Huawei Cloud Astro Zero

FAQs

Issue 01

Date 2025-08-28

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

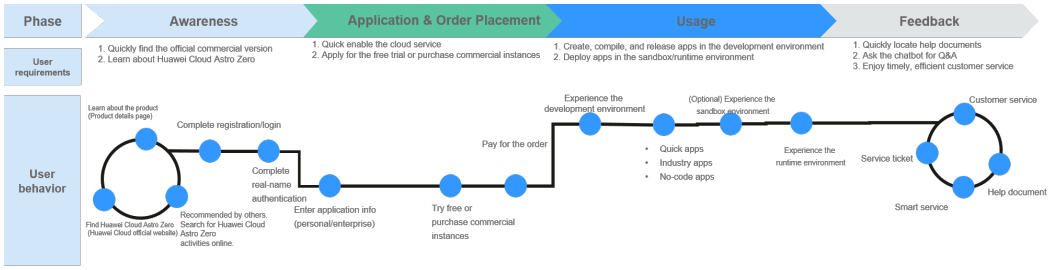
Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Common FAQs

Using Huawei Cloud Astro Zero consists of four phases, see **Figure 1-1**. The cases provided in this document cover three phases: product understanding, deployment, and use, that is, the first three phases in **Figure 1-1**.

Figure 1-1 Using Huawei Cloud Astro Zero



Product Consulting

- **What Programming Languages Does Huawei Cloud Astro Zero Support?**
- **Does Huawei Cloud Astro Zero Support Third-Party Databases?**
- **How Do I Disable Email Notification for Huawei Cloud Astro Zero Service Thresholds?**

Application Backend Development

- **How Do I Distinguish Database Objects of Different Accounts?**
- **Why Cannot Referenced Resources Be Edited in Flows or BPMs?**
- **How Can a Resource Be Orchestrated and Called by Multiple Applications?**
- **What Languages and Libraries Does Huawei Cloud Astro Zero Support?**
- **How Do I Use Huawei Cloud Astro Zero Scripts to Export an XLS File for Users to Download?**

Application Frontend Development

- **How Do I Quickly Select Widgets on a Standard Page with Many Widgets?**

- **How Do I Run a Logic Segment After All Widgets Have Loaded on an Advanced Page?**
- **How Do I Enable Collaborative Development?**
- **How Do I Control User Permissions During Collaborative Development?**

2 Product Functions

2.1 What Shortcuts Are Supported When I Use Huawei Cloud Astro Zero to Develop Apps?

Huawei Cloud Astro Zero support shortcuts for flows, scripts, triggers, decision tables, data access, state machines, and advanced pages.

Table 2-1 Shortcuts

Shortcut	Description
Alt+F2	Obtains a lock. After user A obtains the lock, the editor page is locked. User A can perform various configurations to prevent other users from editing the page at the same time.
Alt+F3	Release a lock. After the configuration is complete, user A needs to release the lock so that other users can obtain the lock again and perform editing operations.
Alt+Shift+P	Release the current page (the page is automatically saved).
Alt+Shift+V	Preview the current page.
Alt+L	Display the widget list.
Alt+Shift+E	Disable or enable flows, BPMs, scripts, and data access.
Ctrl+S	Save the settings.
Alt+Shift+S	Save as a new version, instance, or flow.
Ctrl+R	Run flows, BPMs, scripts, and data access.
Ctrl+Q	Stop the data access operation that is being performed.

Shortcut	Description
Ctrl+Z	Cancel the modification of flows, scripts, and widgets. Currently, this shortcut is supported for advanced pages but not for large screen pages.
Ctrl+Y	Restore the modification of flows, scripts, and widgets.
Alt+Shift+L	Open the tracker page.
Alt+Shift+R	Shortcut of the BPM editor page, which is used to open the BPM runtime-state configuration page.
Alt+Shift+P	Shortcut for opening the attribute configuration page of the trigger and script editor pages. Shortcut for advanced page development, used to release pages.
Alt+Shift+W	Shortcut for advanced page development, used to switch the development page to the PC style.
Alt+Shift+M	Shortcut for advanced page development, used to switch the development page to the mobile style.
Alt+Shift+C	Copy the data push address to the clipboard during data access.
Ctrl+Shift+K	Delete a line in the script editor.

2.2 How Does Huawei Cloud Astro Zero Manage Product Data?

Huawei Cloud Astro Zero provides the import and export capabilities of data tables. Developers can customize the import and export logic as required to import and export data in batches. You can export the data in the current system to migrate and back up data. For example, you can migrate the old metadata to another environment or account for object consistency and compatibility. In this way, data can be migrated between different accounts or environments, avoiding repeated or incompatible development.

- For details about how to import and export metadata, see **Importing and Exporting Object Metadata of Applications**.
- For details about how to import and export data, see **Importing and Exporting Object Data in Applications**.

2.3 What Programming Languages Does Huawei Cloud Astro Zero Support?

Huawei Cloud Astro Zero is a low-code development platform tailored for industry customers, partners, and developers. It allows you to develop common

applications through drag-and-drop operations, without relying on any programming language. We use TypeScript for scripts, JavaScript for page events, and Go for the underlying layer.

2.4 What Types of Data Does Huawei Cloud Astro Zero Support?

Data connections of Huawei Cloud Astro Zero can be classified into the following types:

- Message data

  Use the data access capability provided by Huawei Cloud Astro Zero to obtain data from external data sources (such as ROMA MQS, Kafka, and IoTDA). For details, see **Integrating Message Data into Huawei Cloud Astro Zero Through the Data Access Function**.

- API data

  Use **Rest services**, **SOAP services**, or HTTP standard library **scripts** provided by Huawei Cloud Astro Zero to make third-party API requests.

2.5 Does Huawei Cloud Astro Zero Support Third-Party Databases?

Yes. In Huawei Cloud Astro Zero, you can create a ROMA connector to interconnect with ROMA Connect. After the interconnection, you can call ROMA Connect APIs to indirectly access the database. For details, see **Interconnecting with ROMA Connect to Call Service APIs**.

2.6 When Does Huawei Cloud Astro Zero Need to Access Databases?

Huawei Cloud Astro Zero accesses databases in the following scenarios:

- Service data model construction

  Create objects in a visualized manner supported by Huawei Cloud Astro Zero.

- CRUD operations on service data

  This function can be implemented using either the database standard library in Huawei Cloud Astro Zero scripts or through visual orchestration of related diagram elements in flows.

- Simple service data query test

  Use the data debugging tool on the Huawei Cloud Astro Zero console to query table data using SQL statements. For details, see **Querying Object Data Using SQL Statements**.

- Frontend display of business data

  The frontend page accesses the database to obtain data from data sources for display.

2.7 How Does Huawei Cloud Astro Zero Ensure Performance in the Case of High Concurrency and Big Data?

The Huawei Cloud Astro Zero engine layer is implemented in Go. Goroutines of the Go language supports high concurrency. In addition, ELB and Nginx are used at the network layer for load balancing to dynamically balance service loads.

2.8 Which Browsers Can Be Used to Access Huawei Cloud Astro Zero?

The application development process consists of three stages: **development, testing, and deployment**. Each stage includes specific environments: development, sandbox, and runtime. For optimal performance, we recommend using Google Chrome to access these environments. Alternatively, other browsers may also work, but they will not provide the same level of access or user experience as Google Chrome.

3 Application Backend Development

3.1 How Do I Distinguish Database Objects of Different Accounts?

Huawei Cloud Astro Zero uses metadata to store data. The data of a tenant includes preset and tenant-defined metadata. The metadata preset in Huawei Cloud Astro Zero is identical and stored in containers for tenants to share. Tenant-defined metadata is physically stored together and logically isolated.

3.2 How Do I Troubleshoot the Reference Error When a Field Is Deleted from the Huawei Cloud Astro Zero Object and Then Created Again?

Description

Object fields used in page references, process references, and trigger applications are referenced by field IDs. If a field is deleted and a field with the same name is recreated, the field ID changes. Huawei Cloud Astro Zero regards them as two different fields. As a result, the reference is incorrect.

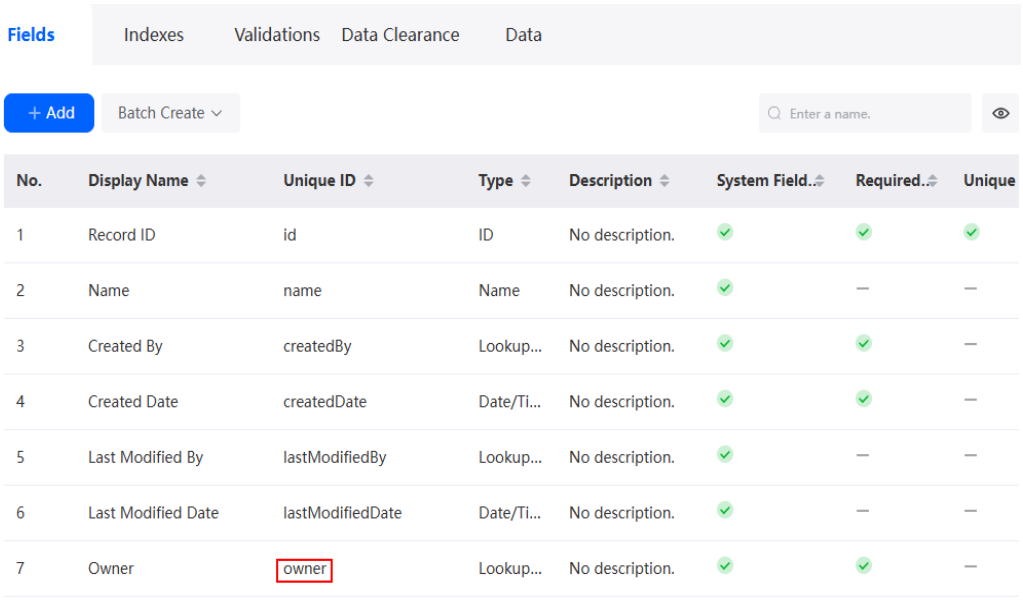
Solution

Open and edit the trigger, select the field that is deleted and recreated, and save and enable the field.

3.3 What Users Are Recorded by the owner Field Preset in a Huawei Cloud Astro Zero Custom Object?

Huawei Cloud Astro Zero presets the **owner** field for all custom objects, as shown in **Figure 3-1**.

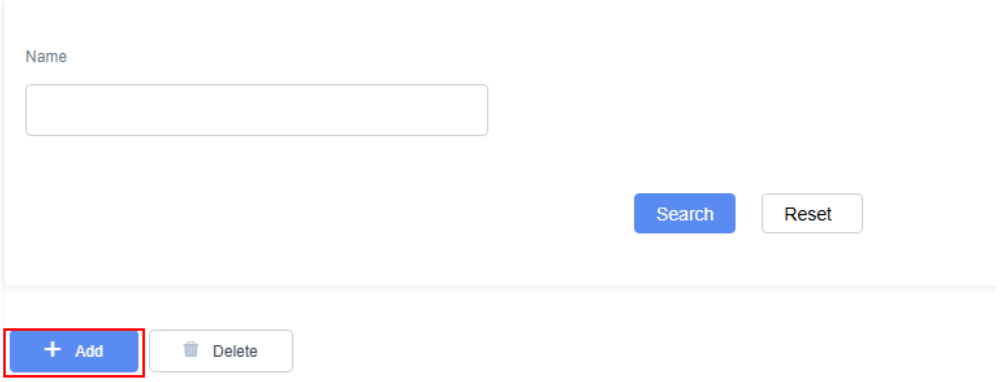
Figure 3-1 Preset owner field



The **owner** field records the following data:

- For the object data entered by a user on the GUI, the **owner** field records the ID of the user who enters the data.

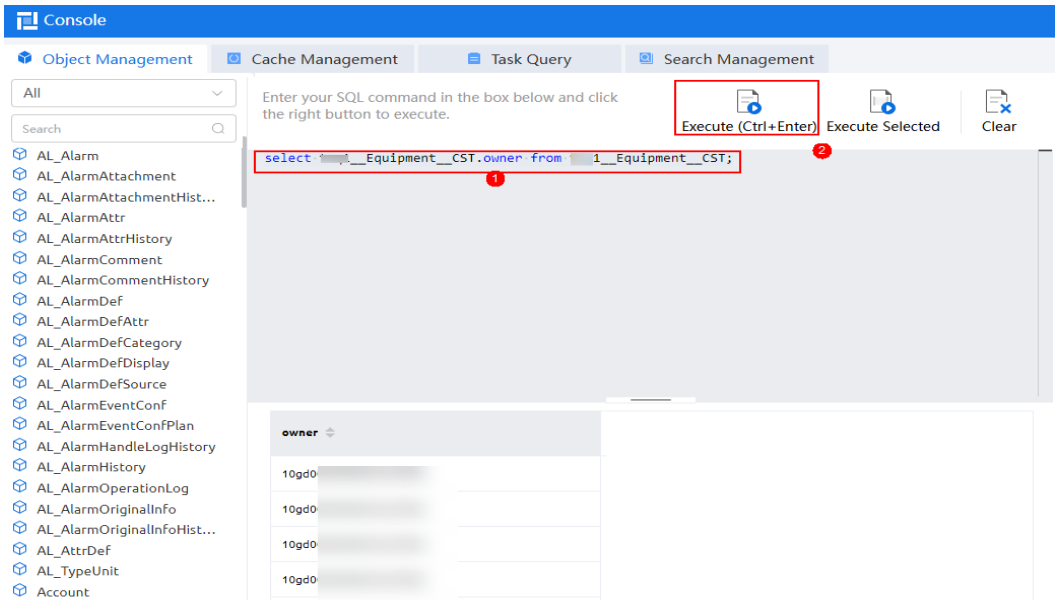
  Figure 3-2 Adding data on the GUI

- For object data entries triggered by flows or scripts through scheduled tasks, the **owner** field records the user ID of the task creator.
- For the object data recording triggered by events, the **owner** field records the system user ID **00000000000000000001**.

In the **data debugging (development environment)**, run the SQL statement **select *Namespace_Object*.owner from *Namespace_Object*;** to obtain the value of **owner**.

Figure 3-3 Querying the owner value



3.4 What Languages and Libraries Does Huawei Cloud Astro Zero Support?

In the script editor of Huawei Cloud Astro Zero, TypeScript is used to create scripts, and JavaScript is used for page scripts. For scripts, only the standard library APIs encapsulated by the platform can be used.

3.5 Can I Not Use @useObject to Modify Data Objects When Executing SQL Statements in Huawei Cloud Astro Zero Scripts?

Scenario

The method shown in the following figure is used when **db.sql().execute()** is executed in the script. That is, data objects are dynamically used.

Figure 3-4 Dynamically using data objects

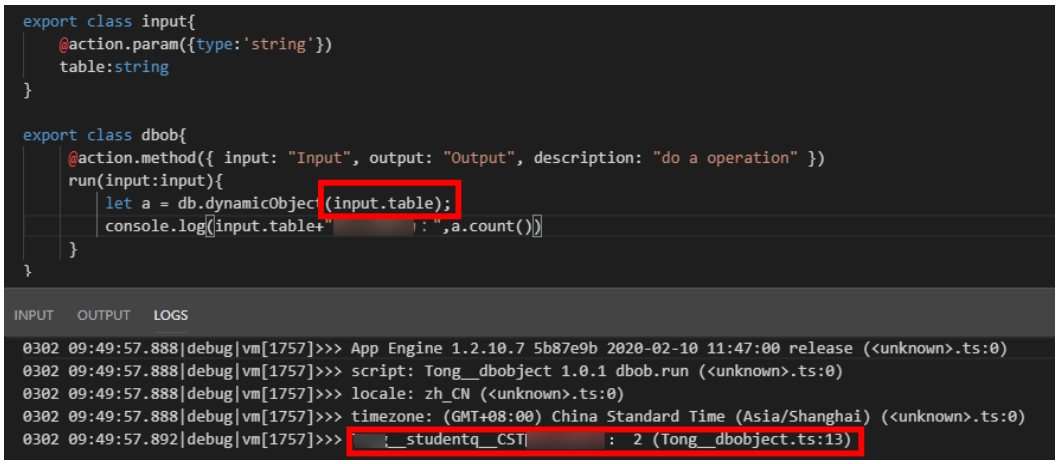
```
2214  /**
2215   * create a custom object handler
2216   * @param name the custom object's name
2217   * @param dynamic the dynamic object that does not need use @useObject to mark it
2218   */
2219  export declare function object(name: string, dynamic?: boolean): Orm;
2220  /**
2221   * create a dynamic object handler
2222   * @param name the object name
2223   */
2224  export declare function dynamicObject(name: string): Orm;
```


For example, an error will be reported if you do not use **@useObject(['User'])** when using **db.sql().execute("SELECT * FROM User")**. Do not use **@useObject(['User'])** when using **db.dynamicObject("User")**.

Solution

The dynamicObject method is used when the object table to be operated is unknown. For example, a value is assigned externally in the form of a parameter instead of a specific table name. The db.object method is used when the object table to be operated is determined. Therefore, @useObject needs to be used for declaration, as shown in the following figure.

Figure 3-5 Using @useObject for declaration



3.6 How Do I Return the Content of a ZIP File in the Huawei Cloud Astro Zero Script?

Example script for returning the content of a .zip file:

```
import * as zip from 'zip'
import * as buffer from 'buffer'
import * as text from 'text';
import * as context from 'context';

let bytes = text.bytes('I want to be binary');

let buff = buffer.from("some data");

let fileMap = {
    "/testdir/test.txt": "this is the data in test.txt",
    "/testdir/demo/bytes.txt": bytes,
    "/testdir/demo/buff.txt": buff.bytes(),
}

let stream = zip.compression(fileMap);

let resp = context.getHttp().response;
resp.setHeader("Content-Type", "application/zip");
let size = len(stream);
resp.setHeader('Content-Length', <string><any>size);
// Only the ZIP binary stream is returned to the browser. The browser needs to parse the stream based on Content-Type.
resp.setBody(stream);
```


3.7 How Do I Use Huawei Cloud Astro Zero Scripts to Export an XLS File for Users to Download?

Description

Huawei Cloud Astro Zero provides a standard Excel script library for operating Excel files, for example, generating Excel files. The standard output format of Huawei Cloud Astro Zero script orchestration is JSON. Therefore, some special processing is required so that the system can generate non-JSON data to export .xlsx files.

Procedure

Step 1 Create a blank script named **cube__download** by referring to **Creating a Blank Script**.

Step 2 In the script editor, enter the following script code:

```
import * as excel from 'excel';
import * as context from 'context';
import * as buffer from 'buffer';

export class Downlaod {

    @action.method({ input: "Input", output: "Output", description: "do a operation" })
    run(): void {
        // Raw data, which can be obtained from the data table. Static data is used here for demonstration.
        let binary = excel.encode(['a', 'b', 'c'], [{ 'a': 1, 'b': 2, 'c': 3 }, { 'a': 4, 'b': 5, 'c': 6 }, { 'a': 7, 'b': 2, 'c': 3 }]);

        // For an .xlsx file of Excel 2007 or later, set the content type to application/vnd.openxmlformats-officedocument.spreadsheetml.sheet.
        context.getHttp().response.setHeader('Content-Type', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;base64');
        // Return the binary value as the request body.
 context.getHttp().response.setBody(buffer.fromBytes(binary).toString(buffer.Encoding.Base64));
    }
}
```

In this case, the backend returns a Base64-encoded binary data, which is not in JSON format. Therefore, no output is generated when the script is directly executed. You need to match the data on the frontend page to process it.

Step 3 On the standard page, call the cube__download script.

To simplify the configuration, the script is not encapsulated into a public API. In actual use, the service needs to use the public API for encapsulation to implement finer-grained permissions management.

The following uses the ajax() method of jQuery as an example:

```
var url = "/u-route/baas/script/v1.0/run/cube__download";

context.$utils.getCSRFToken().then(function(token) {
    $.ajax({
        type: "POST",
        headers: {
            'Content-Type': "application/json",
            'responseType':"arraybuffer",
            'csrf-token': token,
        },
```



```
        url: url,
        data: JSON.stringify({}),
        dataType: 'text',
        async: false,
        success: function(resp){
            var fileName = "test" + '.xlsx';
            var file = new Blob([s2ab(atob(resp))], {type: ''});
            if (window.navigator.msSaveOrOpenBlob) { //Download using Internet Explorer
                window.navigator.msSaveOrOpenBlob(file, fileName);
            } else {
                var fileUrl = URL.createObjectURL(file);
                var a = document.createElement('a');
                a.href = fileUrl;
                a.target = '_blank';
                a.download = fileName;
                document.body.appendChild(a);
                a.click();
            }
        },
        error: function(resp){
        }
    });
});
```

s2ab() is the key to data processing. This method is short for "string to array buffer", which is used to convert a character string to the ArrayBuffer type so that the frontend can process binary data. The function is defined as follows:

```
function s2ab(s) {
    var buf = new ArrayBuffer(s.length);
    var view = new Uint8Array(buf);
    for (var i=0; i!=s.length; ++i) {
        view[i] = s.charCodeAt(i) & 0xFF;
    }
    return buf;
}
```

Pay attention to the following in the preceding code:

- Add **'responseType':"arraybuffer"** to the header.
- The type in the header is **dataType: 'text'**.
- Add csrf-token to the header.

