CEC
2.5.0.0.0

Operation Management - Agent Workbench Integration with Third-Party Web Pages

Issue 01

Date 2024-03-01

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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 Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Overview

An enterprise can integrate customized pages into the CEC agent framework to implement specific functions, for example, viewing customer information on a customized page. Five integration modes are provided for your reference. You can select a proper integration mode based on your requirements.

Scenario	Screen Pop-up Type	Description	Applicable To
Scenario 1	External page +URL GET request parameters	Configure the URL of an external page to integrate an enterprise's own page into the CEC, when parameters are transferred in the URL in GET request mode.	Enterprises that have their own system pages and want to integrate existing pages into the CEC
Scenario 2	External page +URL POST request parameters	Configure the URL of an external page to integrate an enterprise's own page into the CEC, when parameters are transferred in the page console form in POST request mode.	Enterprises that have higher security requirements on data transfer based on scenario 1

Scenario	Screen Pop-up Type	Description	Applicable To
Scenario 3	External page +web API	Configure the URL of an external page to integrate an enterprise's own page into the CEC, when the JavaScript SDK is added for the enterprise to invoke to implement event listening and other functions, reducing the development workload.	Enterprises that want to directly invoke the event listening function to offer a better one-click outbound call function based on scenario 1
Scenario 4	Internal page +external API	Obtain information using the API provided by an enterprise, customize a page in the CEC based on the information, and integrate the page into the agent workbench.	Enterprises that have their own information systems and want to customize integration pages by directly transferring information using APIs
Scenario 5	Internal page +customized data	Record information to the CEC and use the saved information to integrate a page into the CEC agent workbench.	Enterprises that have a small number of customers and want to use the CEC to save some information and use the information to integrate pages

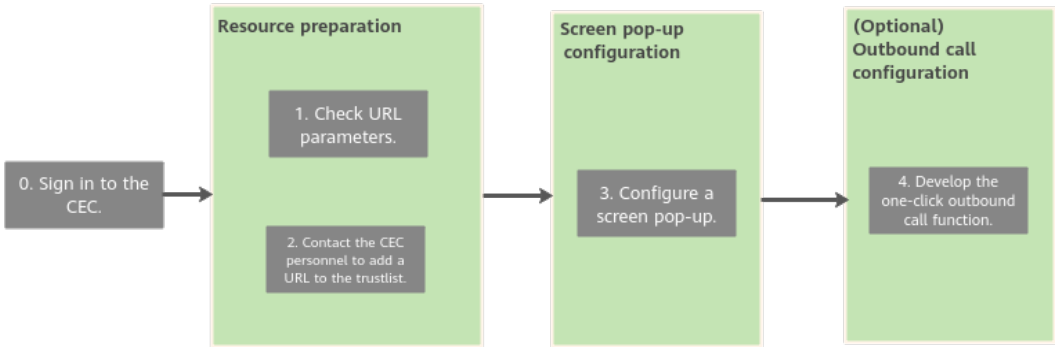
2 Scenario 1: Integration by Using a URL (in GET Parameter Transfer Mode)

2.1 Overview

When an enterprise directly uses the CEC agent framework as the call center system, the system can open a customized page (for example, page for viewing customer information) when an agent answers a call. This section describes the configuration.

The enterprise page embedded in the agent framework is implemented based on the inbound call screen pop-up configuration. The following figure shows the integration principle.

Figure 2-1 Integration principle



The data transmitted by the URL includes:

- Calling number
- Called number
- Media type (voice or multimedia)
- Call type (inbound call or outbound call)
- Platform agent ID of an agent
- Skill queue
- Skill queue ID
- Business account of an agent

2.2 Scenario Description

When an agent in the call center of an enterprise answers a call, the customer information details page of the calling number (customer) can be displayed at the same time.

The following describes the home page of the enterprise customer service system by taking https://www.huawei.com as an example.

2.3 Configuring an Inbound Call Screen Pop-up

Resource Preparation

1. A page provided by an enterprise can be properly displayed as an inbound call screen pop-up only when the following requirements are met:
   - At least one of the URL parameters listed in the following table can be used to display information on the GUI.

Table 2-1 URL parameters

Parameter	Description
callerNum	Calling number. • When the call center receives an inbound call, the customer number is the calling number. • When the call center makes an outbound call, the system access code (or the outbound calling number configured in the system) is the calling number.
calledNum	Called number. • When the call center receives an inbound call, the system access code is the called number. • When the call center makes an outbound call, the customer number is the called number.

Parameter	Description
callid	Session ID. Unique ID of the current session in a call. The value must be the same as that of **Call S/N** on the **Customer Contact History** > **Contact Record** page.
mediaType	Media type. • **1**: voice or video call • **5**: multimedia text chat
callType	Call type. • **0**: inbound call • **1**: outbound call (Multimedia text chats do not support this call type.)
workNo	Platform agent ID of the agent that performs the current operation.
callskill	Skill queue used by the agent in a call.
callskillid	ID of the skill queue used by the agent in a call.
agentName	Business account of the agent in a call.

  - The URL containing parameters can be opened using a browser.

    The following URL required by Huawei contains all preceding parameters:

    https://*www.huawei.com*/

2. Submit an application to the operations administrator to add https://www.huawei.com to the trustlist. To standardize the management of pages configured by enterprises, the CEC uses a trustlist to control pages that can be opened.

Procedure

Step 1 Sign in to the CEC as a tenant administrator and configure a screen pop-up.

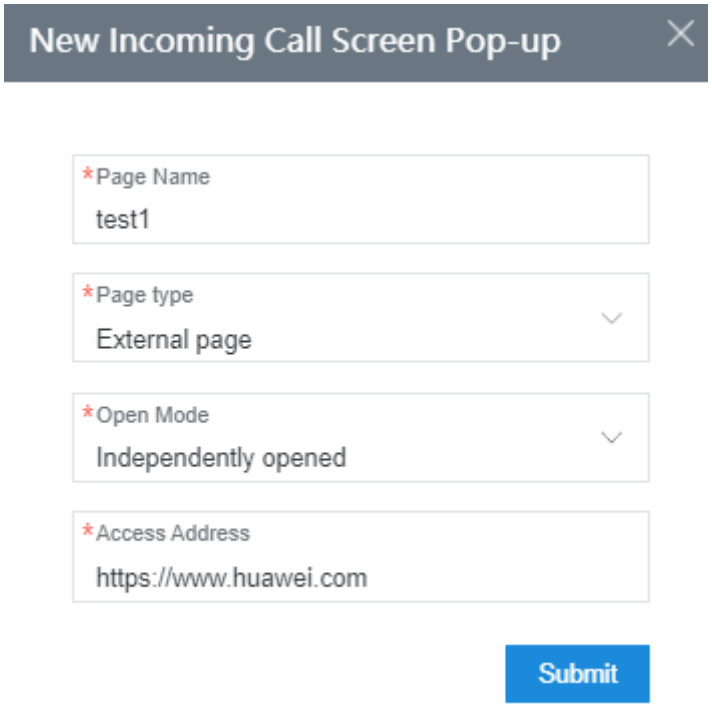
1. Choose **Configuration Center** > **Expansion Management** > **Screen Pop-up**.
2. Click the **Audio/Video** or **Multimedia** tab and configure a screen pop-up of the corresponding type.

