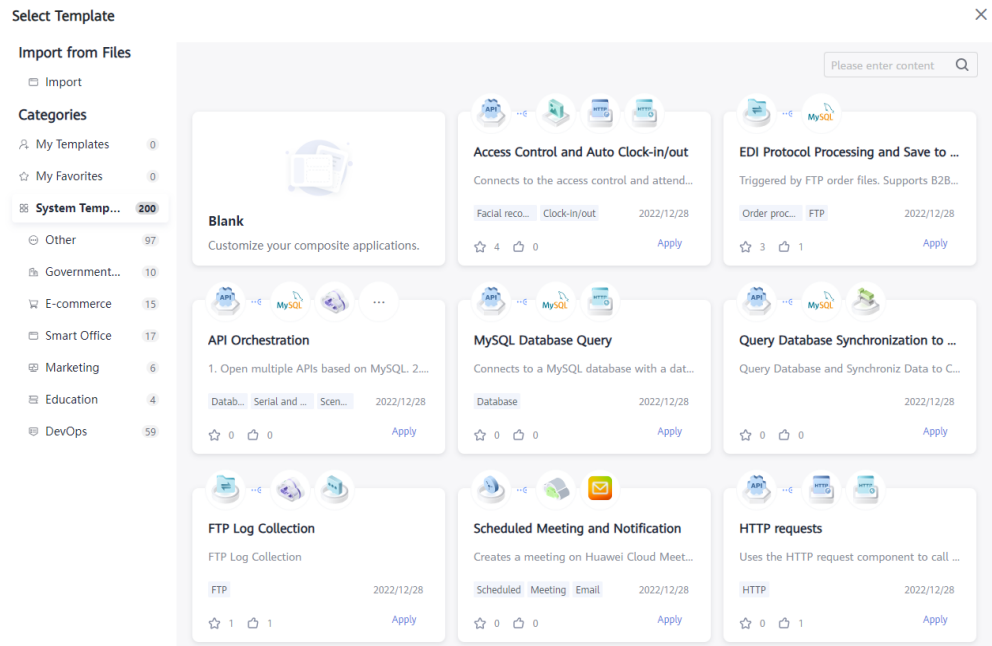
- You have created a ROMA Connect instance (new version).
- You have downloaded and installed [Postman](#) on your local PC.

Step 1: Creating a Composite Application

1. Log in to the ROMA Connect console (new version).
2. In the navigation pane on the left, choose **Application**. On the page displayed, click **Create**.
3. Select the blank template to customize a composite application.

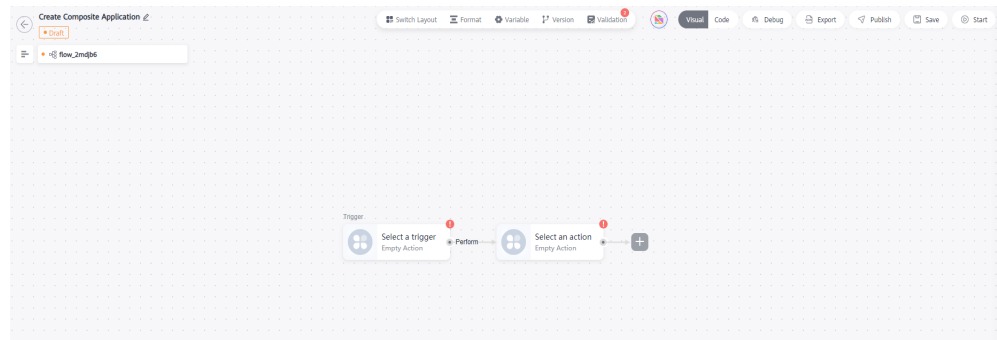
You can also select a service template to create a composite application. This section uses a blank template as an example.

Figure 5-1 Selecting a blank template



4. On the editing page for creating a composite application, design the application according to the instructions on the canvas.