----End

3.8 Do Huawei Cloud Astro Zero Scripts Support multipart/form-data Requests?

The standard script library provides multipart to support multipart/form-data requests. The sample code is as follows:

```
import * as http from 'http';
import * as mp from 'multipart';
import * as buffer from 'buffer';

let w = mp.newWriter();

let bytes = buffer.from("hello, Astro Zero");

w.setBoundary("--ABC");

let mimeHeader = {
  "Content-Disposition": ['form-data; name="upload_file"; filename="a.txt"'],
  "Content-Type": ['application/octet-stream']
```

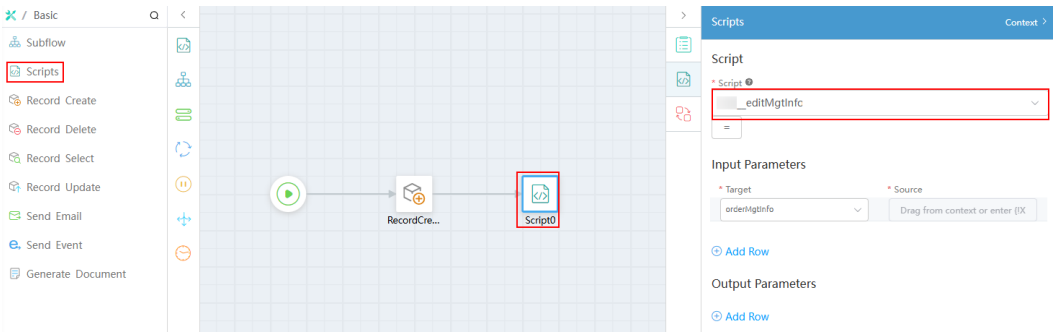


```
}

w.writeBuffer(mimeHeader, bytes);

w.writeField("name", "Trump");
w.close();
let client = http.newClient();

let req : http.Request = {
  data: w.buffer().bytes(),
  headers: {
      "Content-Type": w.formDataContentType(),
  }
}
let resp = client.post('ip', req);

console.log("response = ", resp);
```

3.9 Can I Roll Back a Script Diagram Element That Involves Data Operations in Flows If Other Diagram Elements Fail to Be Operated?

If a script involves operations such as adding, deleting, and modifying data, a rollback is performed by default for the diagram element after other script diagram elements failed. If an independent transaction is written in the script, the rollback is not performed.

Figure 3-6 Script diagram element



The following shows the format of an independent transaction. If the script includes the bolded content, the independent transaction is enabled. Even if the flow referencing this transaction fails, the transaction operation will not be rolled back.

```
import * as db from 'db';

@userObject(['student__cst'])
class Demo{
  test():void{
      let student = db.object('student__cst');
      //Perform some database operations.
      let op =function(): viod{
          //Perform other database operations.
      };
   //The database operations in op are independent transactions and do not affect the transaction operations outside op.
      db.transaction(op)
```



```
    }
  }
```

Use **db.transaction** to perform a series of operations in a transaction, either all successful or all failed. The success or failure of **db.transaction** is independent of external operations, and vice versa.

3.10 Why Does the Event-Triggered Flow Not Run After an Event Is Sent?

Description

After an event is sent, the event-triggered flow and the associated state machine did not run.

Solution

Step 1 Check whether the flow has been activated.

- If the fow is not activated, click  above the flow editing area to activate it.
- If yes, go to **Step 2**.

Step 2 Check whether the event is enabled.

- If the event is not enabled, click the enable button on the basic information tab page of the event to enable the event.
- If yes, go to **Step 3**.

Step 3 On the subscription tab page of the event, check whether the flow exists in the subscriber list.

- If not, check whether the startup type of the flow is **Event Trigger** and whether the triggering event is correctly configured.
- If yes, go to **Step 4**.

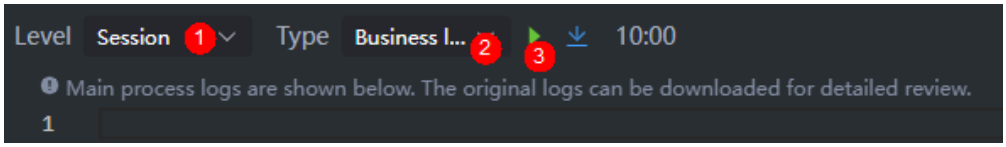
Step 4 Use the log tracing function to further analyze the logs.

1. On the Huawei Cloud Astro Zero console, click **Access Homepage** to go to the application development page.
2. In the upper left corner of the page, click  and choose **Environments** > **Environment Configuration**.
3. On the environment configuration page, click  on the right to enter the log tracing page.

Figure 3-7 Accessing the log tracing page

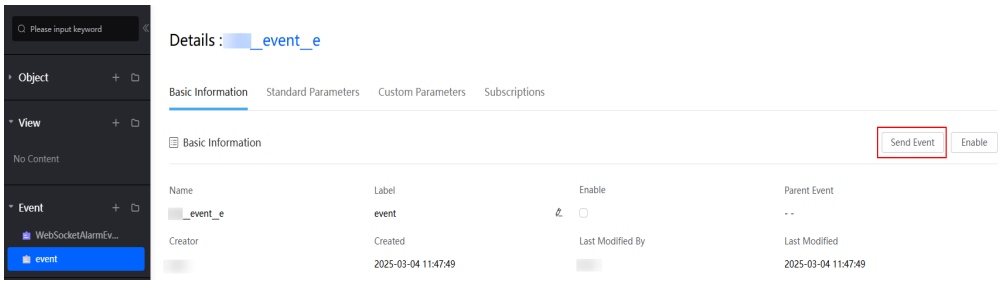
4. Set tenant-level complete logs and click to enable log tracing.

Figure 3-8 Enabling log tracing



5. On the event list page shown in **Figure 3-9**, test the sending of the event.

Figure 3-9 Testing event sending



6. View logs to check whether the fault is caused by incorrect parameters or abnormal process execution. If yes, rectify the fault and try again.

Step 5 If the flow still cannot be triggered after the event is sent, the Kafka may be abnormal or messages are stacked. In this case, contact O&M personnel.

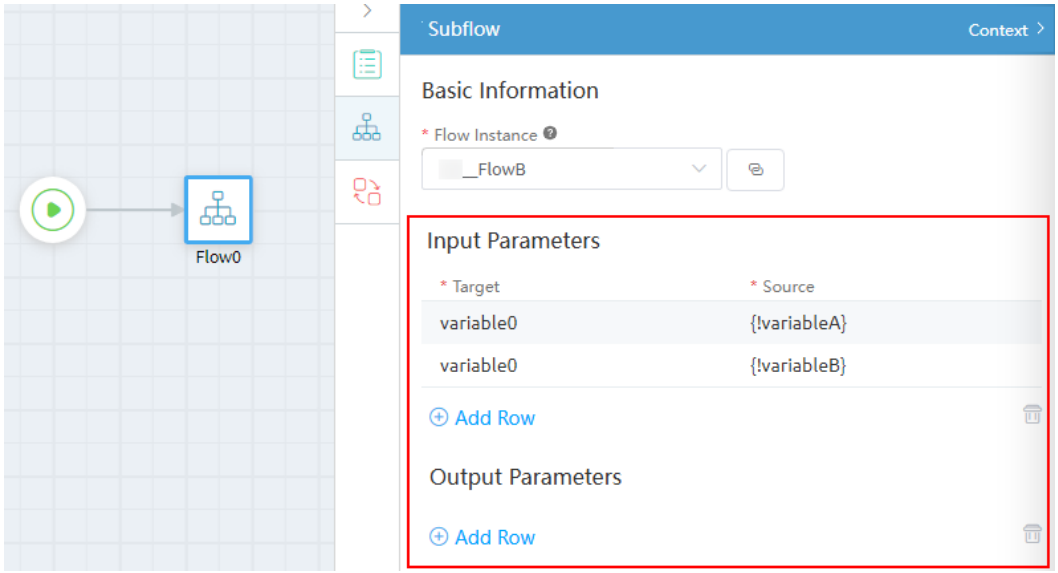
----End

3.11 Why Cannot Referenced Resources Be Edited in Flows or BPMs?

Description

When a lock is obtained, the referenced resources in the flow and BPM cannot be edited. As shown in the following figure, ***Namespace*__FlowA** calls ***Namespace*__FlowB** of another service. When ***Namespace*__FlowA** is locked, the input and output parameters of ***Namespace*__FlowB** cannot be edited.

Figure 3-10 Failed editing in the flow



Solution

In the latest version, flows and BPMs call only resources (such as scripts, flows, and objects) of the **current application**. Resources cannot be called across applications.

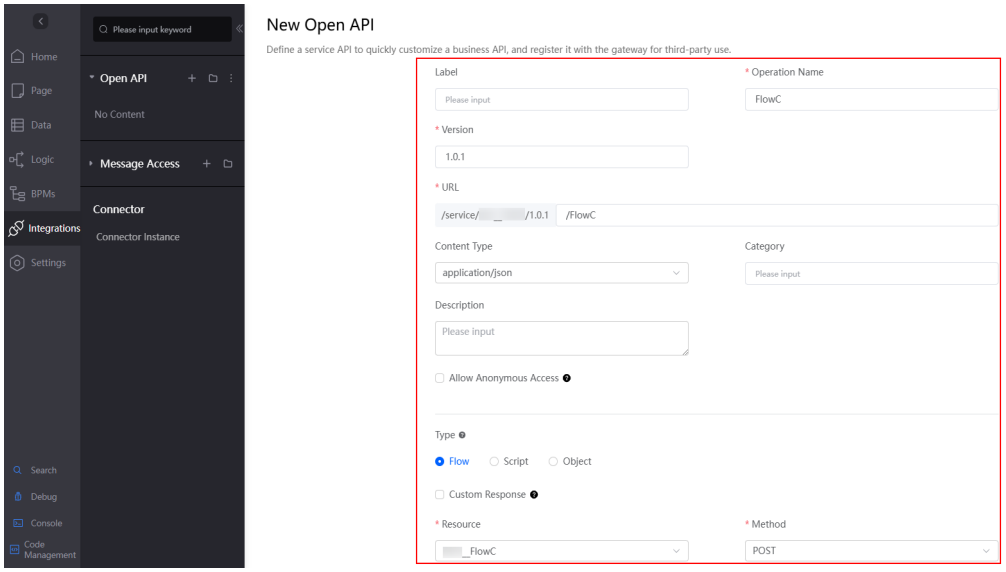
If the flow and BPM created in the earlier version call resources of different applications, **the runtime state is not affected**. The called resources cannot be edited in the flow and BPM in the development environment.

In the example in **Description**, if you need to edit the input and output parameters of ***Namespace*__FlowB**, you are advised to create a public resource in the BO, import the BO to the internal or external dependencies of these applications, and then call the resource.

Step 1 Create a BO and encapsulate the resources to be called into public APIs in the BO.

1. Create a label and a BO named **BOTest** by referring to **Creating a Business Object**.
2. In the BO, create the ***Namespace*__FlowC** to be called. The definition of ***Namespace*__FlowC** is the same as that of ***Namespace*__FlowB** to be called.
3. Create an open API in the BO by referring to **Figure 3-11**. The API is used to encapsulate ***Namespace*__FlowC**.

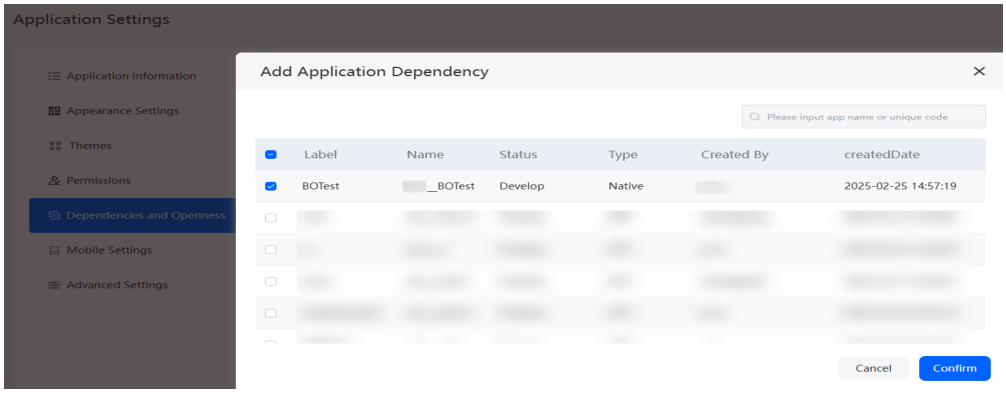
Figure 3-11 Creating an API to encapsulate Namespace__FlowC



Step 2 Go to the application to which the flow belongs, import the BO, and use BO diagram elements to call APIs in the flow to call resources.

1. Go to the designer of the application where ***Namespace*__FlowA** is located.
2. Click the settings button in the navigation pane of the application designer.
3. On the displayed page, choose **Dependencies and Openness** > **App Dependencies** > **Add**.
4. Select the BO to be added and click **OK**. After the BO is added, it can be called in the flow of the application.

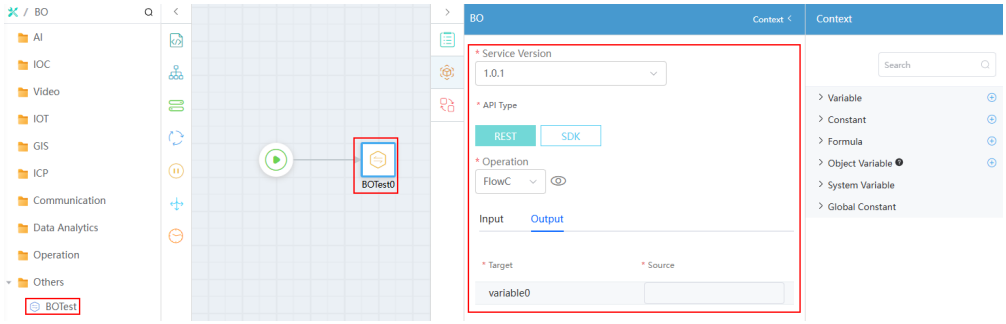
Figure 3-12 Selecting the BO to be added



5. Use the BO diagram element instead for ***Namespace*__FlowA** and call its public API. Set the API type to **REST**. For details, see **Figure 3-13**.

   On the left of ***Namespace*__FlowA** editor page, select **BO** and drag **BOTest** defined in **Step 1.1** from **Others** to the canvas. Select the BO diagram element in the canvas and click . On the BO page displayed, configure the BO diagram element and call its API.

Figure 3-13 Calling a BO



----End

3.12 How Can a Resource Be Orchestrated and Called by Multiple Applications?

Handling Method

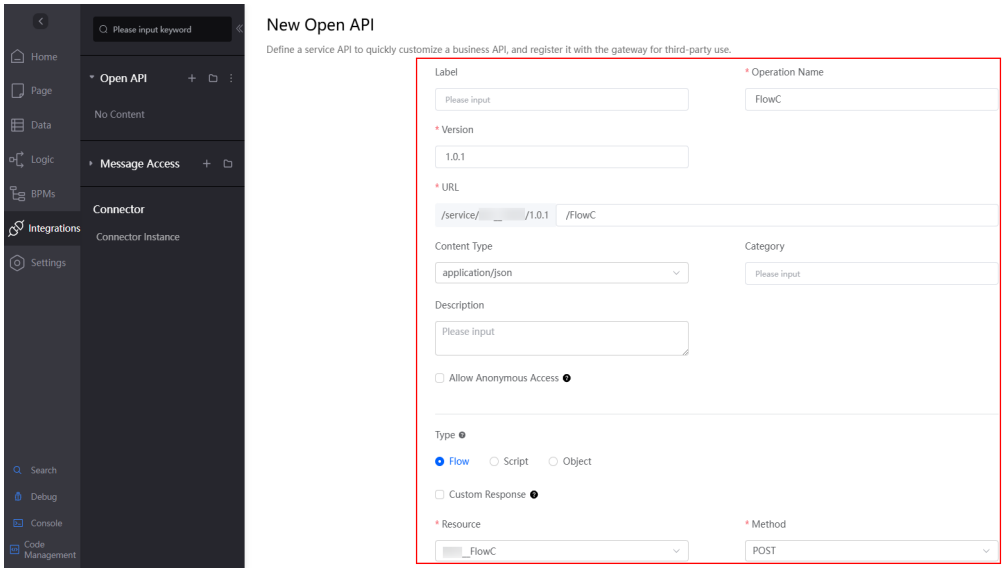
In a BO, create and develop the resource, encapsulate the resource into a public API, import the BO to multiple applications, and call the resource.

Procedure

Step 1 Create a BO and encapsulate the resources to be called into public APIs in the BO.

1. Create a label and a BO named **BOTest** by referring to **Creating a Business Object**.
2. In the BO, create ***Namespace*__FlowC** to be called and enable it.
3. Create an open API in the BO by referring to **Figure 3-14**. The API is used to encapsulate ***Namespace*__FlowC**.

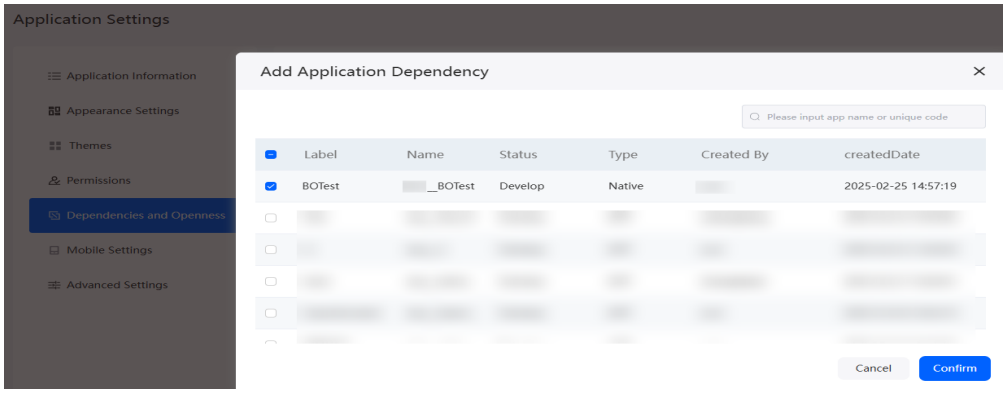
Figure 3-14 Creating an API to encapsulate Namespace__FlowC



Step 2 Go to the application to which the flow belongs, import the BO, and use BO diagram elements to call APIs in the flow to call resources.

1. Go to the designer of the application where ***Namespace*__FlowA** is located.
2. Click the settings button in the navigation pane of the application designer.
3. On the displayed page, choose **Dependencies and Openness** > **App Dependencies** > **Add**.
4. Select the BO to be added and click **OK**. After the BO is added, it can be called in the flow of the application.

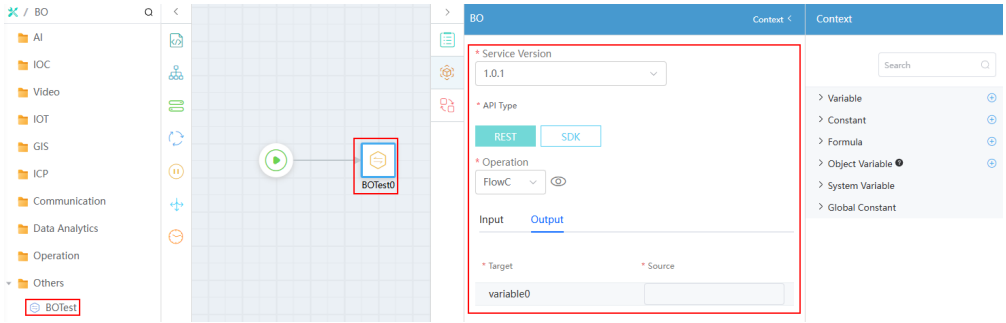
Figure 3-15 Selecting the BO to be added



5. Use the BO diagram element instead for ***Namespace*__FlowA** and call its public API. Set the API type to **REST**. For details, see **Figure 3-16**.

   On the left of ***Namespace*__FlowA** editor page, select **BO** and drag **BOTest** defined in **Step 1.1** from **Others** to the canvas. Select the BO diagram element in the canvas and click . On the BO page displayed, configure the BO diagram element and call its API.

Figure 3-16 Calling a BO



----End

3.13 Can Resources Be Referenced Across Applications in Flows?

Flows do not support cross-application resource reference.

3.14 How Do Huawei Cloud Astro Zero Developers Use Their Own Email Addresses to Send Emails?

Description

When sending an email using the **Send Email** diagram element in a flow or the **sendEmail** method in a script, the default sender is the email address configured on the platform. You can perform the following operations to use your own email address to send emails.

Solution

Try either of the following:

- Use the sendCustomEmail method provided in the script.

  Method: sendCustomEmail(userName: string, password: string, sender: string,template: string, input: Map, smtpHost string)

  - userName and password: username and password for logging in to the mailbox
  - sender: email address of the sender
  - template: (optional) name of the email template used to send emails
  - smtpHost: (optional) email server. The sender email address must match the email server.
  - input: email recipient (Address: a string. Separate multiple address by semicolons), email subject (Subject: a string), and email content (Body: a string)

    After the email-sending script is defined, you can call it in a flow to send emails using the tenant's own email address. The **Send Email** diagram element is no longer needed.
- Configure a connector to connect to the SMTP, and call the connector in a flow or script to send emails. For details, see **Interconnecting with SMTP to Send Emails**.

3.15 How Do I Parse CSV/Excel Files in an OBS Bucket?

Description

How can I parse data from CSV or Excel files stored in an OBS bucket while developing applications in Huawei Cloud Astro Zero? For example, if an Excel file with a supplier list is uploaded to the OBS bucket, how do I extract and use that data in Huawei Cloud Astro Zero?