   Screen pop-ups can be integrated into the corresponding agent workbenches. For example, an **Audio/Video** screen pop-up can be integrated into the **Audio and Video Workbench**, and a **Multimedia** screen pop-up can be integrated into the **Online Chat Workbench**.
3. Click **New** in the upper right corner. The page shown in **Figure 2-2** is displayed.
4. Enter the page name and URL for the screen pop-up and click **Submit**. In the **Successful** dialog box that is displayed, click **OK**.

   The maximum number of screen pop-ups of the **Audio/Video** and **Multimedia** types is specified by the system parameter **Max. Screen Pop-ups**

for Each Tenant. The value ranges from 1 to 10. By default, a maximum of five screen pop-ups can be added for each type.

Figure 2-2 Configuring an inbound call screen pop-up

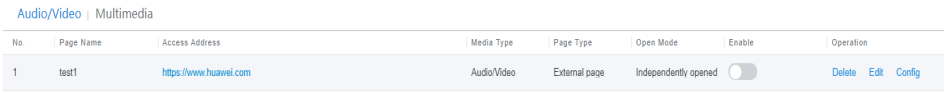


NOTE

The values of parameters in the URL dynamically change based on the actual session content. You can use the example values during configuration.

5. (Optional) Set screen pop-up parameters for the external page. If these parameters are not set, the following sixnine parameters are passed by default: **callerNum**, **calledNum**, **callid**, **mediaType**, **callType**, **callskill**, **callskillid**, **agentName**, and **workNo**. If these parameters are set, only the configured parameters are passed.
   a. Click **Config** in the **Operation** column, as shown in **Figure 2-3**.

      Figure 2-3 Setting parameters

   b. Click **New**. The maximum number of screen pop-up URL parameters to be passed is specified by the system parameter **Max. Parameters Transferred in URL of Screen Pop-up for Each Tenant**. The value ranges from 1 to 20. By default, a maximum of 10 parameters can be added.
   c. Customize **UrlParamName** (screen pop-up URL parameter) and set **RelSysParamType** (type of an associated system data parameter) to **Call Data** or **Channel Associated Data**, as shown in **Figure 2-4** and **Figure 2-6Figure 2-5**.

Figure 2-4 Selecting Call Data

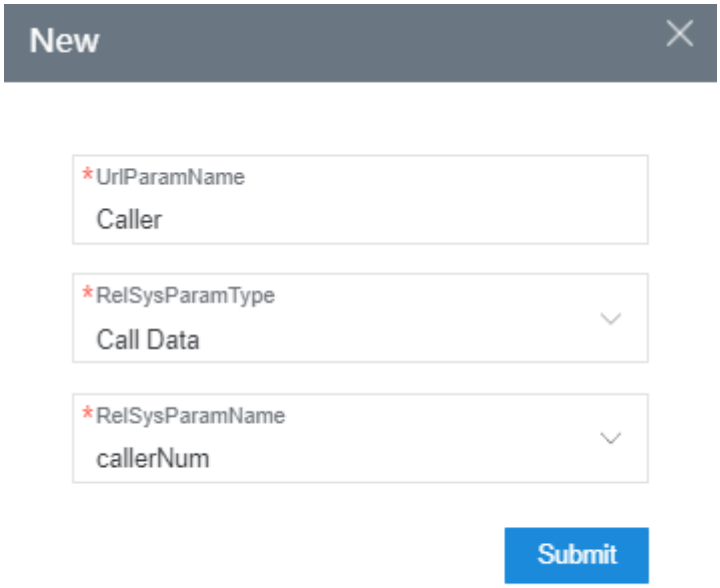


Figure 2-5 Selecting Channel Associated Data

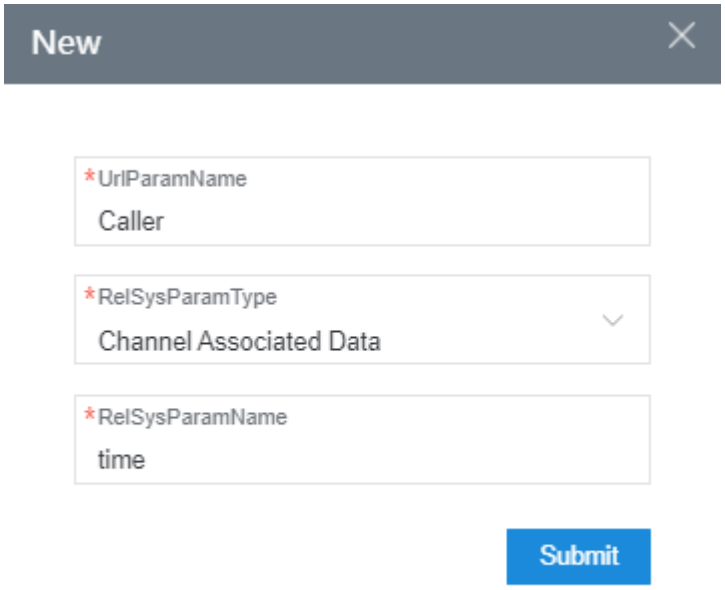
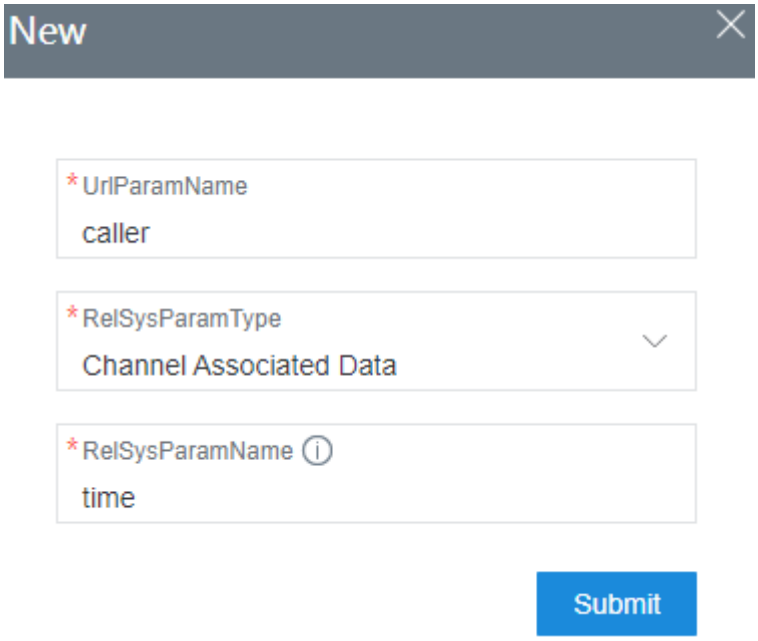


Figure 2-6 Selecting Channel Associated Data



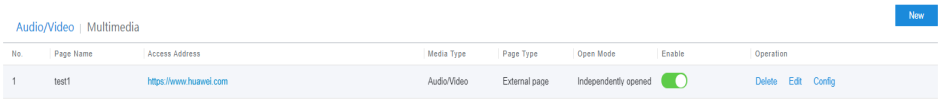
NOTE

The value of **RelSysParamName** (associated system data parameter) is customized.

d. Click **Submit**.