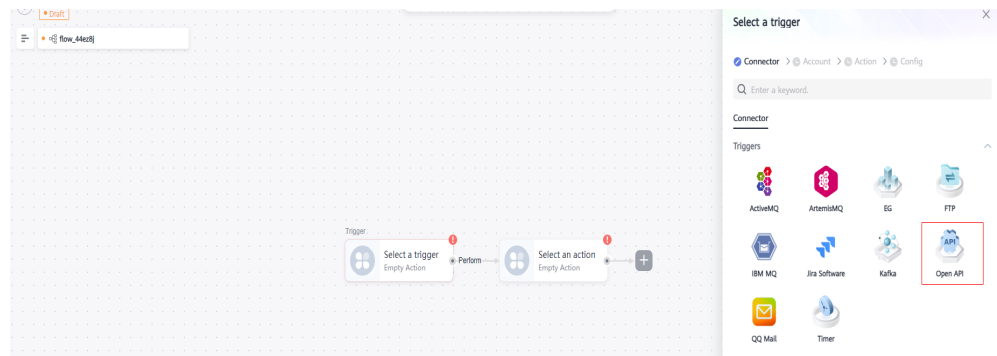
Figure 5-2 Creating a composite application



Step 2: Configuring a Workflow

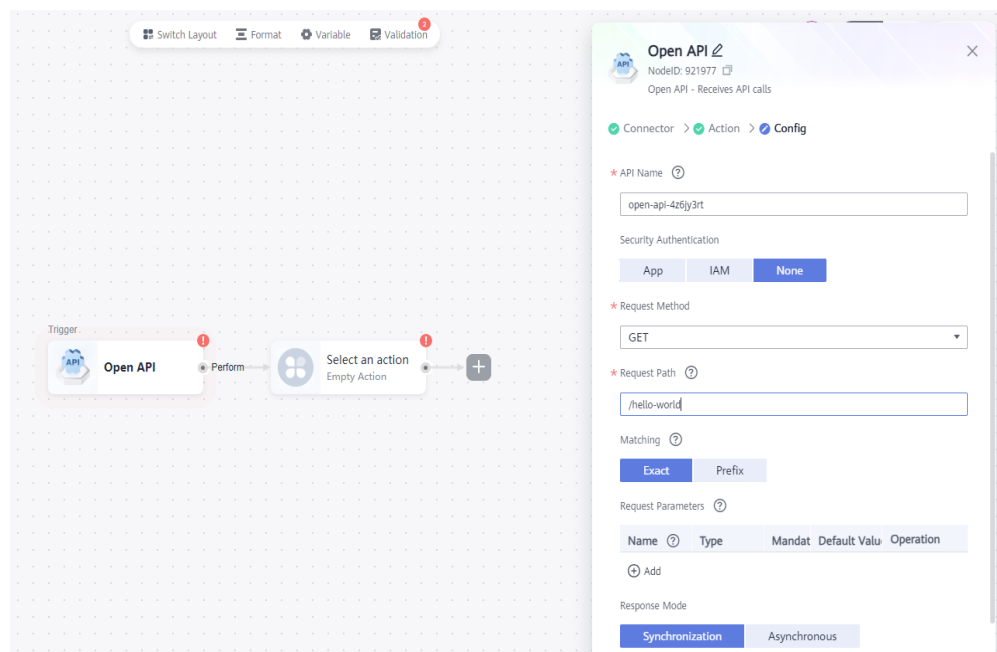
1. Click the first empty node and select **OpenAPI** under the **Triggers** tab from the panel on the right. The start node of each workflow (one default workflow for a blank template) must be a trigger.