Figure 3-17 Supplier list

Corporate nam	Company type	Date of Incc	Procureme	Assessmer	Supplier level
HUBER SUHNER	Sole foreign inves	1964-01-01	Mr. He	In assess	gold medal
intel	listed company	1968-07-18	Mr. Guo	Assessmer	gold medal
NXP	Limited by shares	2006-08-22	Ms. Mao	In assess	gold medal
Xilinx	listed company	1984-01-01	Mr. Li	In assess	gold medal
Marvell	Limited by shares	1995-01-01	Mr. He	In assess	gold medal
SYE	Limited by shares	1995-03-21	Mr. We	In assess	gold medal
ZHONGLI GROUP	Limited by shares	1975-09-15	Mr. He	Assessmer	excellent
Fujitsu	Limited by shares	1995-01-01	Mr. Fu	In assess	excellent
WUS Printed C	Limited by shares	1993-06-14	Mr. Wu	In assess	gold medal
Micron	Limited by shares	1994-03-22	Mr. Liu	In assess	optimum

Solution

Step 1 Connect the OBS bucket to Huawei Cloud Astro Zero by referring to **Interconnecting with OBS Instances**.

Step 2 Use a script to obtain and parse an Excel file. That is, call the OBS connector to obtain the Excel file and call the Excel API to parse the Excel file content.

1. In the navigation pane, choose **Logic** and click **+** next to **Script**.
2. Enter a script name (for example, **ScriptTest**) to create a blank script.
3. On the script editor, paste the following code to the code editing area.

```
import * as excel from 'excel';
import * as objectstorage from 'objectstorage';
let ObsCli = objectstorage.newClient(objectstorage.StoreType.OBS, "Namespace__obs", "test"); // Connector name+Bucket name
let data = ObsCli.getObject("Supplier list.xlsx");//File name
let x = excel.decodeAll(data);
console.log(x);
```

   ***Namespace*__obs**, **test**, and **Supplier list.xlsx** are the connector name, OBS bucket name, and Excel file name to be parsed in the bucket, respectively, which are set when you create an OBS connector in **Step 1**.

4. Click  in the upper part of the page to save the script.
5. After the script is saved, click  on the top of the editor page to execute the script.
6. Click  in the upper right corner of the test window at the bottom of the page to return the message headers.

   On the log tab page, you can view that the content has been parsed.

```
0814 09:25:43.931|debug|vm[1]>>> Build #AppCube Core 25.7.0 amd64
Built on 2025-08-07 19:44:28
Commit #d7cfcf8bbe
0814 09:25:43.933|debug|vm[1]>>> node: 11
0814 09:25:43.934|debug|vm[1]>>> js version: 2.0
0814 09:25:43.935|debug|vm[1]>>> log level: debug
0814 09:25:43.936|debug|vm[1]>>> script: Namespace__ScriptTest__1.0.1
0814 09:25:43.936|debug|vm[1]>>> locale: en_US
0814 09:25:43.937|debug|vm[1]>>> os timezone: Local
0814 09:25:43.938|debug|vm[1]>>> user timezone: Local
0814 09:25:43.939|debug|vm[1]>>> organization timezone: (GMT+08:00) China Standard Time (Asia/Shanghai)
```



```
0814 09:25:44.158|debug|vm[1]>>>
 {
   Supplier list: [[["Company name", "Company type", "Date of Incorporation", "Procurement
Specialist", "Assessment status", "Supplier level"], ["Company A", "Foreign-funded", "1964-01-01",
"Mr. He", "Assessing", "gold medal"], ["Company B", "Listed", "1968-07-18", "Mr. Guo", "Assessment
completed", "gold medal"], ["Company C", "Joint-stock", "2006-08-22", "Ms. Mao", "Assessment
started", "gold medal"], ["Company D", "Listed", "1984-01-01", "Mr. Li", "Assessing", "gold medal"],
["Company E", "Joint-stock", "1995-01-01", "Ms. Zhao", "Assessment completed", "gold medal"],
["Company F", "Joint-stock", "1995-03-21", "Ms. Zhao", "Assessment completed", "gold medal"],
["Company G", "Joint-stock", "1993-03-01", "Ms. Wang", "Assessment completed", "excellent"],
["Company H", "Taiwan-funded private", "1974-02-20", "Mr. Wu", "Assessment started", "gold
medal"], ["Company I", "Joint-stock", "1995-04-03", "Ms. Yang", "Assessment completed", "gold
medal"], ["Company G", "Listed", "1996-01-01", "Ms. Li", "Assessing", "Best"]],
   Import rule description and example: [["Single-choice user," "Date and time," "Date," "Multiple-
choice," "Single-choice," "Phone number," "URL," "Email address," "Percentage," "Amount," "Value,"
"Multi-line text," "Single-line text," "Form field type"], ["a00123456", "2000-01-01 12:00:00",
"2000-01-01", "Option 1, Option 2, Option 3," "Option 1", "1333**333", "https://example.com",
"example@***.com", "50%", "100", "100.123456789", "Multi-line text,multi-line text,multi-line text
example," "Single-line text example," "Format specification example"], ["Enter the W3 account (with
letters)," "," "Use different values in this column as multiple options. Use commas (,) to separate
different options.", "Use different values in this column as a single option.", "Enter the mobile phone
number of the Chinese mainland.", "Enter the standard URL format.", "Enter the standard email
address format.", "Enter the percentage symbol.", "", "", "", "Format description"]]
 }  (AppTestSpace__test1:8)
0814 09:25:44.159|debug|vm[1]>>> vm memory usage --  malloc limit: 67108864, peek allocated:
284224
NAME                                COUNT            SIZE
memory allocated                     1998          175264  (87.7 per block)
memory used                          2103          139182  (8 overhead, 17.2 average slack)
atoms                                862            32512  (37.7 per atom)
strings                              4              79  (19.8 per string)
objects                              350            25200  (72.0 per object)
  properties                         1538            26912  (4.4 per object)
  shapes                             78             18152  (232.7 per shape)
bytecode functions                     129            23054
  bytecode                           129             6321  (49.0 per function)
  pc2line                            116             504  (4.3 per function)
C functions                          151
arrays                               3
  fast arrays                        3
  elements                           0               0  (0.0 per fast array)
binary objects                       4             16564
0814 09:25:44.160|debug|vm[1]>>> sys resource usage
NAME                                LIMIT            SIZE
RequestSOQLQueriesAmount                100               0
RequestSOQLRowsAmount                   100000              0
RequestSOSLQueriesAmount                  20              0
RequestSOSLRowsAmount                   50000               0
RequestDMLStatementsAmount                150               0
RequestDMLRowsAmount                    10000               0
RequestCallOutsAmount                   10              0
RequestEmailSendsAmount                   10              0
RequestEventSendsAmount                   50              0
0814 09:25:44.161|info|vm[1]>>> release vm[1], last file: Namespace__ScriptTest, elapse time
230.752169ms, load module time 96.149µs, read file time 1.65µs
load module metrics:
module name        read file time      run code time       sum load time       tree child modules
-----------------  -----------------   -----------------   -----------------   -----------------
excel           910ns           39.769µs        40.679µs        []
objectstorage       740ns           54.73µs         55.47µs         []
```

7. After the script test is complete, click  in the upper part of the editor to activate the script.

----End

4 Application Frontend Development

4.1 How Do I Call a Flow or Script on a Standard Page?

Method 1: Editing an Event

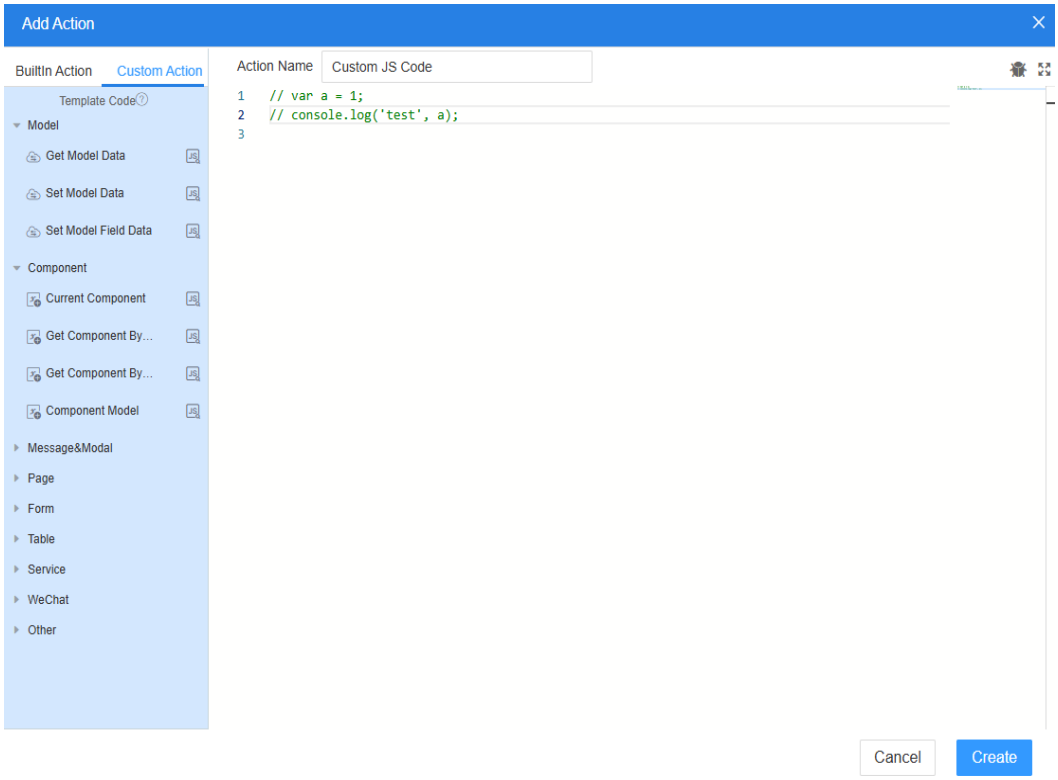
During event orchestration on a standard page, write code in the **Custom JS Code** area to implement the interaction between page widgets and backend APIs (flows or scripts).

The following describes how to edit an event and define the JavaScript code.

Step 1 On the application design page, select a widget, click **Events** in the right pane, and click **+** next to an event.

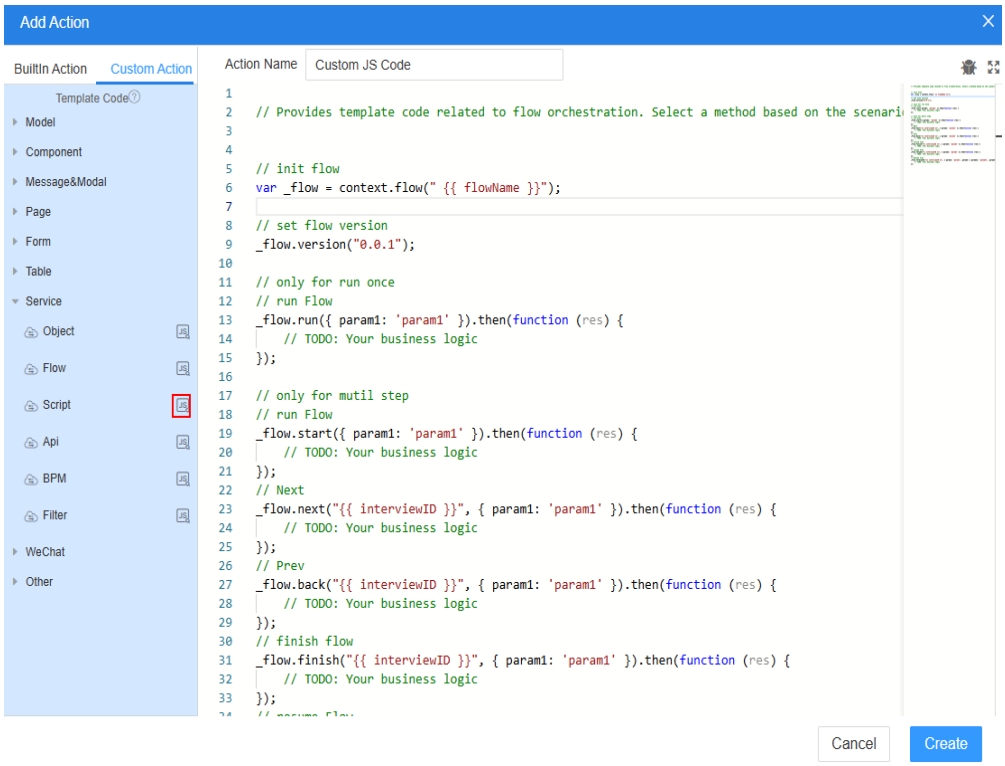
Step 2 In the **Add Action** area, set **Builtin Action** or **Custom Action**. For a custom action, customize the JavaScript code logic. You can click ⛶ on the right of the code area to maximize the JavaScript code page.

Figure 4-1 Custom JavaScript code



Step 3 In the template code on the left, choose **Service** > **Flow** or click , and copy the code to the code editing area. The preset API is displayed.

Figure 4-2 Preset API code



----End

Method 2: Defining a Service Model

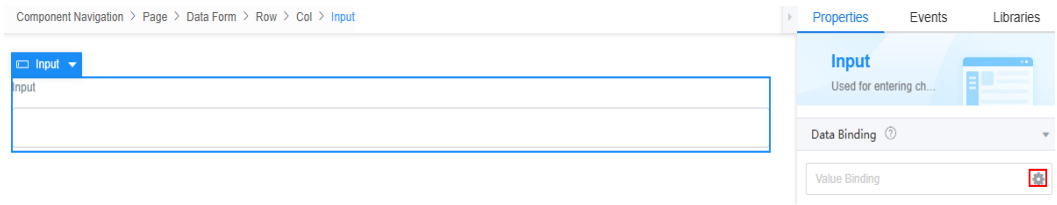
Define a data model of the service model type. The data model is created based on the input and output parameters of the backend services, and the **inputParam** and **outputParam** nodes are generated.

The following describes how to call a backend API (flow or script) for an input box widget by defining a data model of the service model type.

Step 1 Create a standard page and drag the input box widget to the page. For details, see **Creating a Blank Standard Page**.

Step 2 Select the input box widget, choose **Properties** > **Data Binding** on the right of the page, and click ⚙ to set the data source.

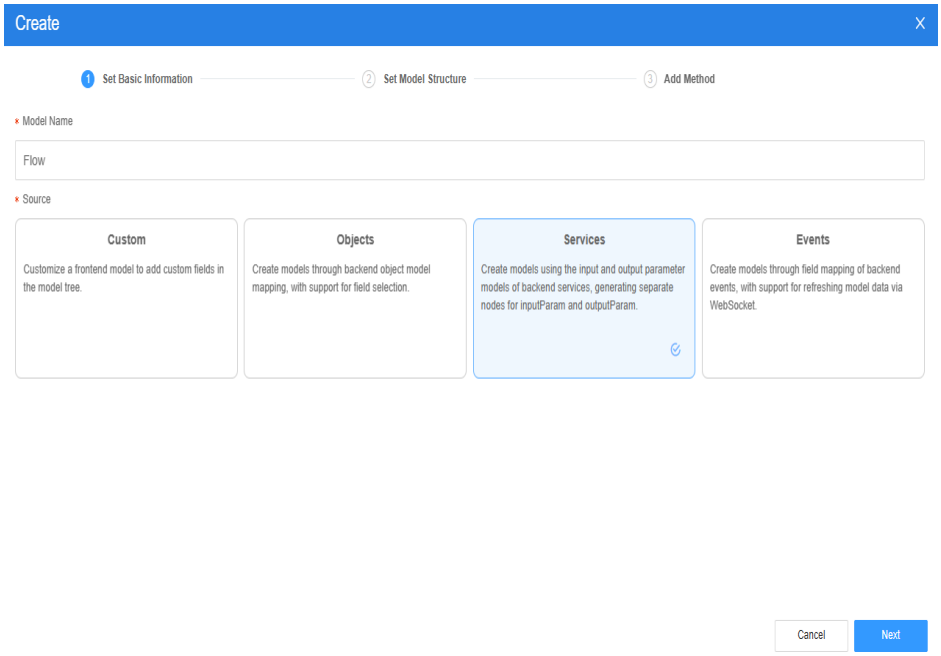
Figure 4-3 Binding a model



Step 3 On the **Select Model** page, click **New**.

Step 4 In the basic information area, set the model name, select **Services** for **Source**, and click **Next**.

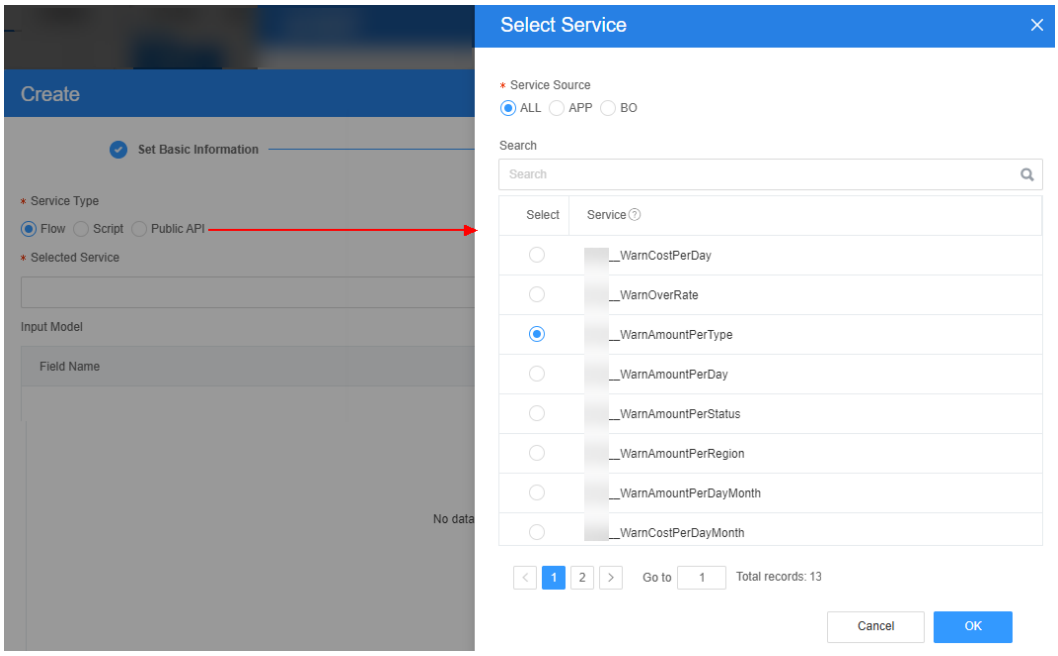
Figure 4-4 Selecting Services



Step 5 In the **Service Type** area, select **Flow** or **Script** to be called.

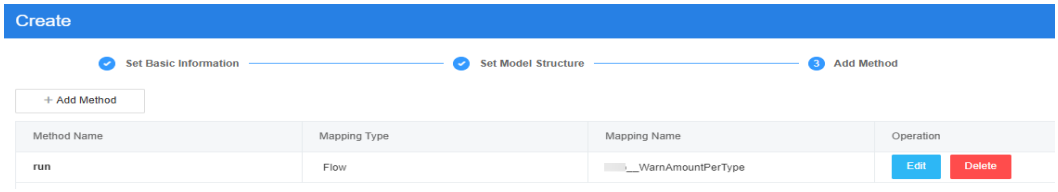
Step 6 On the displayed page, select a flow or script name and click **OK**.

Figure 4-5 Setting a specific service



Step 7 Click **Next**. The system generates the default **run** method. You can click **Add Method** to add other APIs to be called.

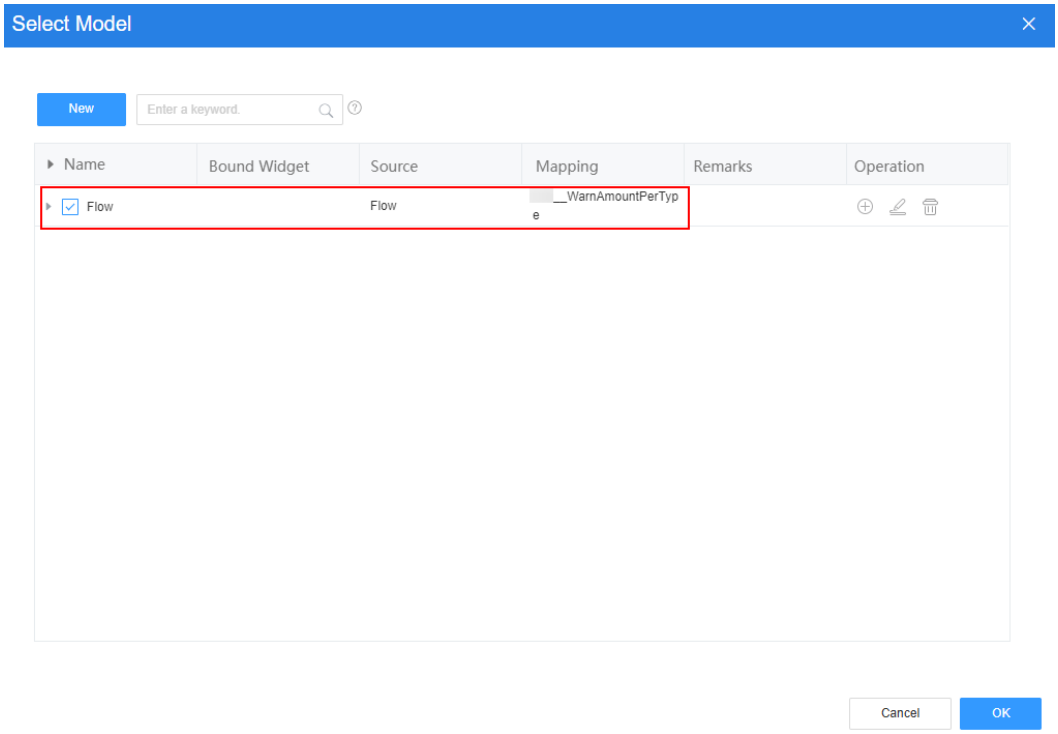
Figure 4-6 Checking the method



Step 8 After the setting is complete, click **OK** to return to the model selection page.