6. In the inbound call screen pop-up list, enable the new page, as shown in **Figure 2-7** (green button).

Figure 2-7 Enabling the page



If multiple inbound call screen pop-ups are enabled, click **Up** or **Down** in the **Operation** column to adjust the pop-up sequence of a screen pop-up. The screen pop-up on the top of the list pops up first.

7. Sign out the agent and sign in again for the configuration to take effect.

NOTE

If the page cannot be embedded and displayed after the preceding configuration is complete, see **2.6.1 What Do I Do If a Page Embedded in the CEC Cannot Be Opened?** under "FAQs."

----End

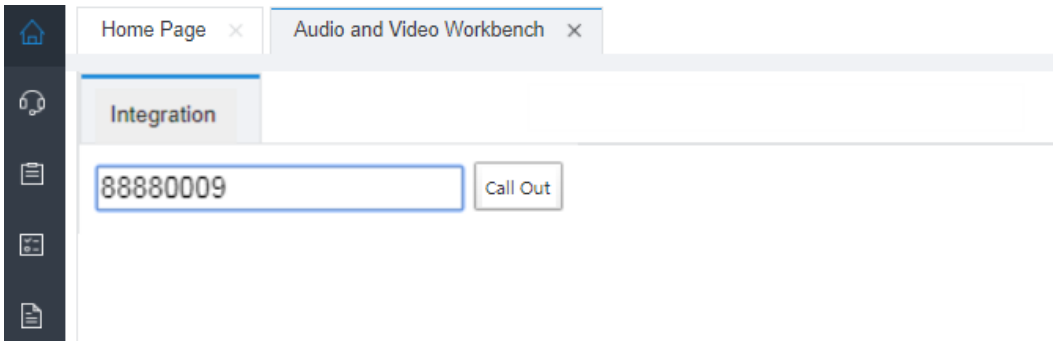
2.4 Implementing the One-Click Outbound Call Function

When an enterprise page is integrated into the multi-tenant framework of the CEC, an agent can click a customer's phone number on the enterprise page to make an outbound call by one click, without entering the phone number. The agent status on the connection bar is consistent. The outbound call function can be invoked on a third-party page integrated into the multi-tenant framework, as shown in **Figure 2-8**.

NOTE

The one-click outbound call function supports only external pages whose **Open Mode** is set to **Embed** in the **New Incoming Call Pop-up Screen** dialog box. For details about the setting, see **Figure 2-2**.

Figure 2-8 One-click outbound call function



For example, an inbound call screen pop-up path is configured for a tenant space. On the enterprise page loaded after the agent workbench is opened, you can make an outbound call by one click. The API parameters are transferred. The API reference code is as follows. For details about the name and parameters of the API, see *CC-Gateway Development Guide*.

```
let targetOrigin =gettUrlParam()['origin'];
  function sendMsg(){
     let ele =document.getElementById("calloutinput");
  let data={'name':'callout','param':[]};
  let number =ele.value;
  data.param.push(number);
  data.param.push('audio');

 top.postMessage(JSON.stringify(data),targetOrigin)
 }

 function  gettUrlParam()
 {
 let url = location.href;
 let paraString = url.substring(url.indexOf("?") + 1, url.length).split("&");
 let paraObj = {};
 for (let i = 0; i < paraString.length; i++) {
    let j = paraString[i];
    paraObj[j.substring(0, j.indexOf("="))] = j.substring(j.indexOf("=") + 1, j.length);
 }
 return paraObj;
 }
```

NOTE

1. A third-party system can invoke **data** in either of the following formats:
   - let data={
     name:'callout',
     param:{
     number:'88880523',//Number for making an outbound call
     mode:'audio'//The options are **audio** (voice call) and **video** (video call).
     }
     }
   - data={
     name:'callout',
     param:['88880813','video']
     }
2. The top.postMessage(JSON.stringify(data),targetOrigin) message is supported by the early connection bar and is compatible with the top.postMessage(data,targetOrigin) message.

2.5 Test and Verification

Prerequisites

You have enabled the voice, video, or multimedia agent function. For details, see **4.3.1 Enabling the Voice, Video, or Multimedia Agent Function in the Customer Service Cloud**.

Procedure

Step 1 Sign in to the CEC, choose **Configuration Center** > **Access Configuration** > **Called Route**, and view access codes.

Step 2 Click 

on the top of the page to sign in an agent.

Step 3 Simulate a user to dial a system access code using a test number.

Step 4 After the call is automatically connected, verify that the configured page can be automatically opened in the agent framework.

1. If screen pop-up URL parameters are set for the external page, the page URL contains the configured screen pop-up URL parameters (**UrlParamName**). (The URL contains the parameters in **Table 2-1** that are configured for the **Call Data** type.)

2. If screen pop-up URL parameters are not set for the external page, the page URL contains the parameters in **Table 2-1**.

3. To check parameters, press **F12** to enter the developer mode, click  , select the content on the external page, and check the parameters after **src** in **<iframe>**, as shown in the following figures.