Figure 5-3 Selecting a trigger



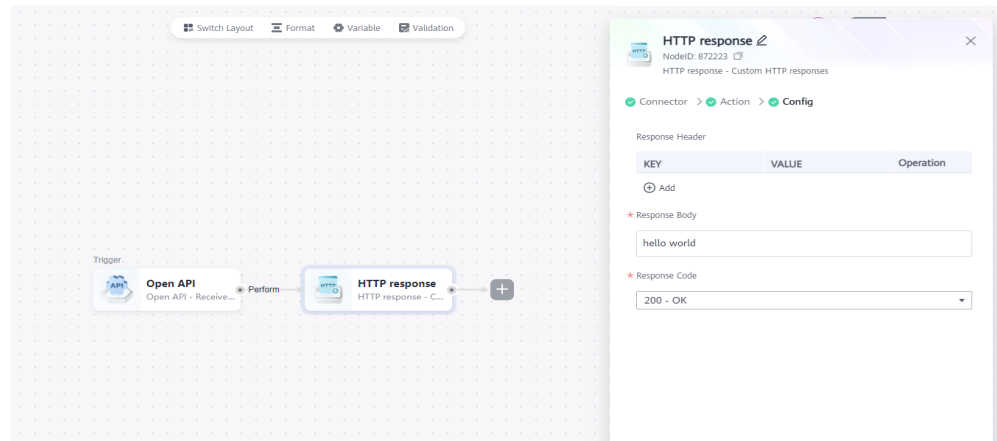
2. Configure the OpenAPI trigger by referring to the following information. For details, see [OpenAPI](#).
 - **Security Authentication:** Select **None**.
 - **Request Method:** Select **GET**.
 - **Request Path:** Enter **/hello-world**.

Figure 5-4 Configuring OpenAPI



3. Click the empty node, select **HTTP response** under the **Triggers** tab from the panel on the right, and set the parameters by referring to the following information. For details, see [HTTP Response](#).
 - Response Header:** Leave it blank.
 - Response Body:** Enter **hello world**.
 - Response Code:** Select **200 - OK**.

Figure 5-5 Configuring the HTTP response

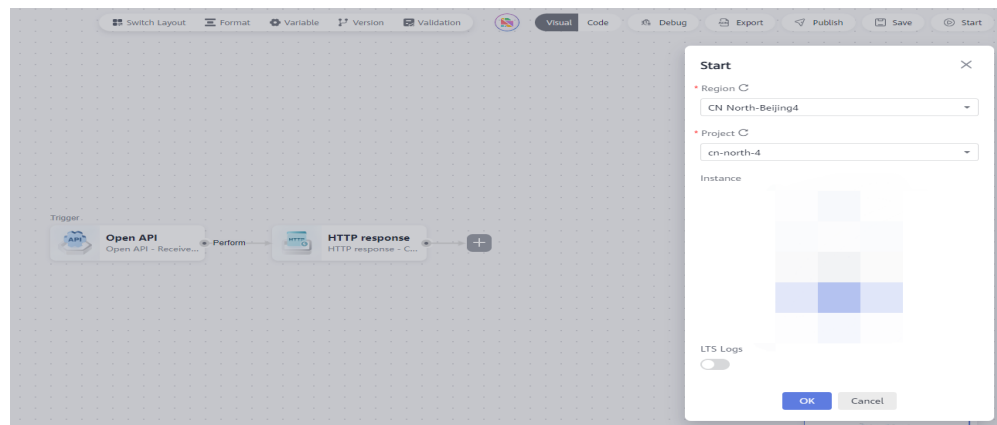


4. In the upper right corner of the page, click **Save**. In the dialog box displayed, set **Name** and **Description**, and click **Yes**.

Step 3: Publishing an Application

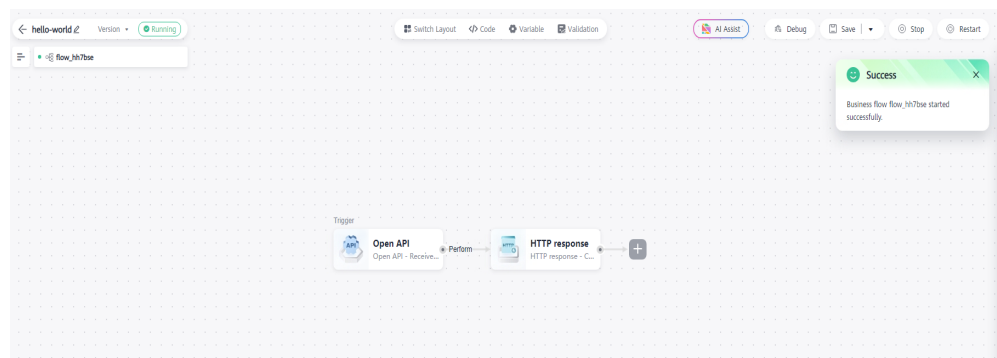
1. In the upper right corner of the page, click **Start**. In the dialog box displayed, set **Region** and **Project**, select an instance, and click **OK**.