Step 9 On the **Select Model** page, select the created model and click **OK** to set the data source for the widget.

Figure 4-7 Setting a data source for the widget



----End

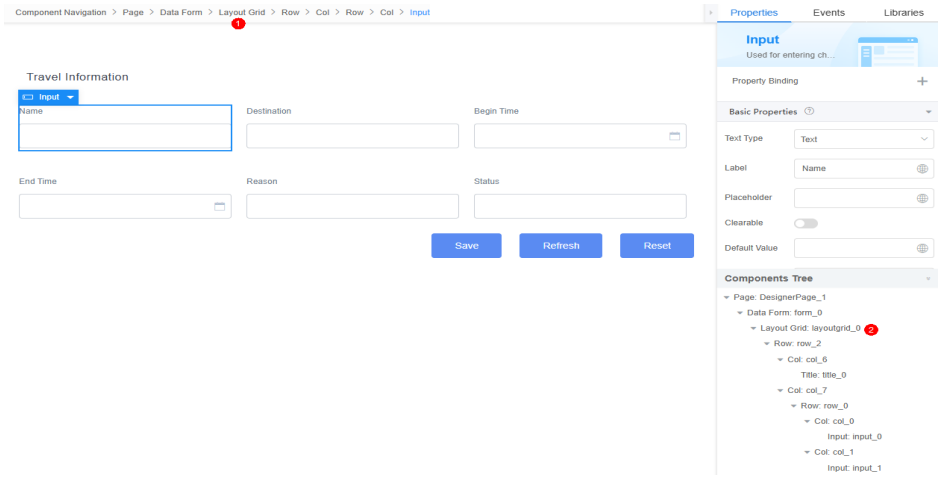
4.2 How Do I Quickly Select Widgets on a Standard Page with Many Widgets?

Method 1: Using the Widget Navigation

If you cannot quickly select a widget, select a subwidget near the target widget. After the subwidget is selected, the system displays its HTML label level in the widget navigation. Click a label level to quickly switch widgets.

As shown in **Figure 4-8**, select a subwidget in the grid container. Then, the widget level before the multi-line input box of the subwidget is displayed. On the widget navigation, click **Grid Container** to quickly select it.

Figure 4-8 Selecting the target widget



Method 2: Using the Widget Tree

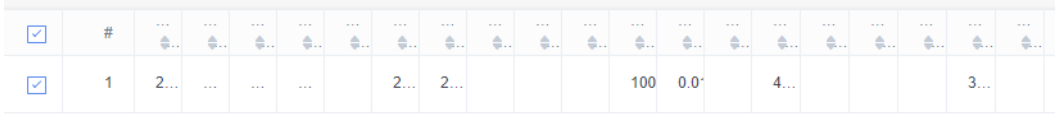
After selecting a widget, expand the widget tree and quickly select the target widget by name.

4.3 What Can I Do If Too Many Fields in a Table Cannot Be Completely Displayed on a Standard Page?

Description

A table cannot clearly display all fields, and there is no horizontal scroll bar.

Figure 4-9 Fields not displayed



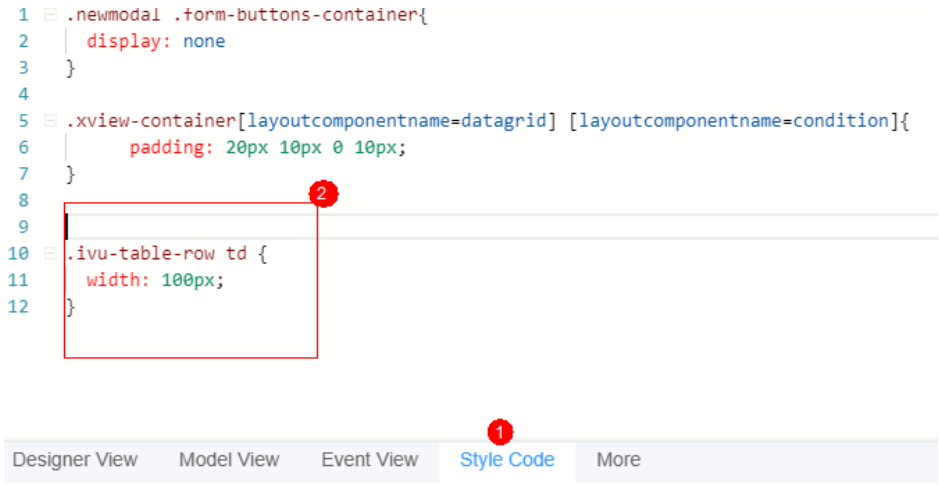
Solution

Add the style code on the **Style Code** area. Enter the following style code by referring to **Figure 4-10**.

```
.ivu-table-row td{
width:100px;
}
```

- If the **Style Code** area is empty, paste the style code and save the page.
- If there are many styles in the **Style Code** area, find the class of the table, add the preceding style code to the class, and change the column width as required. After that, a scroll bar is displayed during preview.

Figure 4-10 Adding the column width style to the style code area



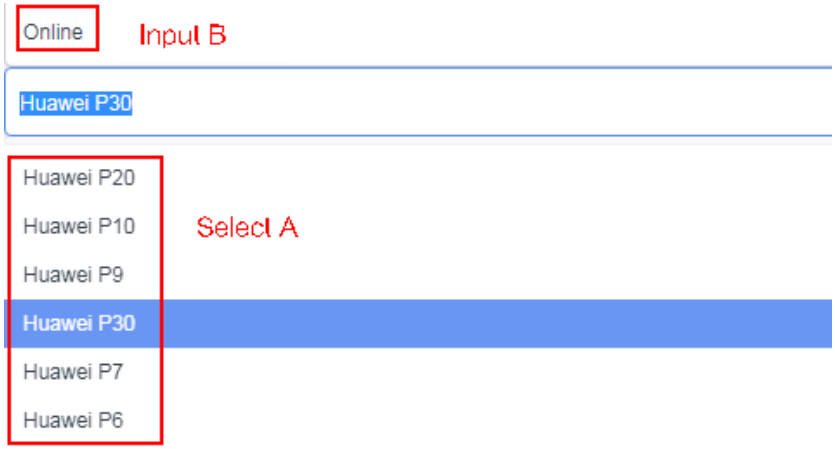
4.4 How Do I Set a Drop-Down List with Filtering Function on a Standard Page?

Description

How do I set a drop-down list to display filtered values based on the input field, instead of all values?

For example, an offering has three states: on-shelf, off-shelf, and test. Currently, the options in the drop-down list box **A** on the offering navigation bar are offerings in all states. As a result, a large amount of dirty data is generated and correct data cannot be selected. In this case, add an input box **B**. Then you can filter the options in the drop-down list box **A** based on the value entered in the input box **B**. Enter **Online** in the input box **B**. Only offering options in **Online** state are displayed in the drop-down list box **A**, as shown in **Figure 4-11**.

Figure 4-11 Implementation result

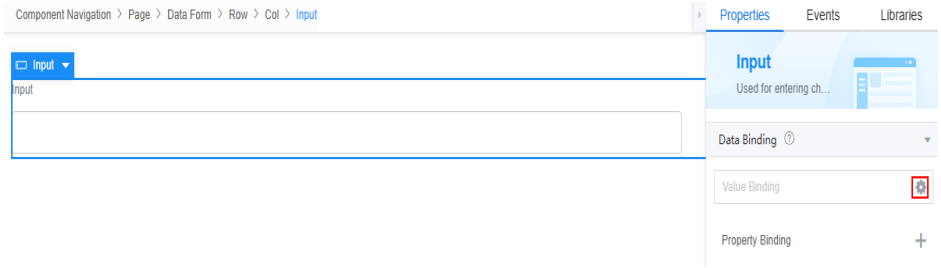


Solution

Set the input widget and drop-down list box widget on the standard page to customize the options in the drop-down list box.

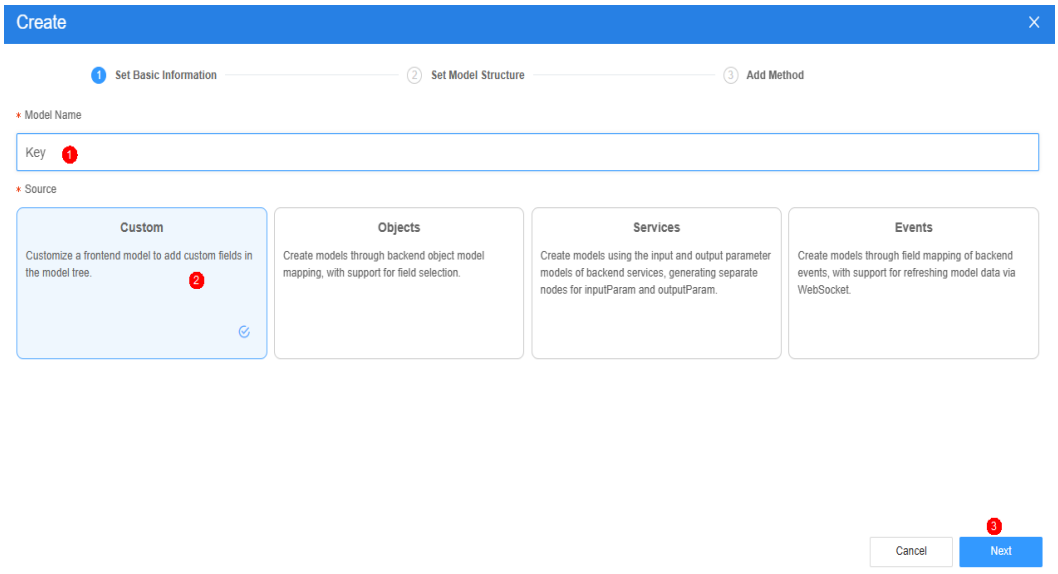
Step 1 Drag an input widget to the work area of the standard page, and click in the property setting area to bind the data.

Figure 4-12 Dragging the input box



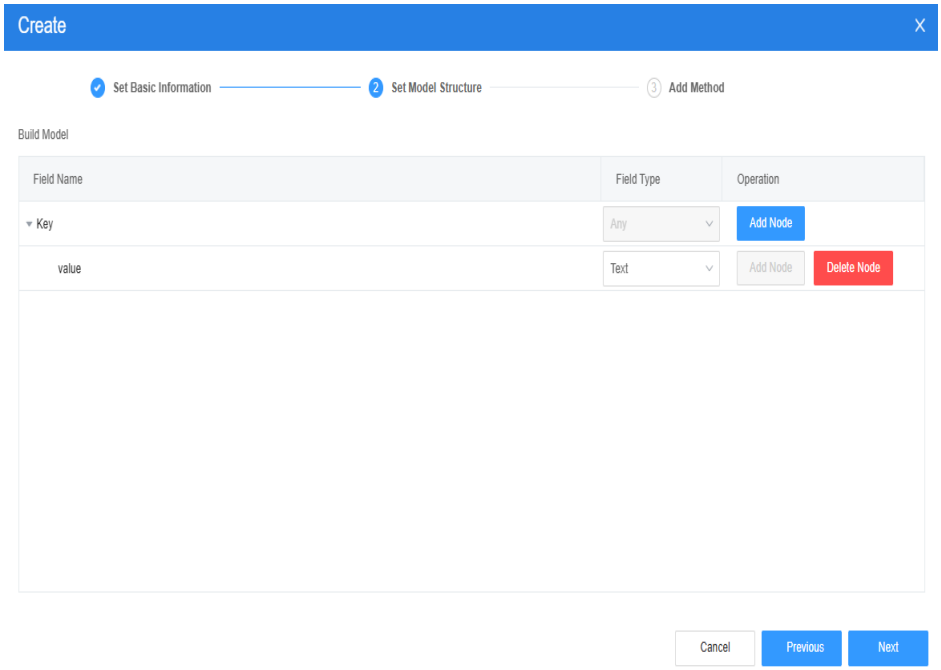
Step 2 Click **New**, set **Model Name** to **Key**, select **Custom**, and click **Next**.

Figure 4-13 Configuring a custom model



Step 3 Click **Add Node**, add the **Value** field, set the field type to **Text**, and click **Next**.

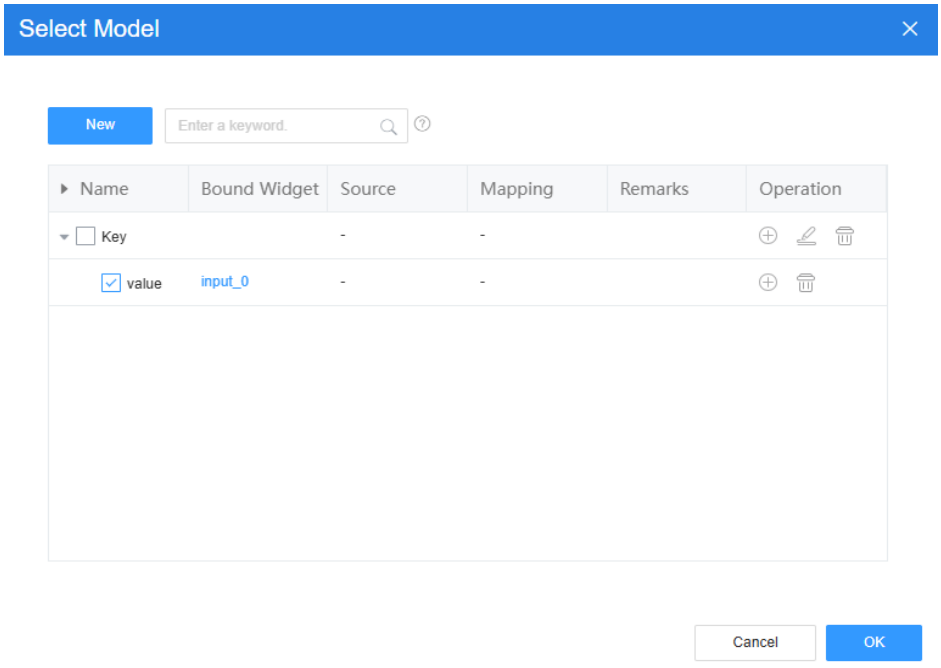
Figure 4-14 Adding a node



Step 4 In this example, you do not need to add a method. Click **OK**.

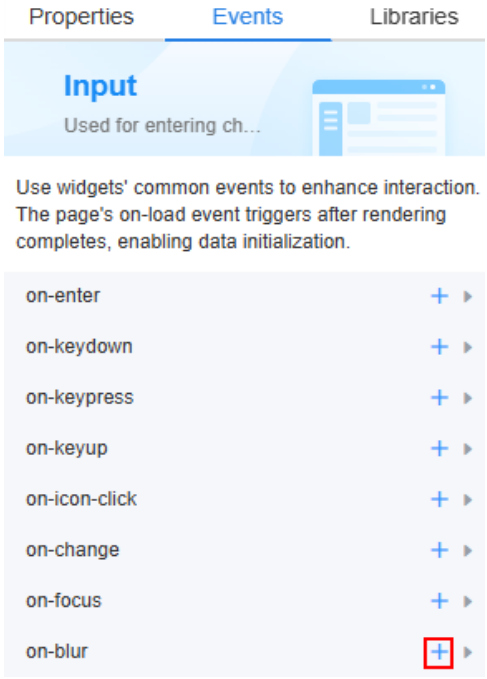
Step 5 In the **Select Model** dialog box, select the **Value** field and click **OK**.

Figure 4-15 Selecting the field



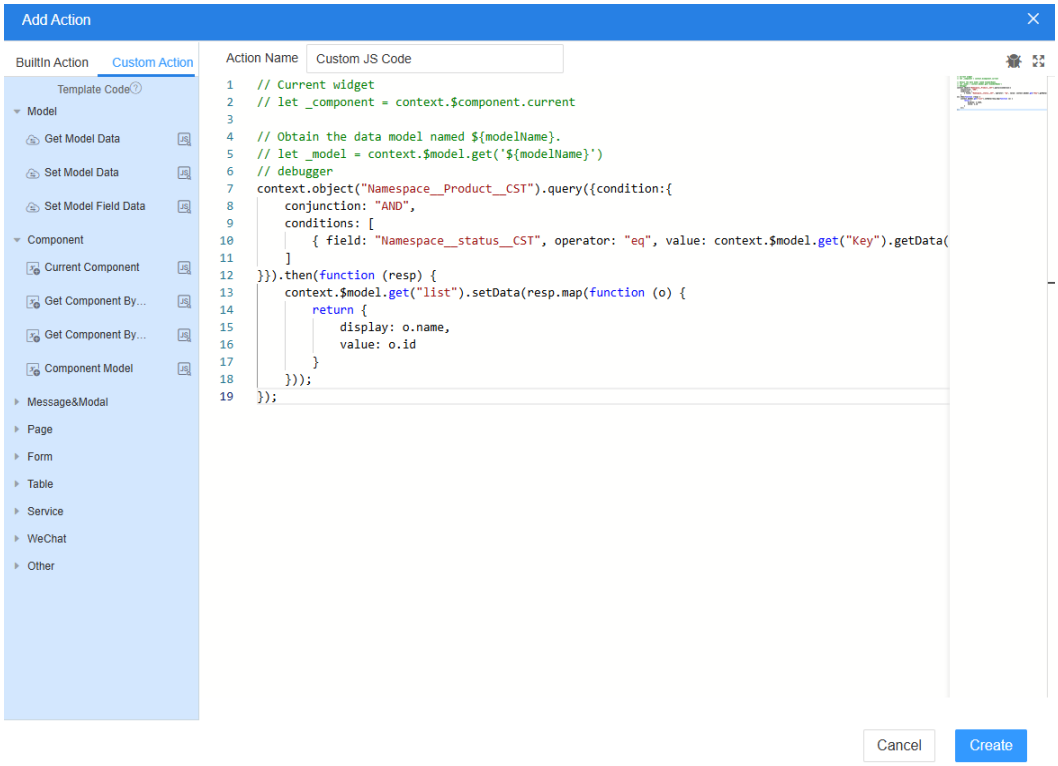
Step 6 Select the input widget. On the **Events** tab page, click + next to **on-blur**.

Figure 4-16 Setting an event



Step 7 In the code editing area, edit the event code and click **Create**.

Figure 4-17 Custom JavaScript code



The offering object *Namespace*__**Product**__**CST** exists in the system and contains the **name** and *Namespace*__**status**__**CST** fields, which indicate the offering name and offering state, respectively. This event code is used to obtain the name of the

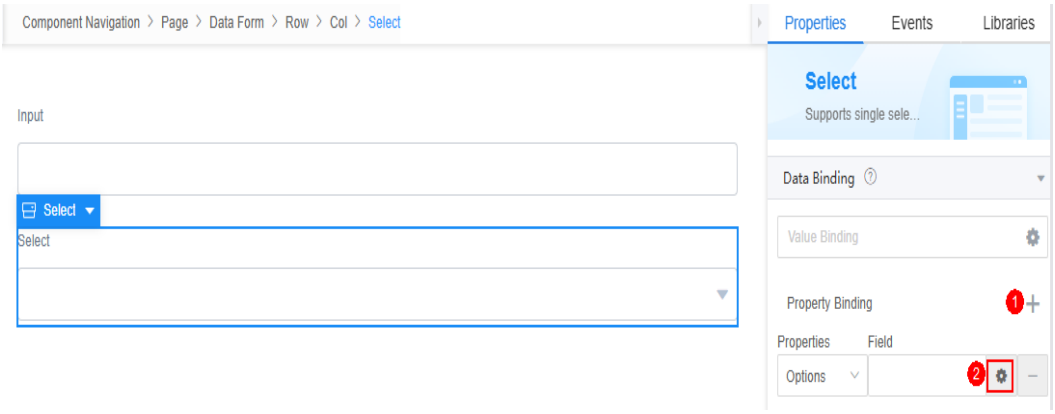
offering whose value in the input box is the same as the offering state and return the offering name to the list.

```
// Current widget
// let _component = context.$component.current

// Obtain the data model named ${modelName}.
// let _model = context.$model.get('${modelName}')
// debugger
context.object("Namespace__Product__CST").query({condition:{
    conjunction: "AND",
    conditions: [
        { field: "Namespace__status__CST", operator: "eq", value: context.$model.get("Key").getData().value }
    ]
}}).then(function (resp) {
    context.$model.get("list").setData(resp.map(function (o) {
        return {
            display: o.name,
            value: o.id
        }
    }));
});
```

Step 8 Click  in the upper part of the page to save the page.