Figure 2-9 Configured screen pop-up URL parameters

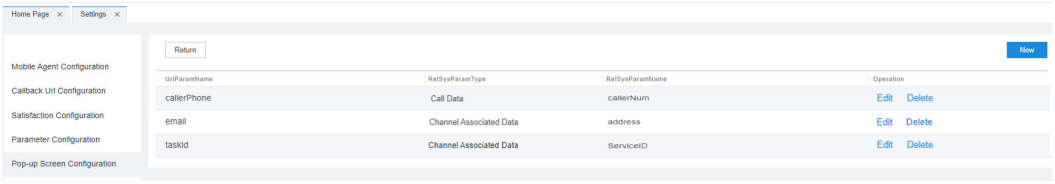


Figure 2-10 Checking configured screen pop-up URL parameters

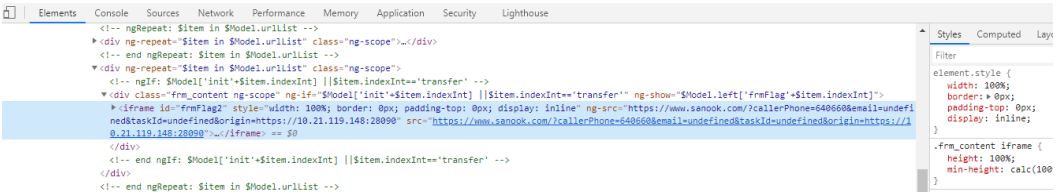
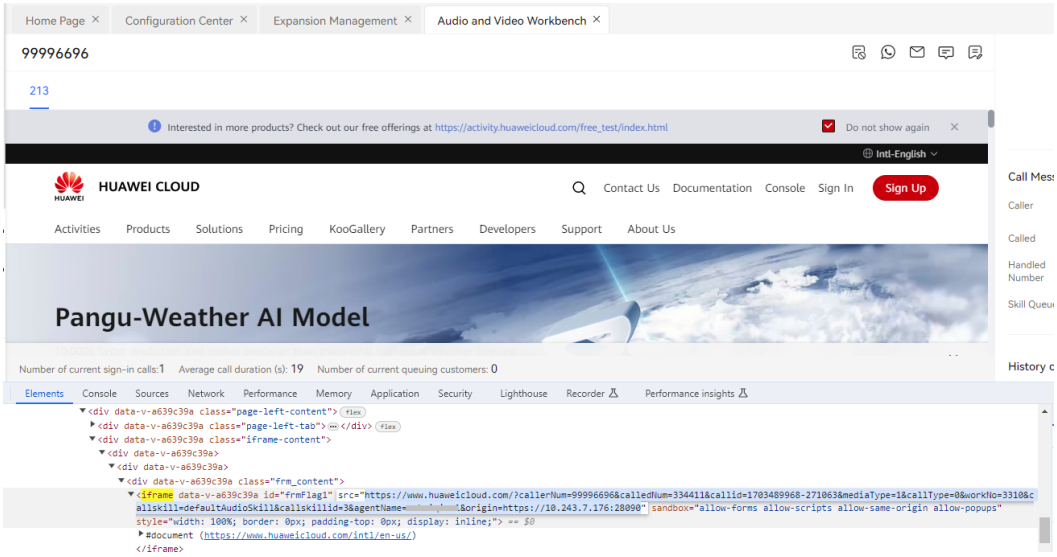


Figure 2-11 Checking configured screen pop-up URL parameters



Step 5 Test the one-click outbound call function on the screen pop-up on the workbench.

----End

2.6 FAQs

2.6.1 What Do I Do If a Page Embedded in the CEC Cannot Be Opened?

Symptom

After an inbound call screen pop-up is configured, the integrated enterprise page cannot be displayed in the CEC, and an error is displayed, indicating that the setting of **X-Frame-Options** is incorrect. **Figure 2-12** shows a possible error.

Figure 2-12 Error displayed on the page

Solution

This product uses iFrame to integrate enterprise pages. To integrate HTML pages, **X-Frame-Options** can be set only to a non-specific value or **ALLOW-FROM** *URI*, where *URI* indicates the URL of the CEC.

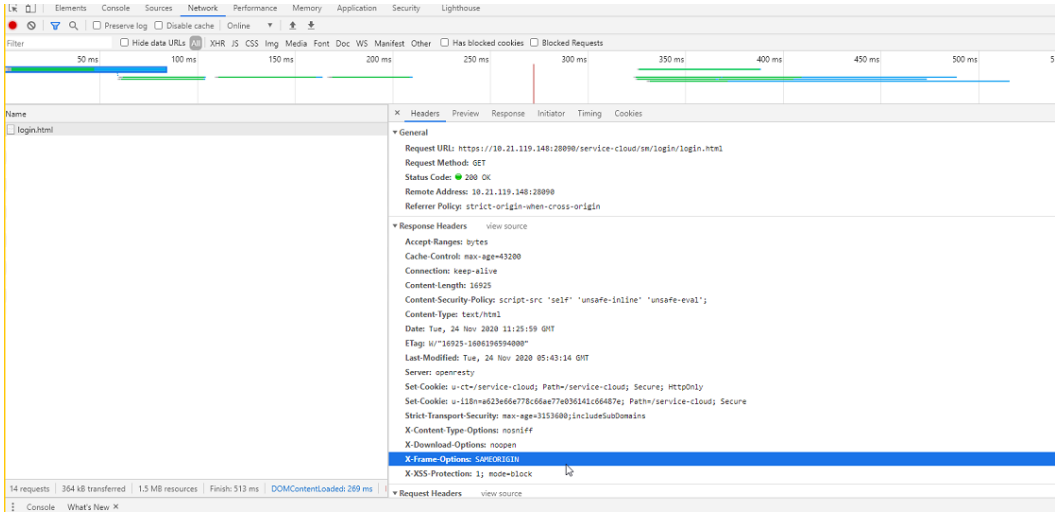
X-Frame-Options in the HTTP response header indicates whether a page can be displayed in **<frame>**, **</iframe>**, or **<object>**. This function allows a website to ensure that its content is not embedded in other websites, as shown in **Figure 2-13**.

Before integration, use the development tool to check the value of **X-Frame-Options** on the enterprise page to be embedded. The options are as follows:

- **DENY**: The page cannot be embedded in any page, including pages with the same domain name.
- **SAMEORIGIN**: The page can be embedded only in pages with the same domain name.
- **ALLOW-FROM** *URI*: The page can be embedded in the specified page.
- None: The page can be embedded in any page.

If the value of **X-Frame-Options** of the enterprise page is **DENY** or **SAMEORIGIN**, choose **Configuration Center** > **Expansion Management** > **Screen Pop-up**, click **Edit**, and set **Open Mode** to **Independently opened** in the **Modify Incoming Call Screen Pop-up** dialog box.

Figure 2-13 Checking the X-Frame-Options attribute



3 Scenario 2: Integration by Using a URL (in POST Parameter Transfer Mode)

3.1 Overview

In scenario 1, key parameters, such as **callerNum**, **calledNum**, **callid**, **mediaType**, **callType**, **callskill**, **callskillid**, **agentName**, and **workNo**, are passed using a URL. If you want the parameters to be passed not using a URL but using a page form to improve the integration security, select this scenario. In this scenario, you need to process the integration address form on the server.

3.2 Requirements Before Development

1. The page can be accessed in POST mode. You can use Postman, Insomnia, or other tools to perform a test. Enter the URL and select the POST access mode.
2. The page can process the submitted form content and render the processing result.
3. For an embedded page, the integrated party must allow iframe integration. For details, see **2.6.1 What Do I Do If a Page Embedded in the CEC Cannot Be Opened?**.
4. You have submitted an application to the system administrator to add https://www.huawei.com to the trustlist. To standardize the management of pages configured by enterprises, the CEC uses a trustlist to control pages that can be opened.

3.3 Configuring an Inbound Call Screen Pop-up

Sign in to the CEC as a tenant administrator and configure a screen pop-up.

Step 1 Choose **Configuration Center** > **Expansion Management** > **Screen Pop-up**.