Figure 5-6 Starting a composite application



2. Check that a message is displayed in the upper right corner of the canvas. If yes, the workflow is started successfully.

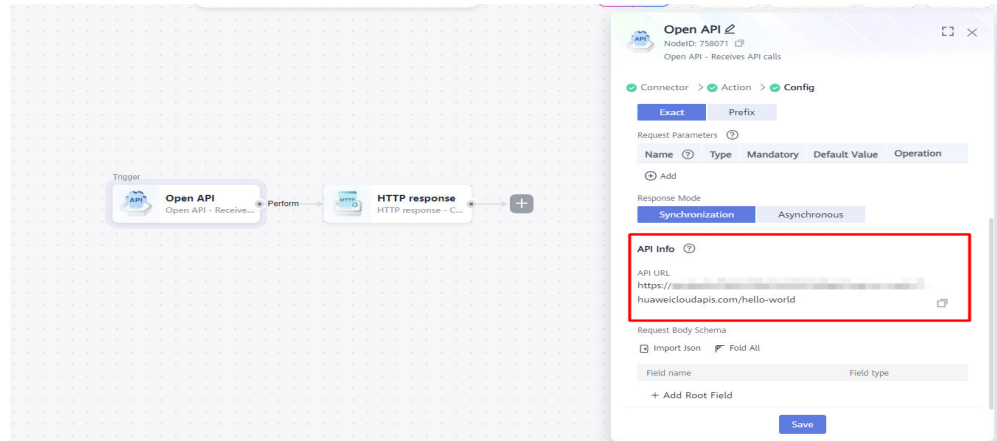
Figure 5-7 Composite application started successfully



Step 4: Viewing Results

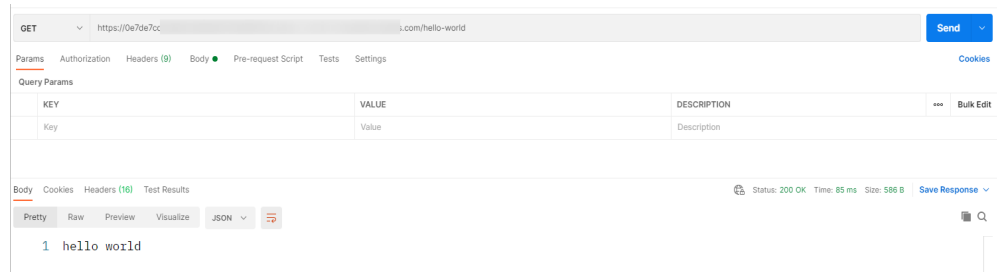
1. On the canvas for editing the composite application, click the OpenAPI trigger and view the API information in the lower part of the panel.

Figure 5-8 API information



2. As **configured in previous steps**, this API uses the GET request method without authentication. Use Postman to test the API information in 1. As shown in the following figure, when the API is accessed, the system returns **hello world**.

Figure 5-9 API testing result



6 Getting Started with Common Practices

After a ROMA Connect instance is created, the following common practices are available for you to choose from based on service requirements.

Table 6-1 Common practices

Practice	Description
Digital Reconstruction of Traditional Parking Lot Management Systems	Uses ROMA Connect to digitize parking lot management systems for intelligent management of parking lots.
Sharing Enterprise Data Using APIs	Uses ROMA Connect APIC to open databases as RESTful APIs, so that data between different service systems can be shared through APIs. Multiple security authentication modes are available to secure access of application system data across networks and regions.
Integrating and Converting Service Data Across Systems	Uses ROMA Connect FDI to convert the format of data in service system A and integrate the data into the database of service system B, and synchronize new data between the two systems periodically according to a preset plan.