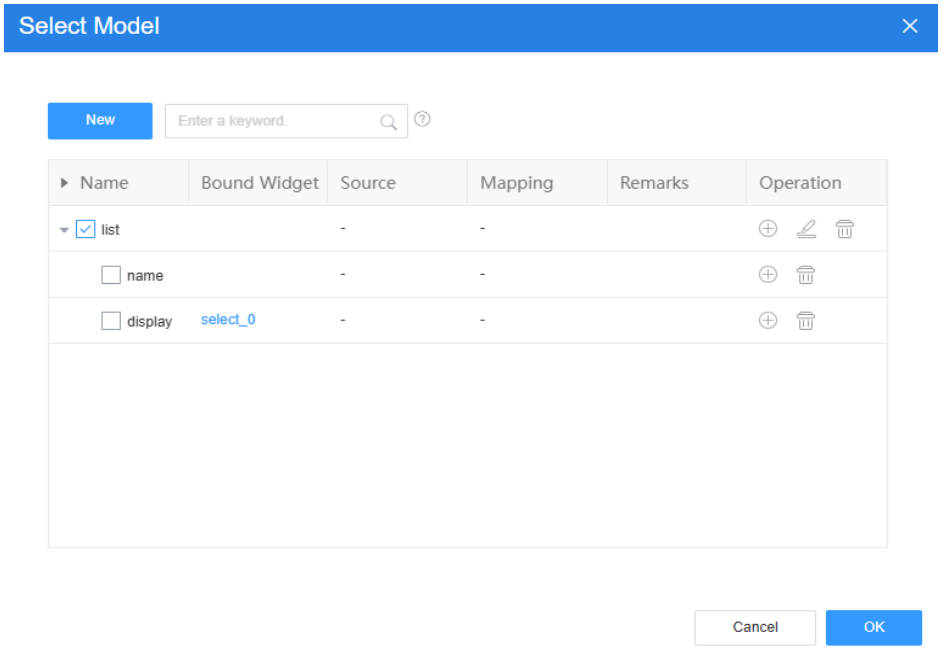
Step 9 Drag a drop-down list box widget to the working area of the standard page. On the **Properties** tab page, click **+** next to **Property Binding**, and then click ⚙.

Figure 4-18 Dragging a drop-down list box widget



Step 10 Refer to **2** to **6**, create a data model named **list**, add the **display** and **name** fields to the model, select **list**, and click **OK**.

Figure 4-19 Adding the list model



Step 11 Click  in the upper part of the page to save the page.

Step 12 Click  to preview the standard page.

----End

4.5 How Do I Implement the Asynchronous Pull-up Data Loading on Mobile Devices on a Standard Page?

On a standard page, add a list view widget to implement asynchronous pull-up data loading on mobile devices. The procedure is as follows:

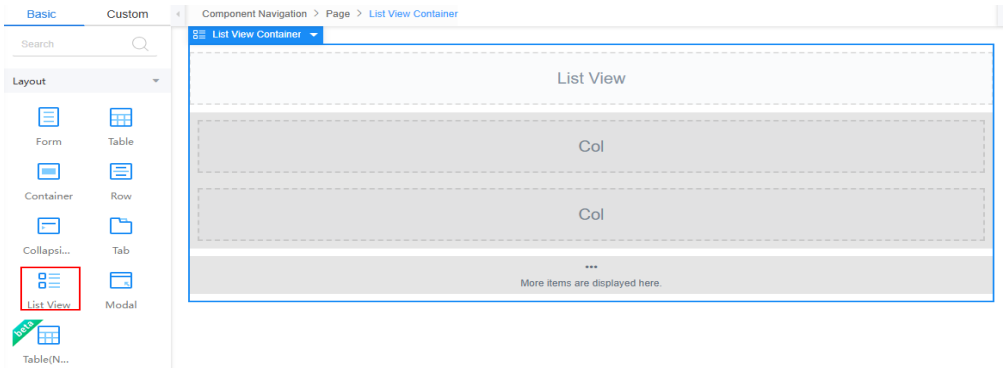
Step 1 On the Huawei Cloud Astro Zero console, click **Access Homepage**.

Step 2 On the **Homepage** > **All Apps** page, click **Edit** next to an application to access the application designer.

Step 3 Create a standard page and design it.

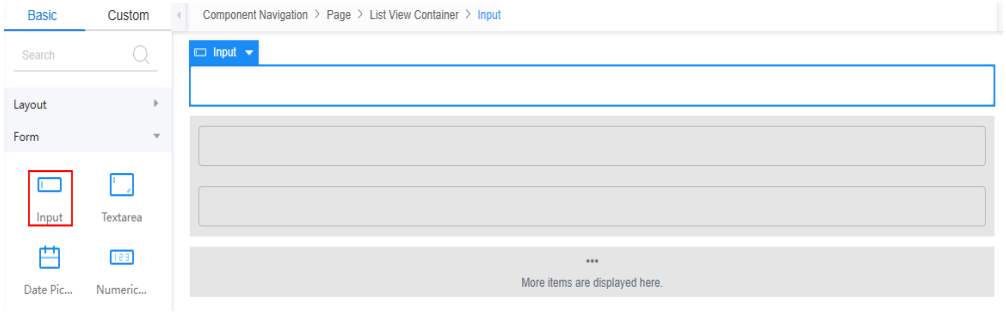
1. On the application designer page, click **+** next to **Page** and enter a page label and name to create a standard page.
2. Choose **Basic** > **Layout** and drag a list view widget to the canvas.

Figure 4-20 Adding a list view widget



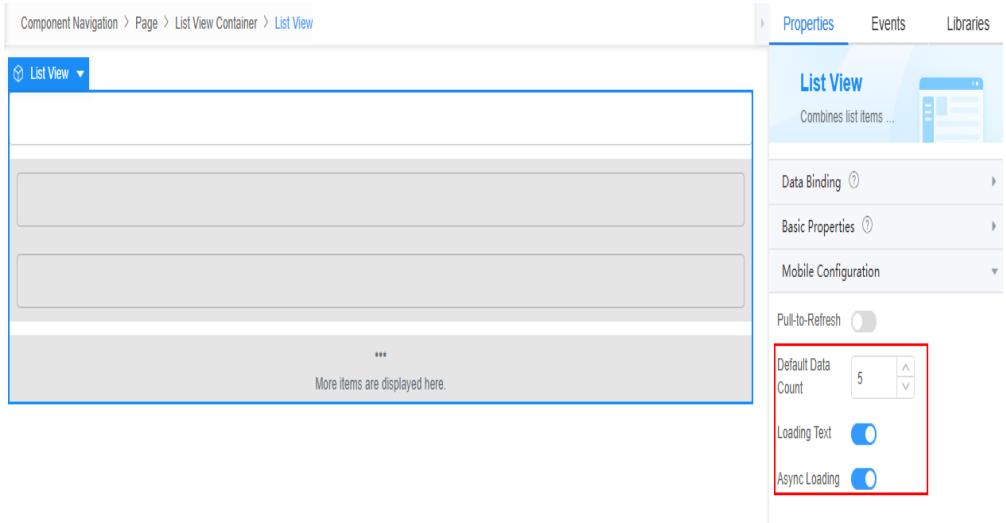
3. Drag an input widget from **Form** to the **List View** widget.

Figure 4-21 Adding an input widget



4. Select the list view widget, choose **Properties** > **Mobile Configuration**, set the value of **Default Data Count** to **5**, and enable **Loading Text** and **Async Loading**.

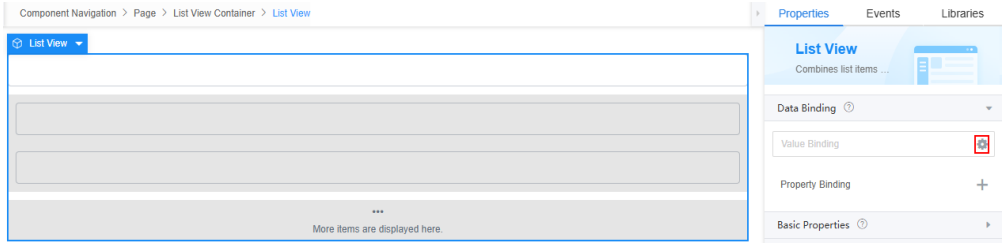
Figure 4-22 Mobile configuration



Step 4 Bind data to the widget.

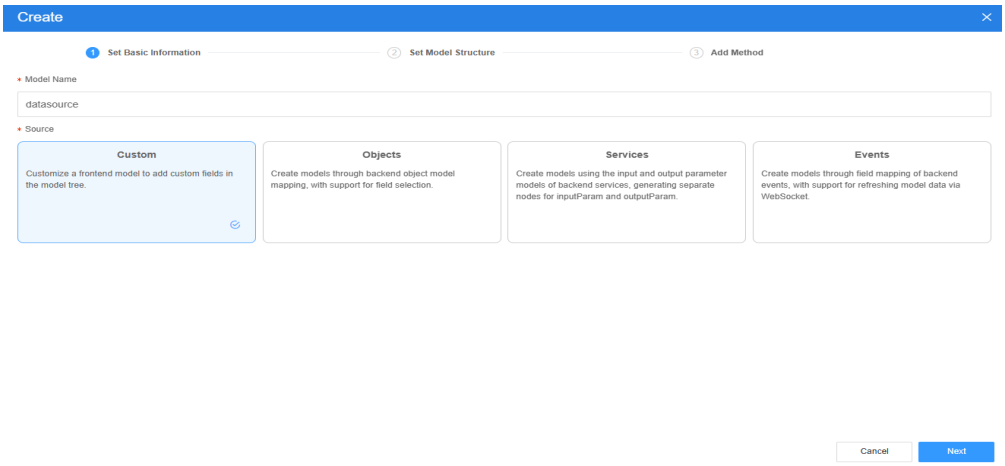
1. Select the list view widget, choose **Properties** > **Data Binding**, and click next to **Value Binding**.

Figure 4-23 Clicking the icon



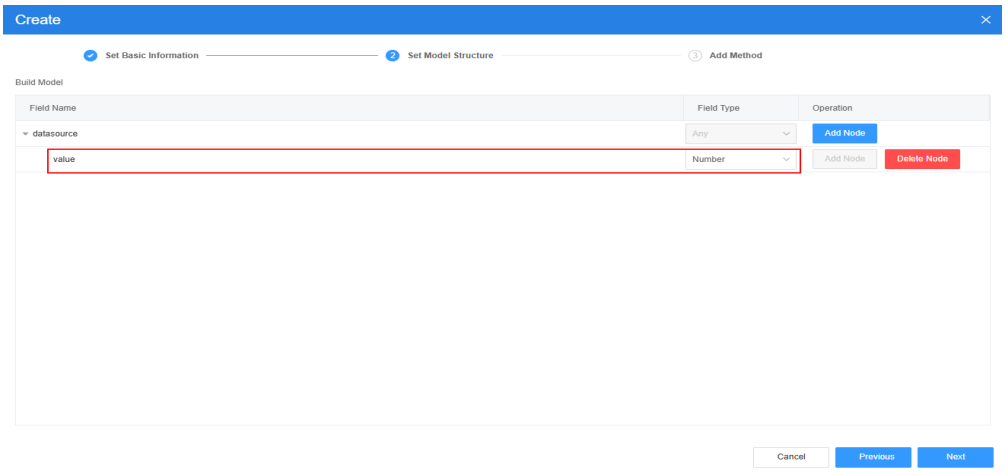
2. Click **New**, specify **Model Name** (for example, **datasource**), select **Custom** for **Source**, and click **Next**.

Figure 4-24 Adding a model



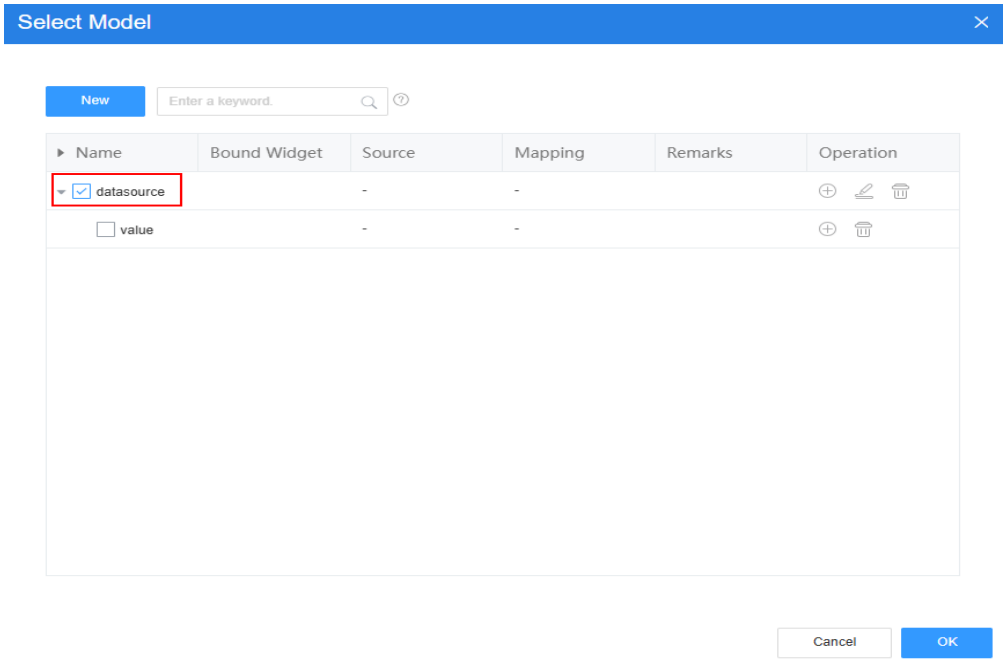
3. Click **Add Node** to add a value node, and click **Next**.

Figure 4-25 Adding a node



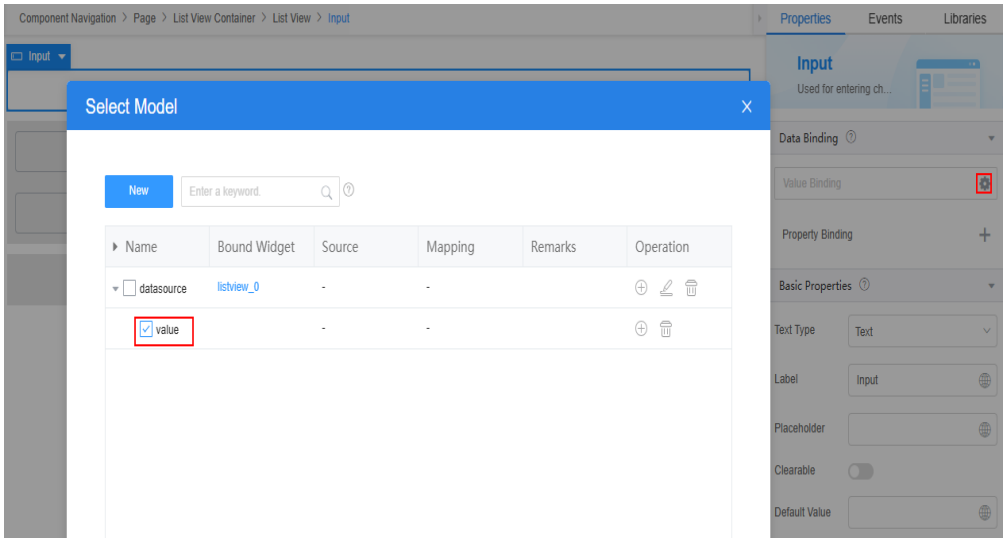
4. Click **OK**. On the model selection page that is displayed, select the created model.

Figure 4-26 Selecting the model



5. Select the input widget and perform the preceding operations to bind the **value** field to the widget.

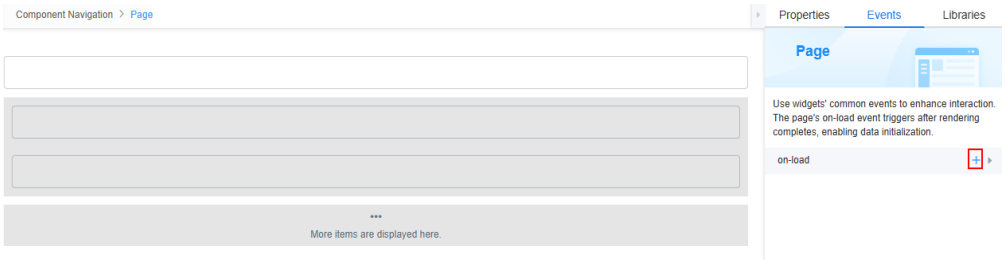
Figure 4-27 Binding a model to the input widget



Step 5 Add an event.

1. Select the page. On the **Events** tab page, click **+** next to **on-load**.

Figure 4-28 Adding an event



2. Under **Custom Action**, enter the following sample code and click **Create**.

```
const list = [];
for (let i = 0; i < 5; i++) {
    list.push({
        value: i + 1
    })
}
$model.ref('datasource').setData(list);
```

Set *datasource* to the model created in **Step 4.2**.

3. Select the list view widget and repeat the preceding operations to add events for it.

Figure 4-29 Adding events

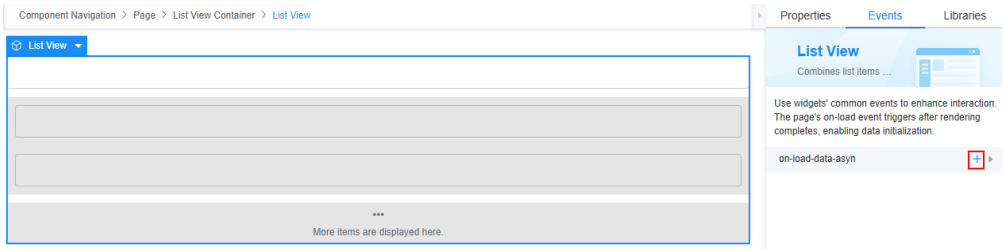
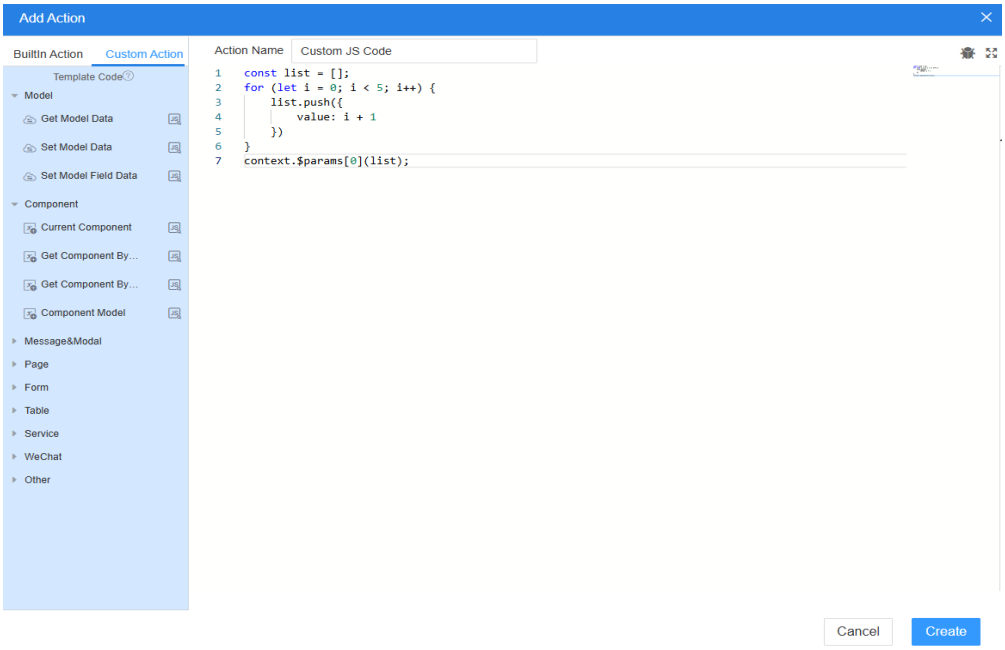


Figure 4-30 Entering the custom JavaScript code




```
const list = [];
for (let i = 0; i < 5; i++) {
```

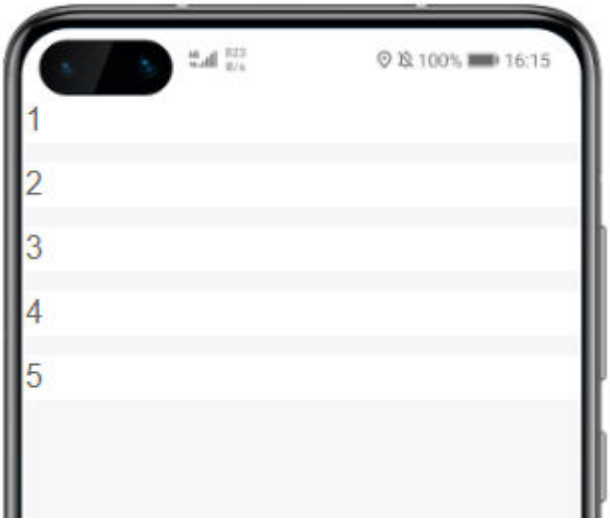


```
    list.push({
        value: i + 1
    })
}
context.$params[0](list);
```

Step 6 Check whether the effect meets the expectation.