Step 2 Click the **Audio/Video** or **Multimedia** tab and configure a screen pop-up of the corresponding type. Screen pop-ups can be integrated into the corresponding agent workbenches. For example, an **Audio/Video** screen pop-up can be integrated into the **Audio and Video Workbench**, and a **Multimedia** screen pop-up can be integrated into the **Online Chat Workbench**.

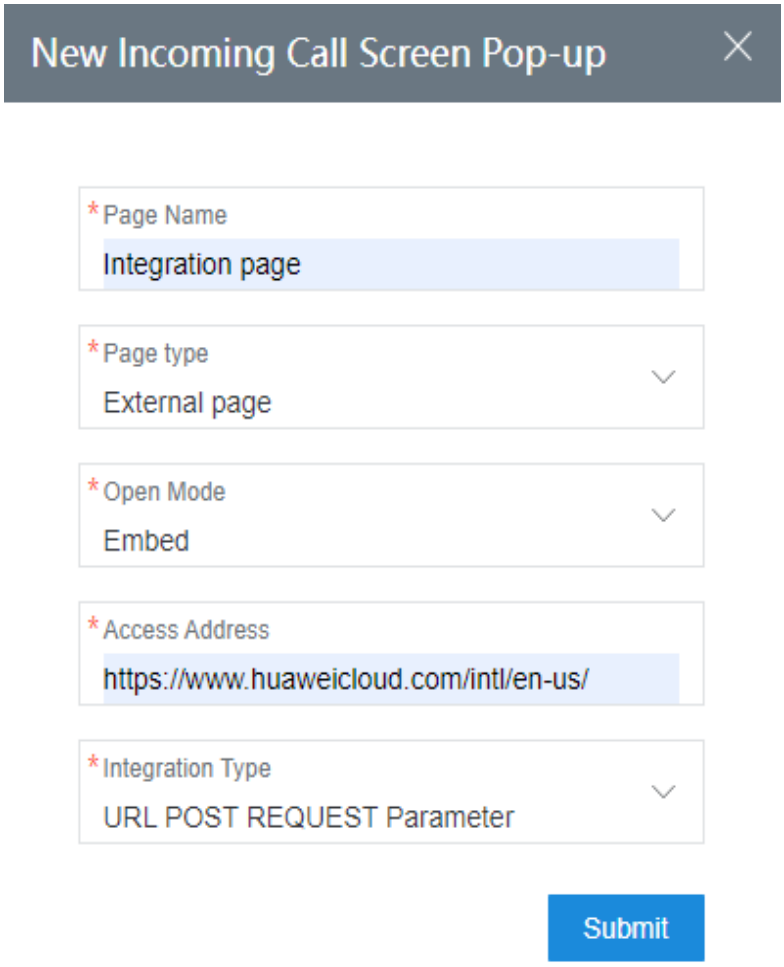
Step 3 Click **New** in the upper right corner. The page shown in **Figure 3-1** is displayed.

Step 4 Enter the page name and access address, set **Open Mode** to **Embed**, and set **Integration Type** to **URL POST REQUEST Parameter**. Click **Submit**. In the **Successful** dialog box that is displayed, click **OK**.

NOTE

The maximum number of screen pop-ups of the **Audio/Video** and **Multimedia** types is specified by the system parameter **Max. Screen Pop-ups for Each Tenant**. The value ranges from 1 to 10. By default, a maximum of five screen pop-ups can be added for each type.

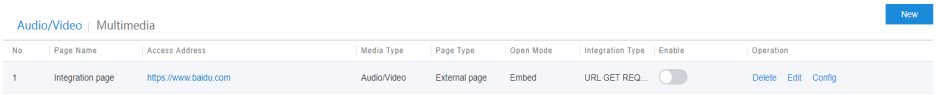
Figure 3-1 Configuring an inbound call screen pop-up



Step 5 (Optional) Set screen pop-up parameters for the external page. If these parameters are not set, the following sixnine parameters are passed by default: **callerNum**, **calledNum**, **callid**, **mediaType**, **callType**, **callskill**, **callskillid**, **agentName**, and **workNo**. If these parameters are set, only the configured parameters are passed.

1. Click **Config** in the **Operation** column, as shown in **Figure 3-2**.

   Figure 3-2 Setting parameters

2. Click **New**. The maximum number of screen pop-up URL parameters to be passed is specified by the system parameter **Max. Parameters Transferred in URL of Screen Pop-up for Each Tenant**. The value ranges from 1 to 20. By default, a maximum of 10 parameters can be added.
3. Customize **UrlParamName** (screen pop-up URL parameter) and set **RelSysParamType** (type of an associated system data parameter) to **Call Data** or **Channel Associated Data**, as shown in **Figure 3-3** and **Figure 3-4Figure 3-5**.