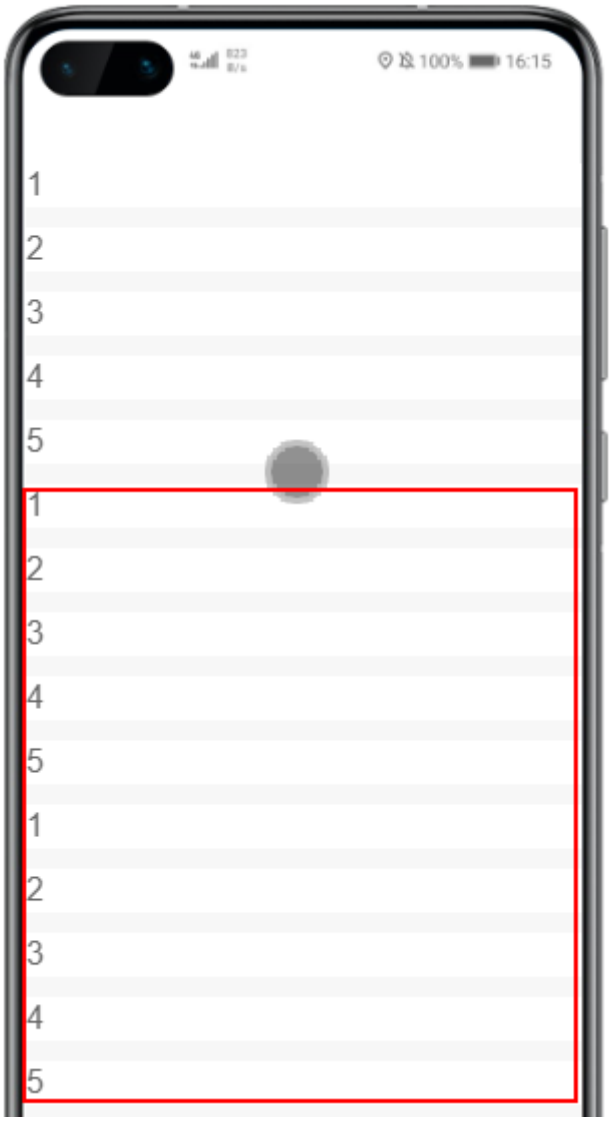
1. Click  in the upper part of the page to save the standard page.
2. Click  to preview the effect on the mobile device.

   Figure 4-31 Original display effect



3. Hold down the left mouse button and drag it up. The data can be loaded properly.

Figure 4-32 Asynchronous data loading



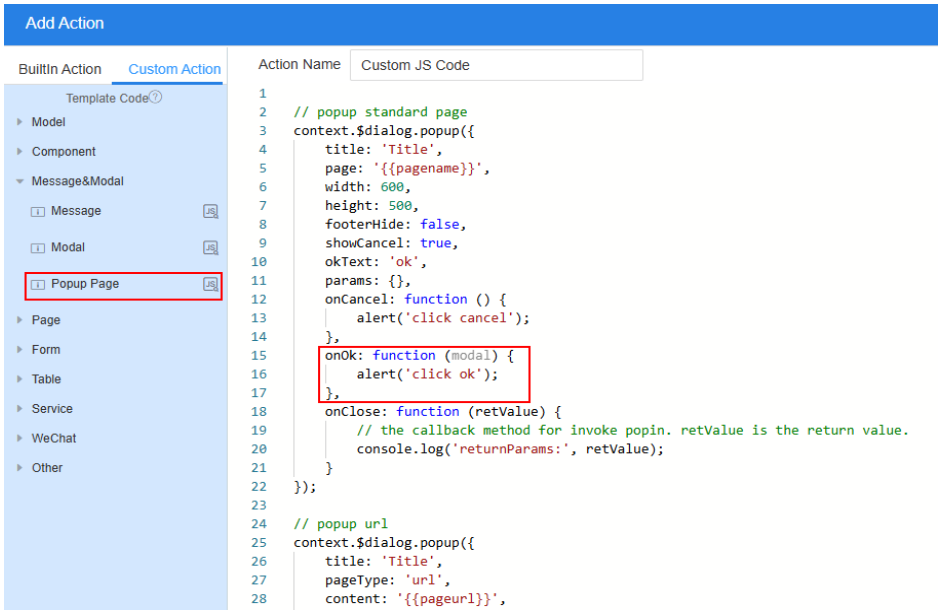
----End

4.6 What Should I Do If I Do Not Want to Close the Pop-up Window in the onOk Event on a Standard Page?

Description

I do not want to close the pop-up window in the **onOk** event on a standard page.

Figure 4-33 Customizing JavaScript code



Solution

The **onOK** event is preset in the system and cannot prevent the window from closing. To address similar needs, set **footerHide** to **true** to hide the default function buttons in the pop-up window. This will allow the pop-up to use the function buttons from the service page, where you can implement logic for validation, prompts, and closing behavior within the button events.

```
context.$dialog.popup({
  title: 'Title',
  page: '{{pagename}}',
  width: 600,
  height: 500,
  footerHide: true, //Modify the customized function buttons.
  showCancel: true,
  okText: 'ok',
  params: {},
});
```

4.7 What Should I Do If the Modified JavaScript File Is Not Working in an Advanced Page Custom Widget?

Description

After the JavaScript file of a custom widget is modified, the modification does not take effect.

Solution

After uploading a new version of a custom widget, upgrade the plugin via the page settings. The JavaScript file will only take effect after this upgrade.

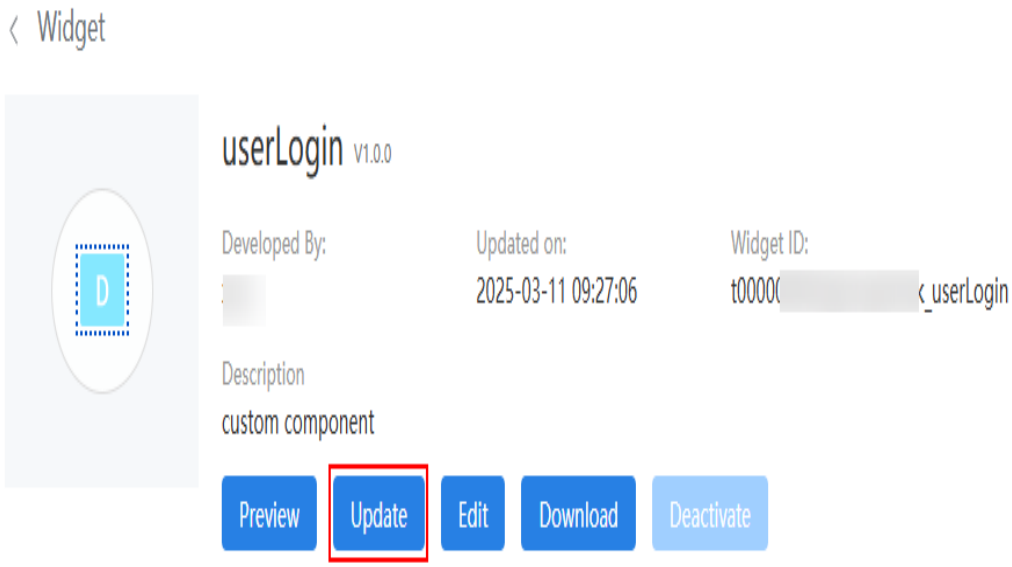
Step 1 Log in to the application designer by referring to **Logging In to the Application Designer**.

Step 2 In the navigation pane, choose **Page** and choose **More** > **Page Setting**.

Step 3 On the **Plugins** tab, find the widget to update and click on its right to open the widget details page.

Step 4 Click **Upgrade** to open the widget upgrade page.

Figure 4-34 Clicking the upgrade button



Step 5 Upload the local widget ZIP package and click **Upgrade**.

Step 6 Return to the plugin list page and click next to the uploaded widget to complete the upgrade.

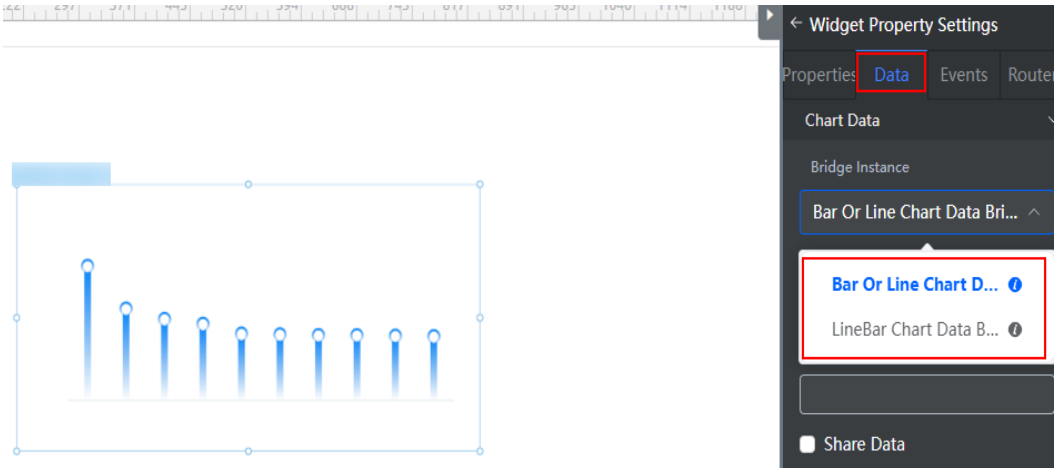
----End

4.8 How Do I Set Drop-Down List Options When Configuring a Bridge Instance of an Advanced Page Widget?

Description

When configuring bridge instance properties on a widget's **Data** tab, where are the drop-down list options sourced from?

Figure 4-35 Bridge instance



Solution

When the **model** value of the bridge property in the widget matches the value defined in the bridge, the bridge instance options become available.

For example, if the value of **model** in the **editor.js** file is **SingleRepireDataViewModel**, the same value must be specified in the **packageinfo.json** file for the line chart data bridge and bar chart and line chart data bridge.

Figure 4-36 Value of model in the widget

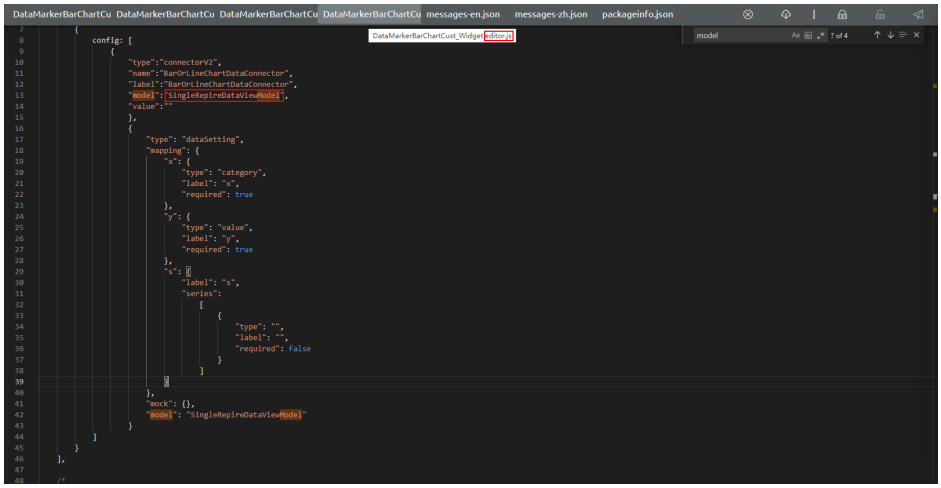
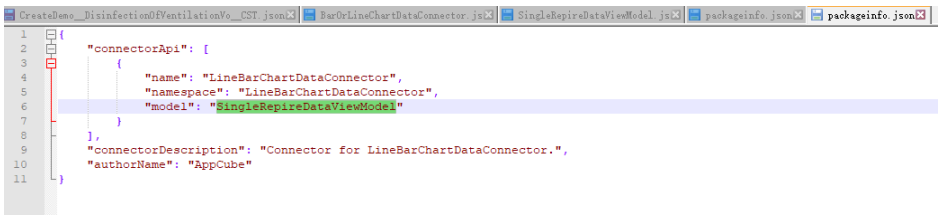


Figure 4-37 Value of model in the bridge package



4.9 How Do I Run a Logic Segment After All Widgets Have Loaded on an Advanced Page?

Description

During advanced page development, how can I run a logic segment after all widgets have loaded?

Solution

Bind the **readerPageWidgetReady** event. The sample code is as follows:

```
$(document).bind('readerPageWidgetReady', function(){
// Run the related logic.
});
```

4.10 How Do I Enable Collaborative Development?

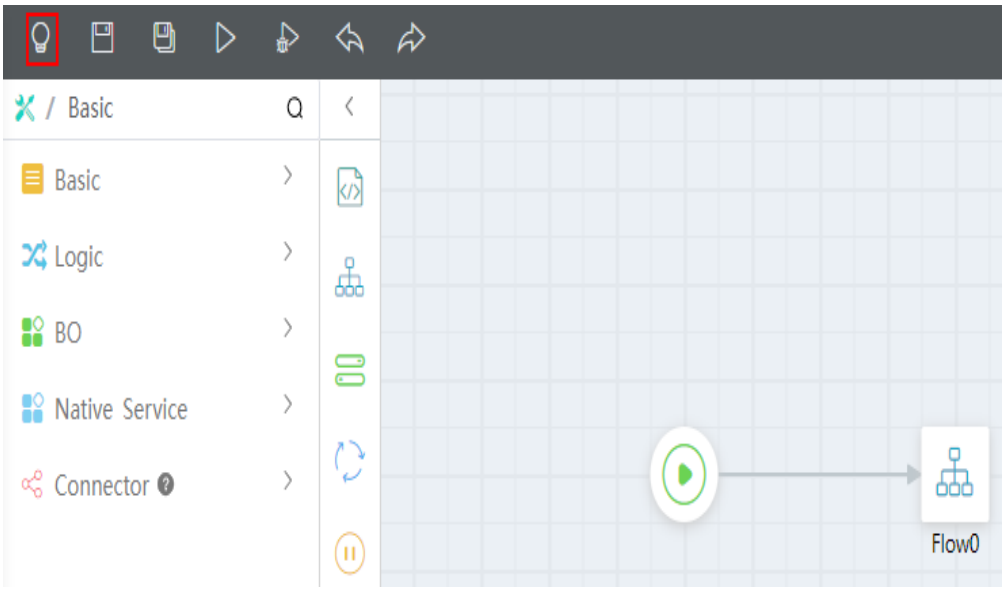
You can create multiple user accounts to support collaborative development. In Huawei Cloud Astro Zero, all member accounts under the same account can view and collaborate on development projects. To prevent conflicts, a lock mechanism is used to isolate users during development.

For example, in flow development, the platform uses a lock mechanism to prevent multiple users from editing the same flow simultaneously. If user A opens and edits flow X, and user B also opens flow X, the editor for User B will automatically enter lock mode, allowing only one user to edit at a time.

- User A has the permission to edit flow X. After user A finishes editing, closing the editing page will automatically unlock it. You can also unlock the page by clicking  .

  There are multiple scenarios where the editing interface for an element can be closed, such as exiting the element's editing interface, leaving the application designer, or closing the application designer's browser page.

Figure 4-38 Clicking the activation button



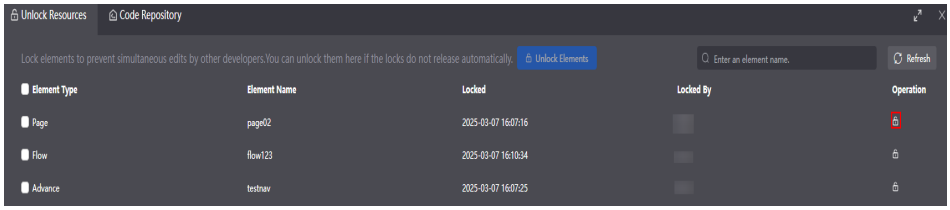
- When user A locks flow X, user B's operations on flow X are restricted to read-only mode, and user B's editor page displays: "This *resource name* has been locked by *user A*, click here to forcefully get edit authority." If user B clicks to forcibly take edit permission, user A's flow X editor page will automatically lock, and a message will indicate that the resource is now locked by user B.

  If an element is locked and you cannot contact the person who locked it, you can forcibly release the lock from the UI.

  a. In the navigation pane, choose **Code Management** > **Unlock Resources**.

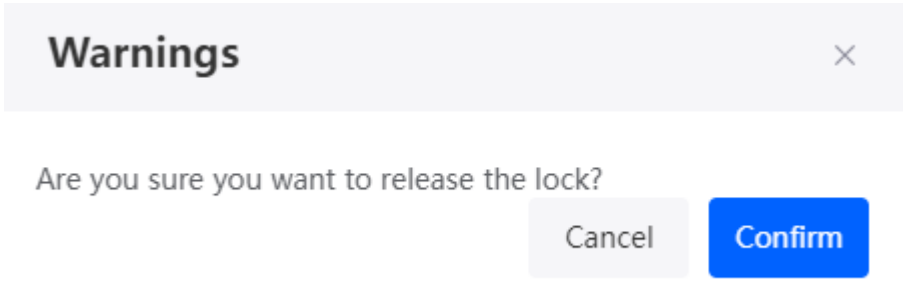
  b. In the locked element list, select an element and click  next to the element.

  Figure 4-39 Obtaining a lock

  c. In the warning dialog box, click the confirm to release the lock.

  Figure 4-40 Confirming whether to release the lock

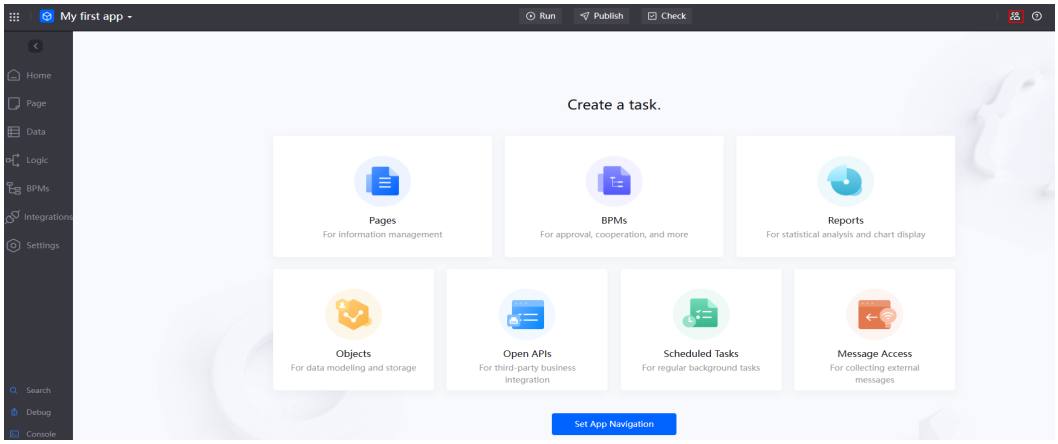
4.11 How Do I Control User Permissions During Collaborative Development?

When developers work on an application, you can assign different permissions (such as viewing, developing, managing) to control their operations.

Step 1 Log in to the application designer by referring to **Logging In to the Application Designer**.

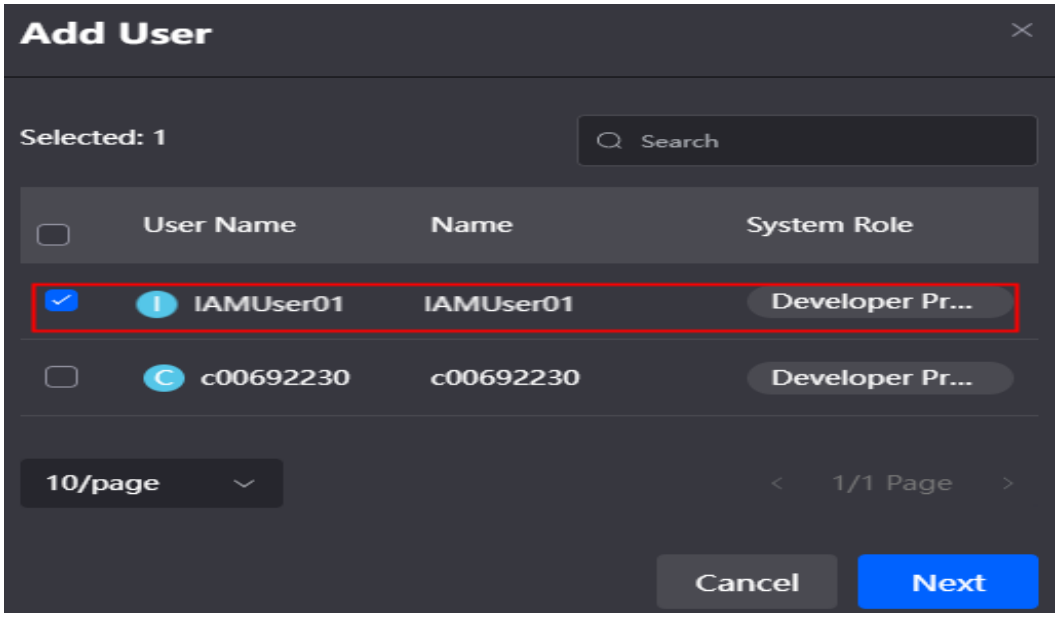
Step 2 In the upper right corner of the application designer, click . The page for collaborative development is displayed.

Figure 4-41 Entering the page for collaborative development



Step 3 Under **Management**, click **Add Developer**, then click **Add** next to **Added Users**, and select the users to collaborate on development.

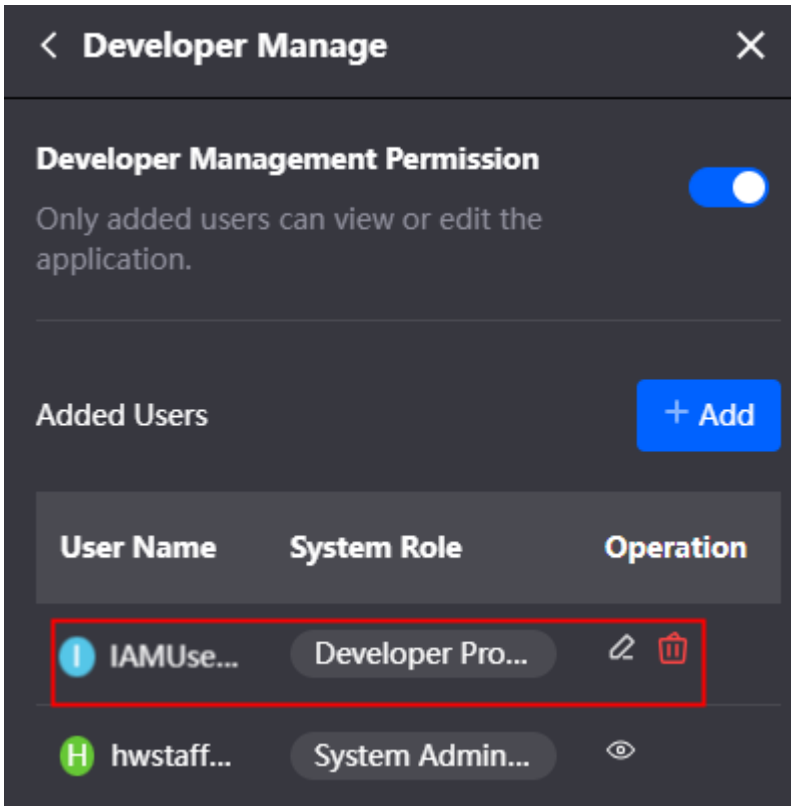
Figure 4-42 Selecting users for collaborative development



Step 4 Click **Next**, set permissions for the added user, and click **Confirm**.

- **View App**: allows you to access and view elements in an application through the application designer.
- **Edit App**: allows you to edit application information, settings, and elements within the application.
- **Delete App**: allows you to delete an application from the environment.
- **Manage Version**: allows you to manage application versions, including releasing, switching, and deleting versions.
- **Build Package**: allows you to package, install, and deploy applications.
- **Manage App**: allows you to manage application developers and security levels.

For example, add the **View App** and **Edit App** permissions for the developer user **IAMUser01**. After the setting, you can view the added user in **Added Users**. After logging in to the application as **IAMUser01**, you can view and edit the current application.



----End

5 Application Release and Deployment

5.1 What Should I Do If a Message Is Displayed Indicating That the *XXX* Field of Metadata Cannot Be Updated During Software Package Installation?

Description

During software package installation, a message is displayed indicating that the *xxx* field of metadata cannot be updated, or there is an invalid metadata parameter: The type of the *Y* field cannot be changed to *Z*, and the field length and number of decimal places cannot be shortened.

Possible Causes

The issue arises because the field type, length, or precision of the object in the installation package does not match those in the target environment.

Solution

To resolve the issue, choose one of the following methods:

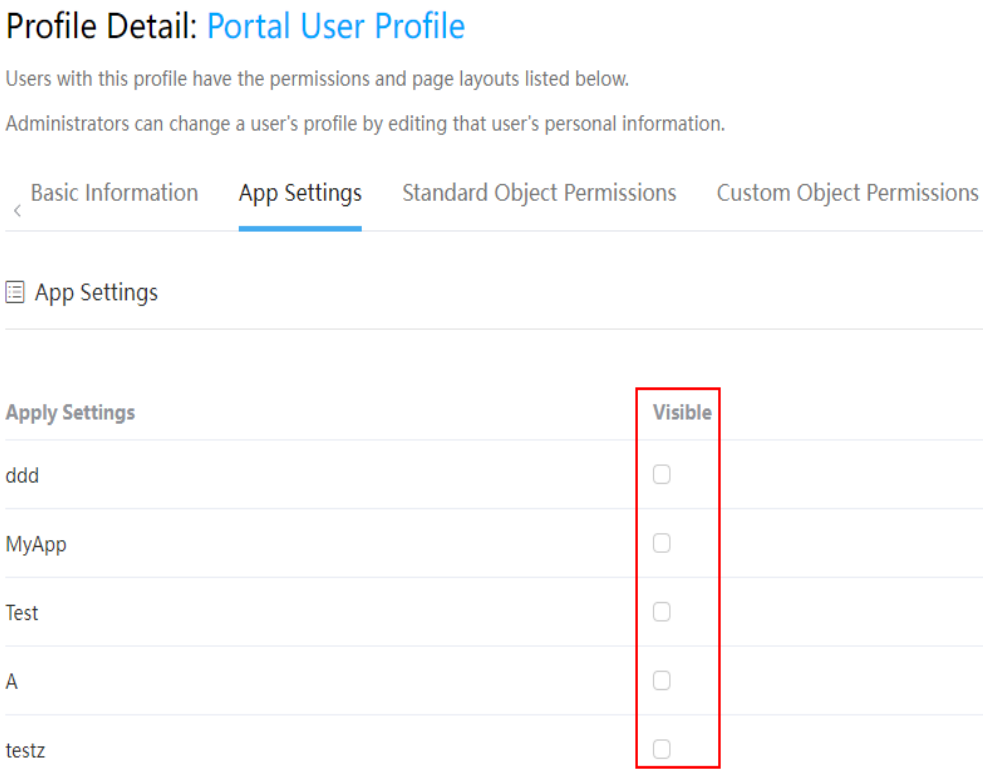
- Back up data, delete objects in the target environment, and reinstall the software package.
- Modify the fields in the development environment to match those in the target environment, then repack and reinstall the package.

5.2 What Should I Do If Installed Applications Is Not Displayed in the Application List?

After an application is installed, choose **Application Management** > **Applications** in the navigation pane of the environment configuration page. If the installed application is not found, the possible causes are as follows:

- In the runtime and sandbox environments, add-on applications are excluded from the list in **Environment Configuration** > **Application Management** > **Applications**.
- BO applications are not displayed in the application list.
- On the current user's profile, the installed application's visibility is not enabled on the **App Settings** tab.

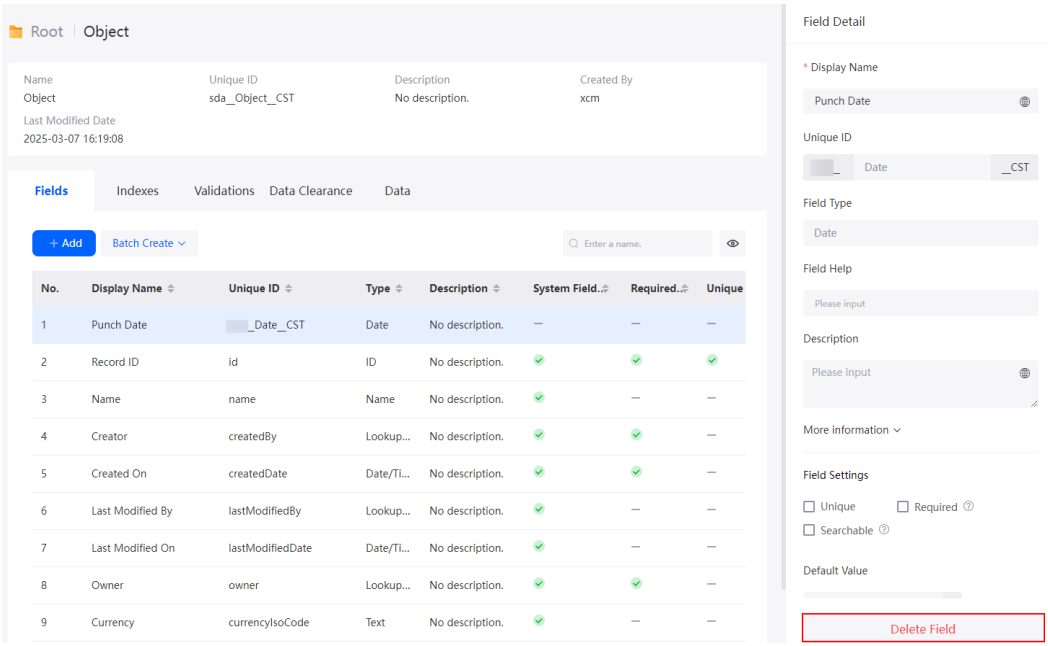
Figure 5-1 Disabled visibility



5.3 Why Does an Application Resource Remain After Deletion and New Version Release?

Huawei Cloud Astro Zero uses an incremental installation mechanism. When an application is installed, only additions or modifications are applied. Before the first release, you can add or delete application resources (such as object fields, scripts, flows, widgets, and service permissions credentials). Once the application is released and installed in environment B, deleting a resource in environment A does not affect the existing resources in environment B. For example, if an object field is deleted in environment A after release, the field will remain in environment B.

Figure 5-2 Deleting a field



5.4 What Should I Do If Software Installation Fails Due to a Unique Index Conflict?

Description

The software package fails to be installed, and a message is displayed indicating that a unique index conflict occurs.

Solution

Step 1 Use the console to check the data in the table where the conflict occurred.

- Method 1: Log in to the Huawei Cloud Astro Zero environment configuration page and click on the right to debug data.
- Method 2: Go to the application designer. In the lower part of the navigation pane, click to debug data.
- Method 3: Enter **https://*Huawei Cloud Astro Zero domain name*/studio/console.html** in the address box of a browser to access the console for data debugging.

Step 2 After the conflict data is found, perform the following operations:

- Delete data in the installation package.
- Delete data from the environment.
- Use the pre-processing and post-processing scripts to delete or modify existing data in the target environment.

For metadata other than scripts, flows, and BPMs, this error occurs if data with the same name but different IDs, or the same ID but different names, exists. To avoid

this, do not delete and recreate resources with the same name in the development environment. Instead, create resources with unique names.

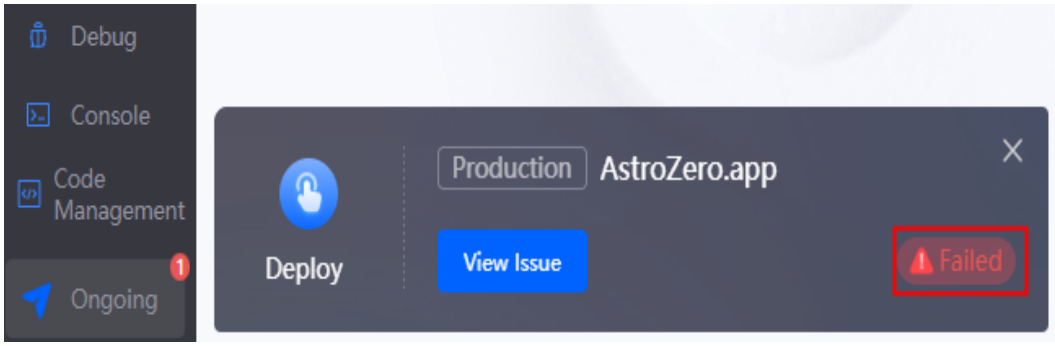
----End

5.5 How Do I Resolve One-Click Application Deployment Failures?

Description

When you choose **Publish** > **Release App** > **One-click deployment** on the menu bar of the application designer, a message indicating deployment failed is displayed.

Figure 5-3 Deployment failed



Solution

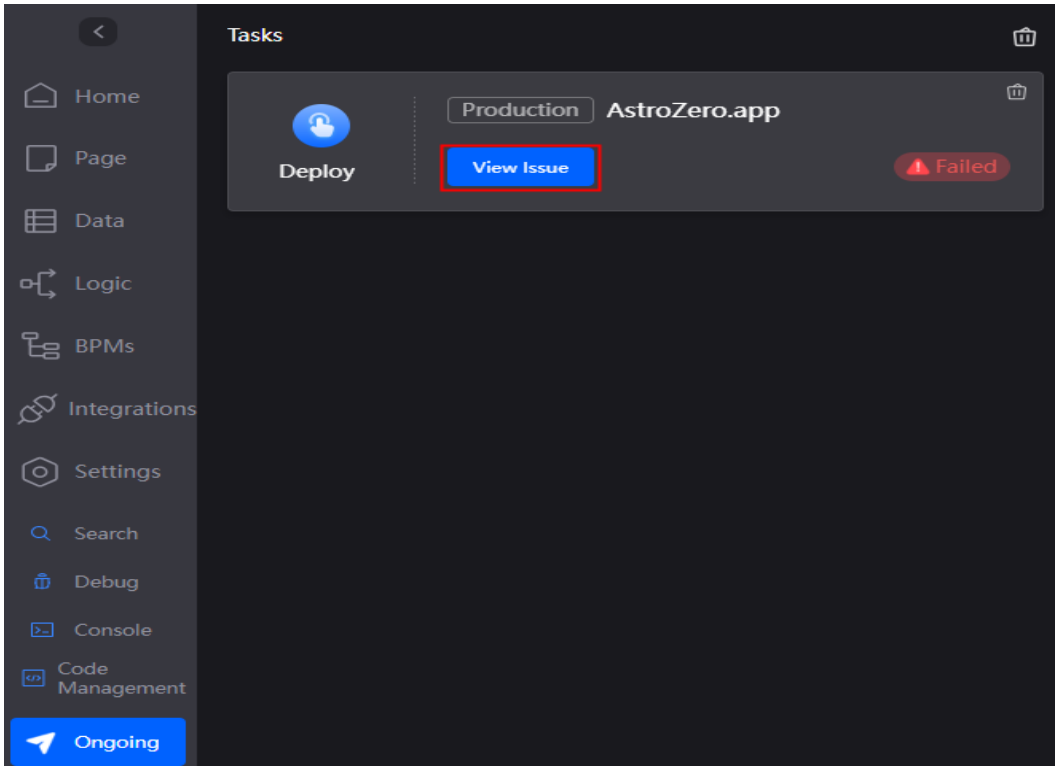
Application deployment might fail for different reasons. Check the error message on the page to find out what went wrong.

Step 1 Log in to the application designer by referring to **Logging In to the Application Designer**.

Step 2 In the navigation pane on the left, click **Ongoing**.

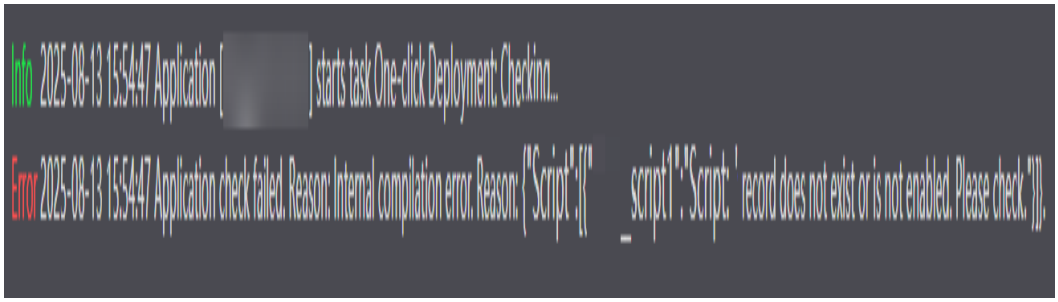
Step 3 In the task list, click **View Issue** in the corresponding deployment task.

Figure 5-4 Clicking View Issue



Step 4 Locate the fault based on the error message displayed on the page.

Figure 5-5 Viewing the error message



For example, **Figure 5-5** indicates that the **Namespace__script1** record does not exist or is not activated. After creating a script, you need to activate it. If not, the system will show a message saying the script does not exist or is not activated. To fix this, follow these steps:

1. In the navigation pane, choose **Logic**.
2. Under **Script**, click **Namespace__script1** to enter the script editor.
3. In the script editor, click  to activate the script.

Figure 5-6 Script inactivated

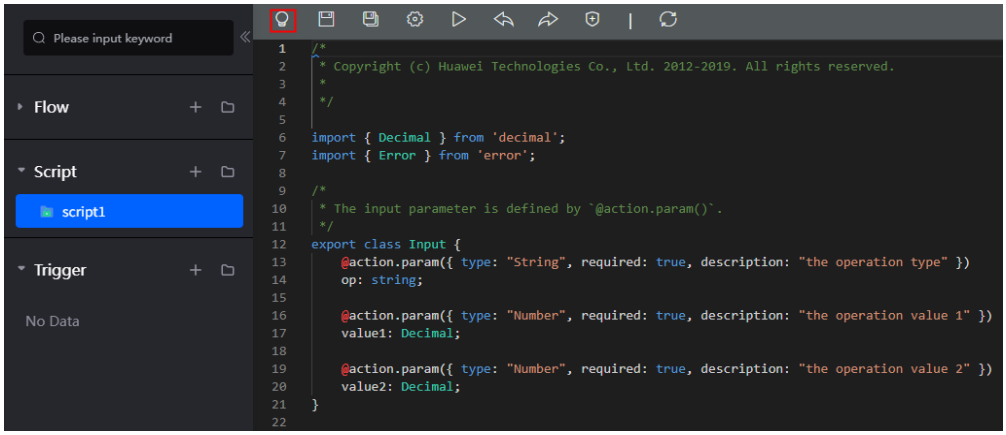
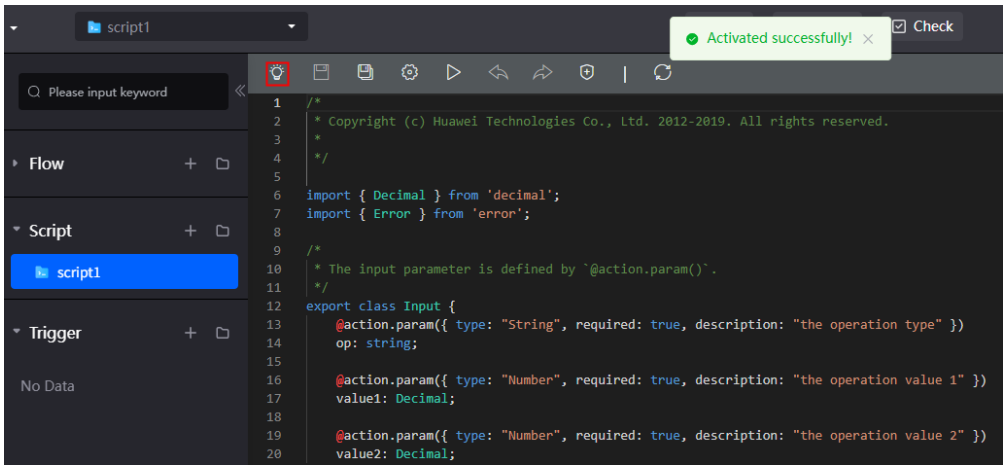
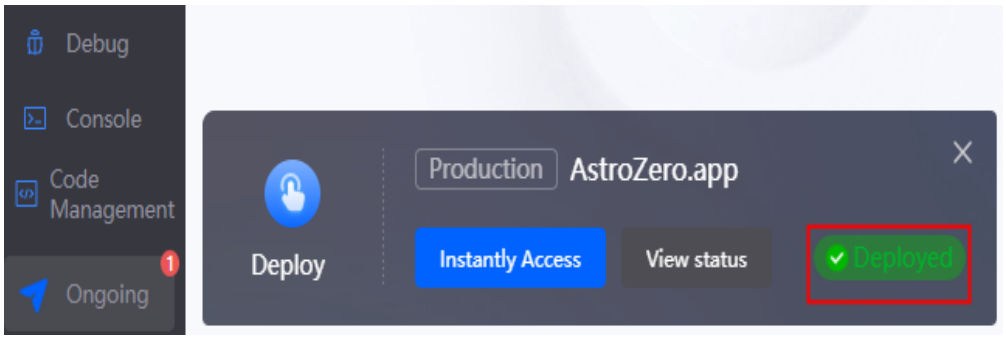


Figure 5-7 Script activated



4. After the script is activated, choose **Publish** > **Release App** > **One-click deployment**. The application can be properly deployed.

Figure 5-8 Application deployed successfully



----End

6 Management Center

6.1 How Do I Obtain the ID of a User or Portal User?

If you need to enter the user ID or portal user ID in the input parameters of the script, flow, or page code execution logic, you can use either of the following methods to obtain the ID.

Obtaining the ID of a User or Portal User

Step 1 On the Huawei Cloud Astro Zero console, click **Access Homepage** to go to the application development page.

Step 2 In the upper left corner of the page, click  and choose **Environments** > **Environment Configuration**.

Step 3 In the navigation tree, choose **User Security** > **Users**, or choose **Maintenance** > **Global Elements** > **Portal Users**.

Step 4 In the list on the right, click a user or portal user name to go to the user details page.

Step 5 The URL of the browser contains the user ID.

For example, if the URL of a user details page is https://*Huawei Cloud Astro Zero domain*/studio/index.html#/admin/user/***10gd000000bcXXXXXXXX***/detail, the information in bold is the user ID.

If the URL of a portal user details page is https://*Huawei Cloud Astro Zero domain*/studio/index.html#/admin/portaluser/***10gg000000mXXXXXXXXX***/detail, the information in bold is the portal user ID.