Figure 3-3 Selecting Call Data

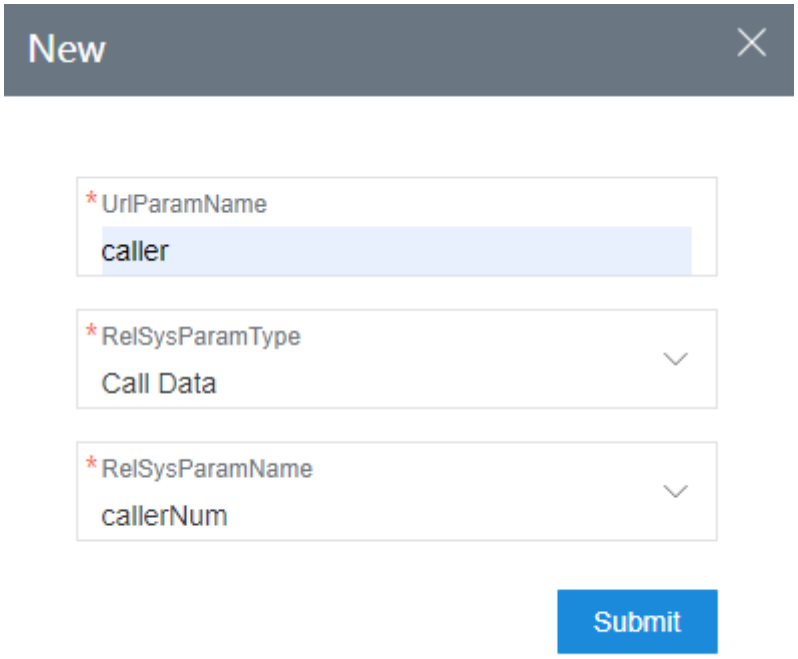


Figure 3-4 Selecting Channel Associated Data

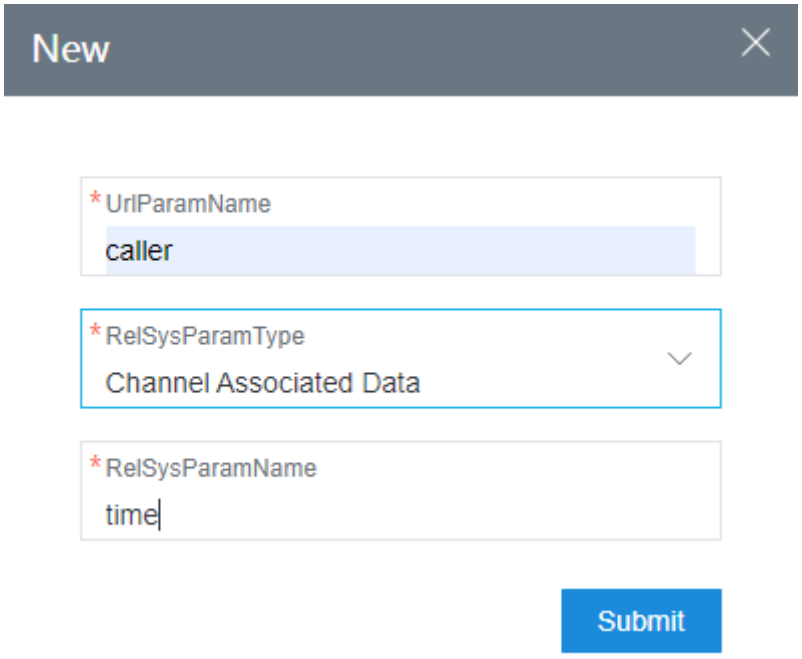
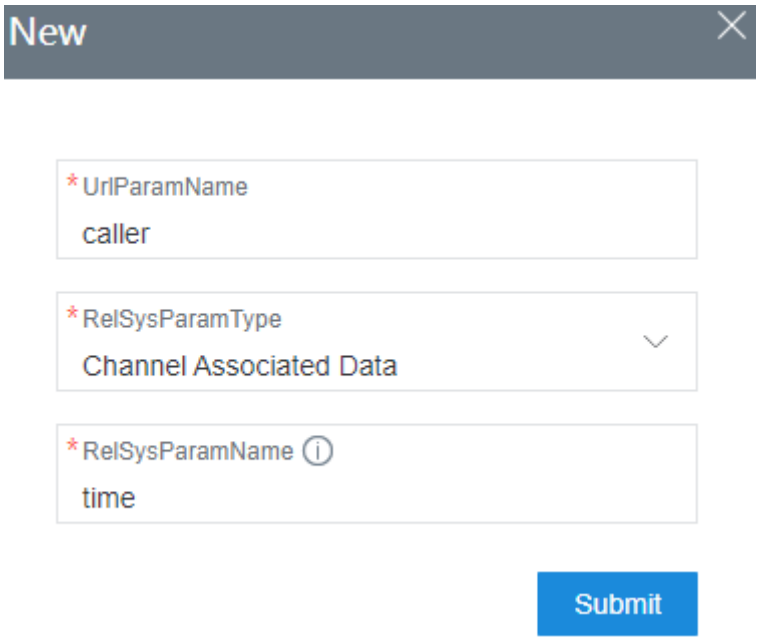


Figure 3-5 Selecting Channel Associated Data



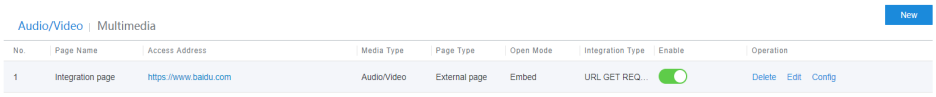
NOTE

The value of **RelSysParamName** (associated system data parameter) is customized.

4. Click **Submit**.

Step 6 In the inbound call screen pop-up list, enable the new page, as shown in **Figure 3-6** (green button).

Figure 3-6 Enabling the page



If multiple inbound call screen pop-ups are enabled, click **Up** or **Down** in the **Operation** column to adjust the pop-up sequence of a screen pop-up. The screen pop-up on the top of the list pops up first.

Step 7 Sign out the agent and sign in again for the configuration to take effect.

NOTE

If the page cannot be embedded and displayed after the preceding configuration is complete, see **2.6.1 What Do I Do If a Page Embedded in the CEC Cannot Be Opened?** under "FAQs."

----End

3.4 Processing Call Data

To obtain call data, you need to process and submit the integration address form on the server and add the processing logic to return the obtained data to the

page. The processing varies according to the server. **Table 3-1** describes the main call parameters passed. The sample code is in Java+Spring Boot+template (Thymeleaf) mode. For details, see the following steps.

Table 3-1 Parameter description

Parameter	Description
callerNum	Calling number. • When the call center receives an inbound call, the customer number is the calling number. • When the call center makes an outbound call, the system access code (or the outbound calling number configured in the system) is the calling number.
calledNum	Called number. • When the call center receives an inbound call, the system access code is the called number. • When the call center makes an outbound call, the customer number is the called number.
callid	Session ID. Unique ID of the current session in a call. The value must be the same as that of **Call S/N** on the **Customer Contact History** > **Contact Record** page.
mediaType	Media type. • **1**: voice or video call • **5**: multimedia text chat
callType	Call type. • **0**: inbound call • **1**: outbound call (Multimedia text chats do not support this call type.)
workNo	Platform agent ID of the agent that performs the current operation.
callskill	Skill queue used by the agent in a call.
callskillid	ID of the skill queue used by the agent in a call.
agentName	Business account of the agent in a call.

Step 1 Create a Maven project.

Step 2 Add the dependency on the template engine Thymeleaf to the **pom.xml** file of the project.

```
    <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```


Step 3 Compile the controller. The controller is used to receive parameters submitted using the form, save the parameters to a map object (**request** in the following figure), and pass the parameters to the template. **Figure 3-7** shows the sample code.

Figure 3-7 Sample code of the controller

```
@RequestMapping("/postParamModelType")
public ModelAndView getIndexPage(HttpServletRequest request, Model model) {
    String inputString = getInputString(request);
    ModelAndView modelAndView = new ModelAndView( viewName: "hello");
    Map<String, Object> map = new HashMap<>();
    if (StringUtils.hasText(inputString)) {
        String[] paramsArr = inputString.split( regex: "&");
        for (String temp : paramsArr) {
            String key = temp.split( regex: "=")[0];
            String value = temp.split( regex: "=")[1];
            map.put(key, value);
        }
    }
    model.addAttribute( s: "map", map);
    return modelAndView;
}

private String getInputString(HttpServletRequest request) {
    StringBuilder builder = new StringBuilder();
    try (BufferedReader reader = request.getReader()) {
        String line;
        while ((line = reader.readLine()) != null) {
            builder.append(line);
        }
        log.info(builder.toString());
    } catch (Exception e) {
        log.error( s: "get reader error", e);
    }
    return builder.toString();
}
```

Step 4 Add the corresponding Thymeleaf template. **Figure 3-8** shows the resource path.

Figure 3-8 Resource path

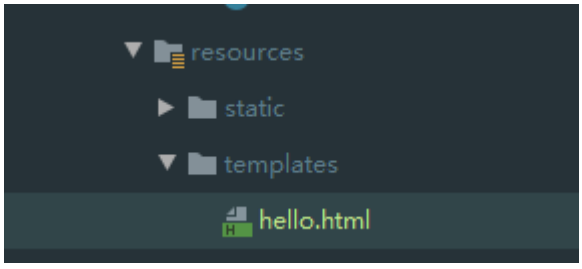


Figure 3-9 shows the content of **hello.html**.

Figure 3-9 hello.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>hello world</h1>
    <table>
        <tr th:each="mapItem : ${map}">
            <td th:text="${mapItem.key} + | --- | + ${mapItem.value}"></td>
        </tr>
    </table>
</body>
</html>
```

----End

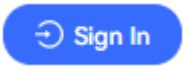
3.5 Test and Verification

Prerequisites

You have enabled the voice, video, or multimedia agent function. For details, see **4.3.1 Enabling the Voice, Video, or Multimedia Agent Function in the Customer Service Cloud**.