----End

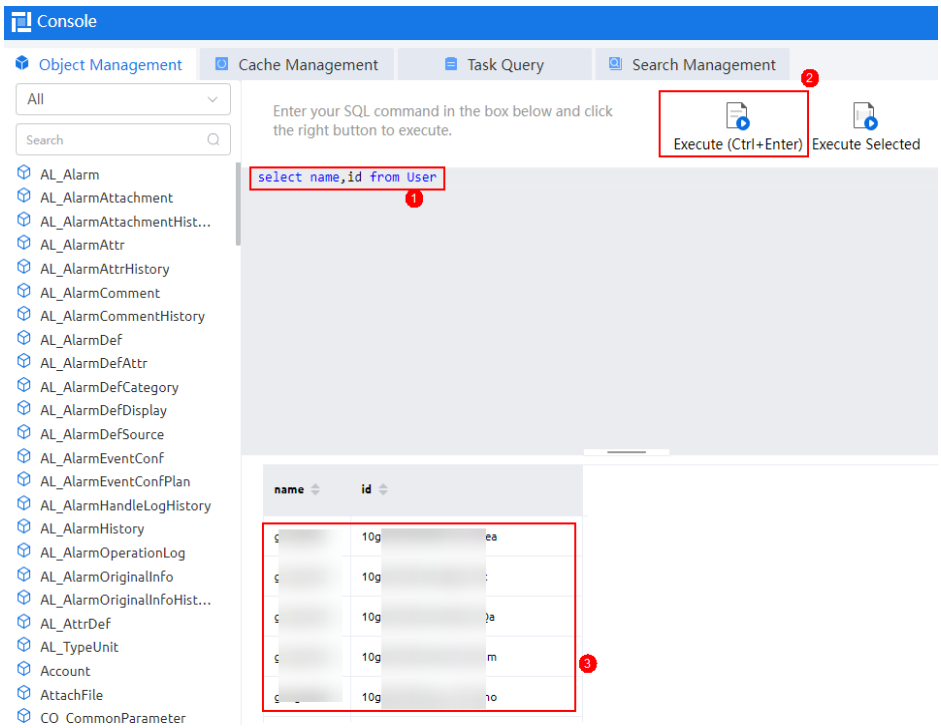
Using the Console to Obtain the IDs of All Users or Portal Users

Step 1 Go to the data debugging page by referring to **Querying Object Data on the Console**.

Step 2 Run the SQL statement to obtain the IDs of all users or portal users.

- The SQL statement for obtaining all user IDs is **select name,id from User**.
- The SQL statement for obtaining all portal user IDs is **select name,id from PortalUser**.

Figure 6-1 Obtaining all user IDs



----End

6.2 How Do I Disable Email Notification for Huawei Cloud Astro Zero Service Thresholds?

Description

In Huawei Cloud Astro Zero, when the service threshold usage exceeds 80%, the system automatically sends an email to notify you that the service threshold is about to exceed or has exceeded the threshold.

Solution

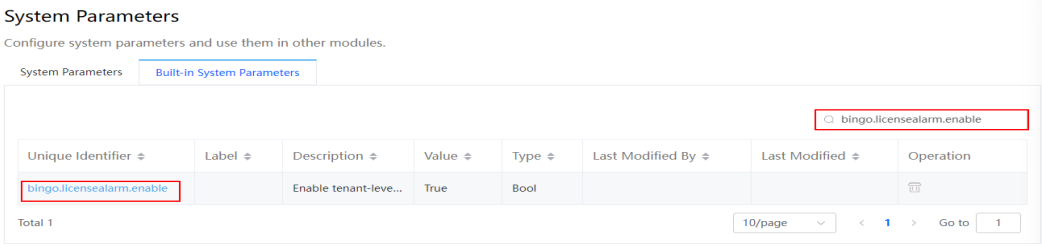
To disable email notification, perform the following operations:

Step 1 Log in to the Huawei Cloud Astro Zero environment configuration page.

Step 2 In the navigation pane, choose **System Settings** > **System Parameters**.

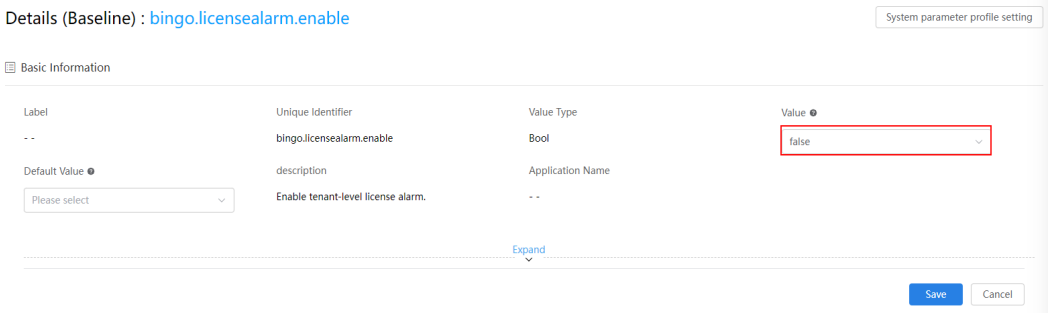
Step 3 On the **Built-in System Parameters** tab page, search for the built-in parameter **bingo.licensealarm.enable**.

Figure 6-2 Searching for bingo.licensealarm.enable



Step 4 Click the parameter name and change the value to **No**.

Figure 6-3 Changing the parameter value



----End

6.3 How Do I Query or Set the Session Duration (Validity Period) of a Huawei Cloud Astro Zero User?

Description

After a user logs in to Huawei Cloud Astro Zero, the user automatically logs out after timeout. Where can I query or set the session validity period?

Solution

Step 1 On the user or portal user details page, view the permissions of the user.

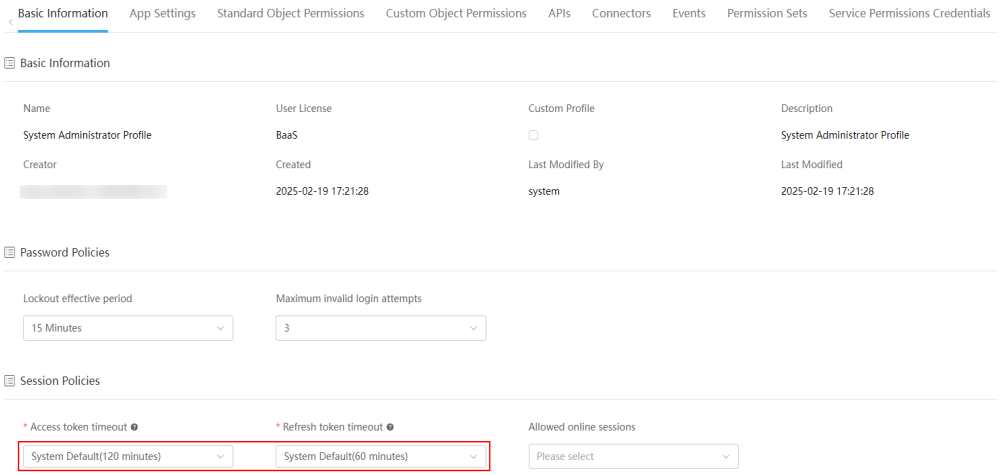
- For the System Administrator Profile, the default period is 30 minutes and cannot be changed.
- If other profiles are assigned, go to **Step 2**.

Step 2 The session validity period is controlled by the access credential duration of the assigned profile. The access credential and refresh credential must be both default or custom.

- Users (developer accounts of Huawei Cloud Astro Zero, who manage applications and portal users):

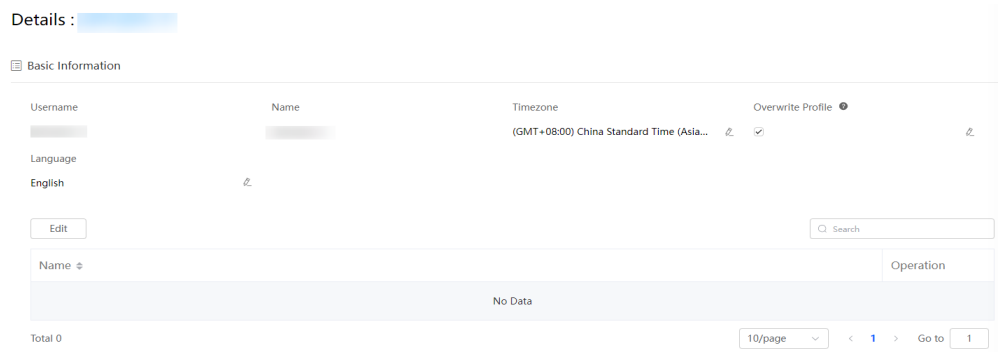
  Generally, only administrators with the System Administrator Profile can modify the access credential duration in the user permissions. Users can only view, but cannot change the access credential duration.

Figure 6-4 Page for editing the profile



- Portal users: The session validity period is controlled based on the access credential duration of the first profile in the permission set on the portal user details page. If no profile is set, the access credential duration of the Portal User Profile is used.

Figure 6-5 Portal user details page



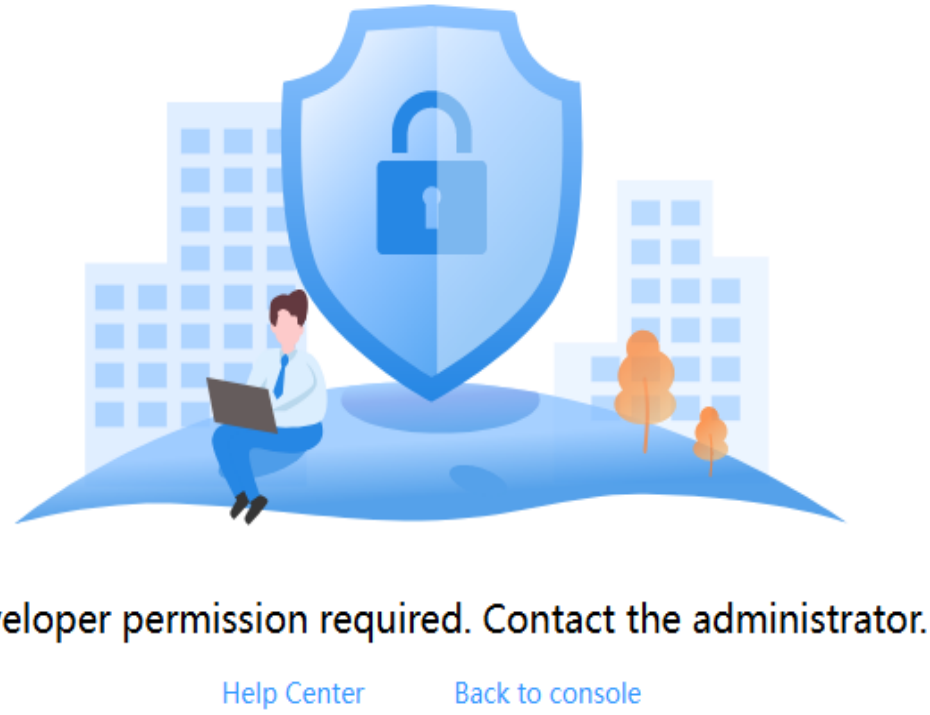
----End

6.4 How Do I Grant Developer Permissions to a Subuser After Authorization Fails?

Description

An authorized IAM user receives a message indicating insufficient permissions when clicking **Access Homepage** on the console.

Figure 6-6 No developer permissions



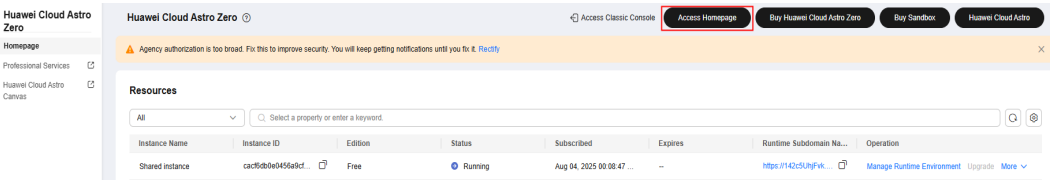
Solution

Huawei Cloud Astro Zero offers user management and profiles beyond IAM's authentication and authorization. It manages users and portal users, and controls their operation permissions. After the administrator authorizes, the authorized user needs to log in to Huawei Cloud Astro Zero and assign permissions to the IAM user.

Step 1 Log in to Huawei Cloud Astro Zero using the account you used to purchase the instance.

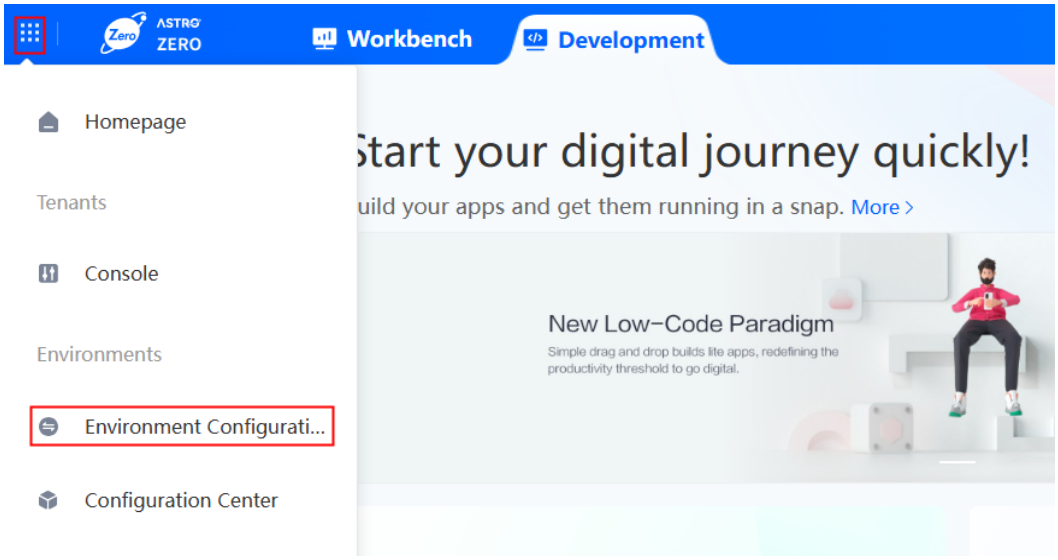
Step 2 On the console, click **Access Homepage** to go to the application development page.

Figure 6-7 Clicking Access Homepage



Step 3 Click  in the upper left corner of the page and choose **Environments** > **Environment Configuration**.

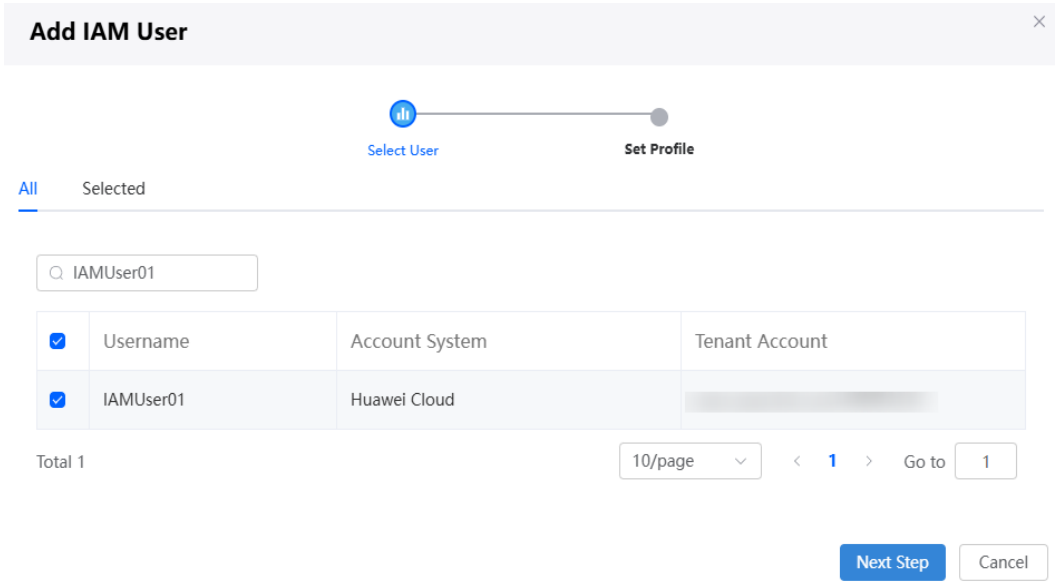
Figure 6-8 Selecting Environment Configuration



Step 4 In the navigation pane, choose **User Security** > **Users**. On the displayed page, click **Add IAM User**.

Step 5 On the **Add IAM User** page, select the target user and click **Next Step**.

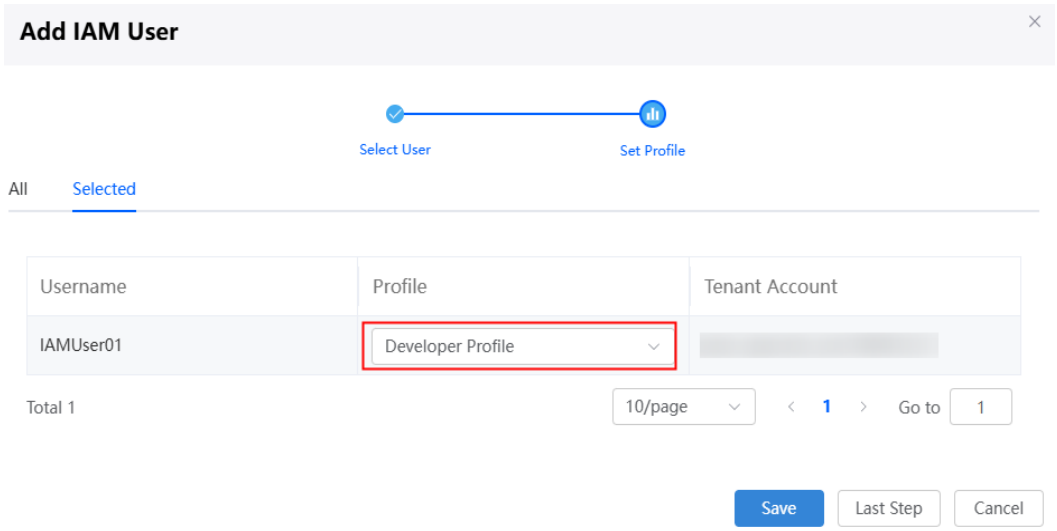
Figure 6-9 Selecting the IAM user to be authorized



Step 6 In the profile settings, select **Developer Profile** for the IAM user.

Developer Profile: developer permission. Users with this permission are developers who can develop applications, for example, adding an object, a field for an object, and a process. For details, see **Understanding Preset Profiles**.

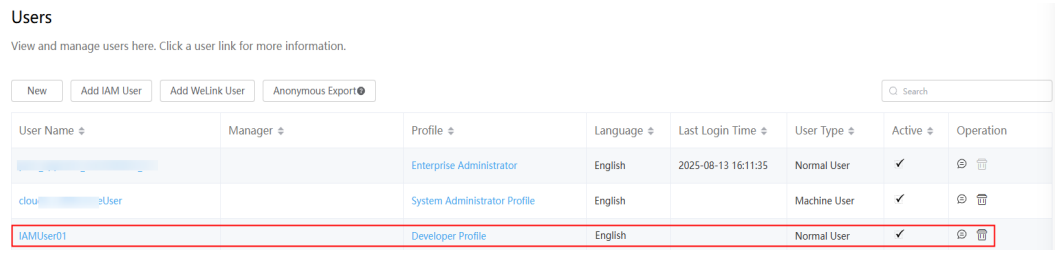
Figure 6-10 Adding Developer Profile to a user



Step 7 Click **Save** and return to the user list page.

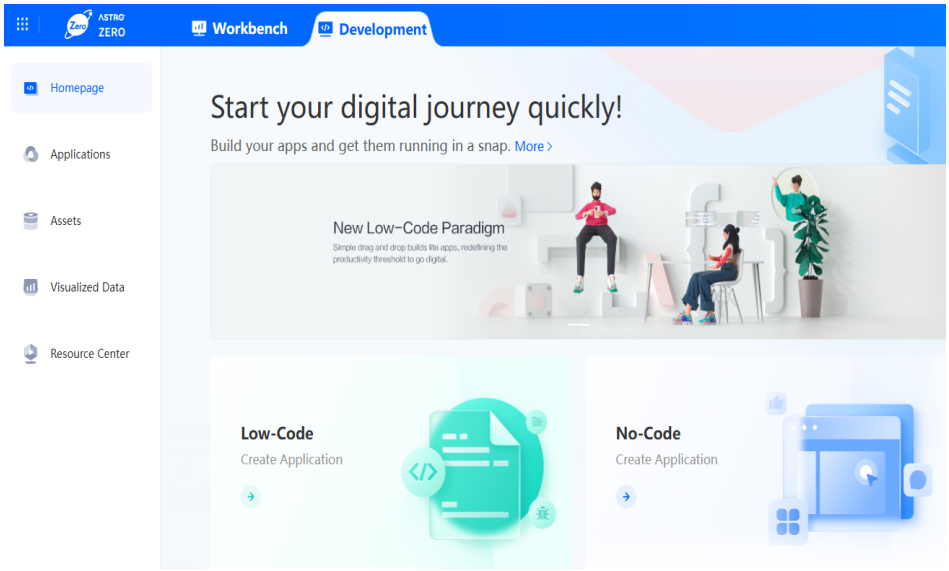
The added IAM user is displayed in the user list.

Figure 6-11 Viewing the added IAM user



Step 8 After the IAM user is added, log in to Huawei Cloud Astro Zero as the IAM user again. On **Homepage**, click **Access Homepage**. The application development page is displayed.

Figure 6-12 Application development page



----End