Procedure

Step 1 Sign in to the CEC, choose **Configuration Center** > **Access Configuration** > **Called Route**, and view access codes.

Step 2 Click 

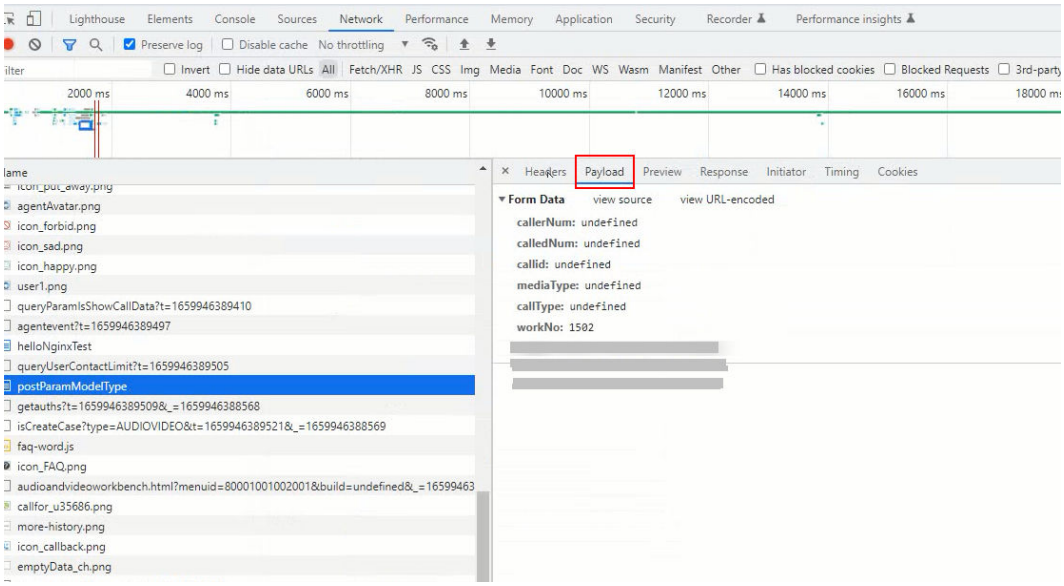
on the top of the page to sign in an agent.

Step 3 Simulate a user to dial a system access code using a test number.

Step 4 After the call is automatically connected, verify that the configured page can be automatically opened in the agent framework. If the configured page is an **Audio/Video** screen pop-up, it is opened on the **Audio and Video Workbench**. If it is a **Multimedia** screen pop-up, it is opened on the **Online Chat Workbench**.

Step 5 Press **F12** to enter the developer mode and verify that the parameters in the form are correct, as shown in **Figure 3-10**.

Figure 3-10 Parameters passed using the form



Step 6 Test the one-click outbound call function on the screen pop-up on the workbench.

----End

4 Scenario 3: Integration by Using a URL (in Page JavaScript Invocation Mode)

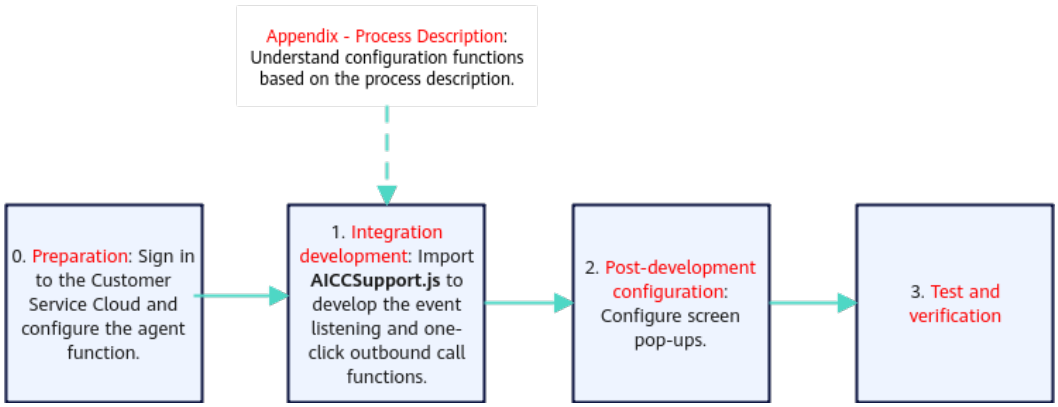
4.1 Overview

AICCSupport.js is used in the scenario where third-party pop-up windows need to be integrated into the voice and video workbench or online chat workbench. During a call, an agent can easily obtain the event parameters passed by the AICC based on the event registration method provided by JavaScript. In addition, an agent can invoke the one-click outbound call API to implement functions such as one-click voice and video calls on the GUI.

4.2 Development Procedure

Develop functions according to the development procedure shown in **Figure 4-1**. For details about the implementation process of each function, see **4.6 Appendix - Process Description**.

Figure 4-1 Development procedure



4.3 Preparation

4.3.1 Enabling the Voice, Video, or Multimedia Agent Function in the Customer Service Cloud

Procedure

Step 1 Sign in to the Customer Service Cloud as a tenant administrator. Obtain the account and password from Huawei O&M personnel after the tenant space is provisioned.

Step 2 Add a voice, video, or multimedia skill queue based on the agent type that you purchase.

1. Choose **Configuration Center** > **Employee Center** > **Skill Queue**. The skill queue management page is displayed, as shown in **Figure 4-2** and **Figure 4-3**.

   NOTE

   - A maximum of 1000 skill queues can be created by default.
   - To create a multimedia skill queue, set **Type** to **Multimedia** and then set **Subtype**. The options include **WECHAT**, **EMAIL**, **WHATSAPP**, and **WEB**.

2. Click **New** and set parameters based on **Table 4-1**.

Figure 4-2 Page for configuring a video skill queue

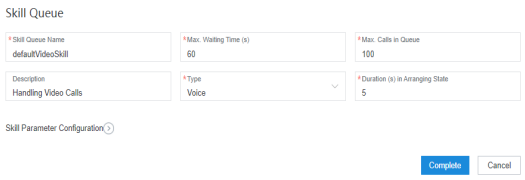


Figure 4-3 Page for configuring a multimedia skill queue

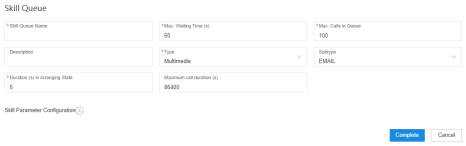


Table 4-1 Parameters for configuring a skill queue

Parameter	Mandatory or Not	Description
Skill Queue Name	Yes	The value can contain a maximum of 20 characters and cannot contain spaces.
Max. Waiting Time (s)	Yes	The default value is **60**. The unit is second. The value ranges from 1 to 60000.
Max. Calls in Queue	Yes	The default value is **100**. The value ranges from 1 to 10000.
Description	Yes	A maximum of 50 characters are allowed.
Type	Yes	The options are as follows: – **Voice**: A voice skill queue handles voice businesses. – **Multimedia**: A multimedia skill queue handles multimedia businesses. – **Video**: A video skill queue handles video businesses. – **Voice Click to Dial**: A voice click-to-dial skill queue is used together with multimedia businesses. During a text chat with an agent, a customer can directly make a voice call to the agent. – **Video Click to Dial**: A video click-to-dial skill queue is used together with multimedia businesses. During a text chat with an agent, a customer can directly make a video call to the agent. **NOTE** ▪ To create a multimedia skill queue, set **Type** to **Multimedia** and then set **Subtype**. The options include **WECHAT**, **EMAIL**, **WHATSAPP**, and **WEB**. ▪ Click-to-dial skill queues apply only to the web channel. The default value is **Voice**.

Parameter	Mandatory or Not	Description
Duration (s) in Arranging State	Yes	Duration during which an agent is in wrap-up state after a call ends. The default value is **5**. After this duration, the agent enters the idle state and can answer calls from customers. The value ranges from 0 to 3600.

Parameter	Mandatory or Not	Description
Skill Parameter Configuration	No	Personalized configurations, which are the processing policies when a customer calls a skill queue and the call cannot be connected. The options are as follows: – **Skill Timeout** ▪ **Process Method**: Processing policy when queuing times out because no idle agent can answer the call. ○ **Release** (default) ○ **Transfer** ▪ **Device Type**: If **Process Method** is set to **Transfer**, you need to configure the skill queue or IVR flow to which the call is transferred. ○ **Skill Queue** ○ **IVR** – **Skill Busy** ▪ **Process Method**: Processing policy when the customer is queuing because no idle agent can answer the call, or the number of queuing customers exceeds the upper limit. ○ **Release** (default) ○ **Transfer** ▪ **Device Type**: If **Process Method** is set to **Transfer**, you need to configure the skill queue or IVR flow to which the call is transferred. ○ **Skill Queue** ○ **IVR** – **Skill NoAgents** ▪ **Process Method**: Processing policy when no agent can answer the call because no agent is on duty. ○ **Release** (default) ○ **Transfer** ▪ **Device Type**: If **Process Method** is set to **Transfer**, you need to configure the skill

Parameter	Mandatory or Not	Description
		queue or IVR flow to which the call is transferred. ○ **Skill Queue** ○ **IVR** – **Queuing and waiting configuration**: When a customer needs to wait in a queue after making an inbound call, a waiting tone can be played to optimize the customer waiting process. ■ **Queuing Method** ○ **Default Wait Tone** ○ **Customizing the Wait Tone** ○ **IVR** – **Keeping and waiting configuration**: When a call needs to be held and the customer needs to wait, a waiting tone can be played to optimize the customer waiting process. ■ **Keeping Method** ○ **Default Keeping Tone** ○ **Customizing the Keeping Tone** – **Skill AnswerMode**: After an agent answers a call from a customer, the employee ID of the agent can be played to the customer. ■ **Answer Type** ○ **Report employee ID** ○ **Report no voice** **NOTE** Calls in voice, video, and click-to-dial skill queues can be transferred to IVR or skill queues. Calls in multimedia skill queues can be transferred only to skill queues. The skill queue selected for call transfer must be of the same type as the skill queue to be created. The waiting tones can be set only for voice and video skill queues and are not displayed for multimedia and click-to-dial skill queues.

3. Click **Complete**.

Step 3 Configure a called route of the voice, video, or multimedia type.

1. Choose **Configuration Center** > **Access Configuration** > **Called Route**.

2. Click **New** to add parameter information for the VDN and click **Complete**, as shown in **Figure 4-4**. **Table 4-2** describes the parameters.

Figure 4-4 Page for configuring a called route

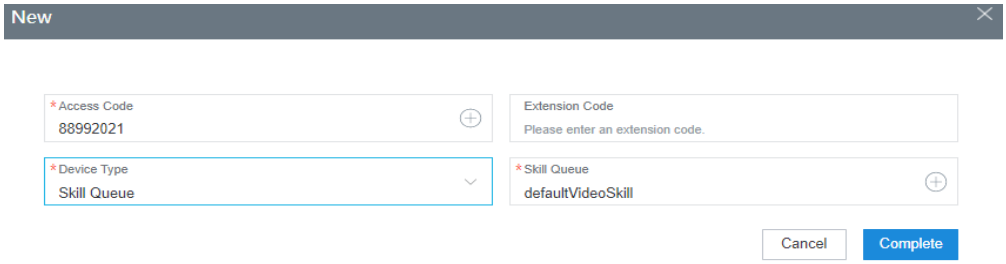


Table 4-2 Parameters for configuring a called route

Parameter	Mandatory or Not	Description
Access Code	Yes	Customer service hotline. Customers can dial the access code to connect to agents. Click ⊕ to select an access code, for example, a multimedia access code, from the list in the dialog box that is displayed.
Extension Code	No	To set one access code for multiple destination devices, you can configure extension codes. For example, if the access code is 12345, you can add extension code 1 to route calls to skill queue A and extension code 2 to route calls to skill queue B. In this way, a customer can dial 123451 to directly access skill queue A.
Device Type	Yes	Select **Skill Queue** to configure a called route of the skill queue type.
Skill Queue	Yes	Associate the skill queue created in **Step 2**. Click ⊕ to select a skill queue from the list in the dialog box that is displayed. The type of the skill queue is the same as that of the access code. For example, if you select a multimedia access code, all available skill queues are of the multimedia type.

Step 4 Configure a business account and skill queue.

1. Choose **Configuration Center** > **Employee Center** > **Agent Management**.
2. Select an agent ID and click **Configure** in the **Configure** column. The page for configuring agent information is displayed, as shown in **Figure 4-5**. **Table 4-3** describes the parameters.

3. Associate a business account and voice, video, or multimedia skill queue with the agent.

Figure 4-5 Page for configuring agent information

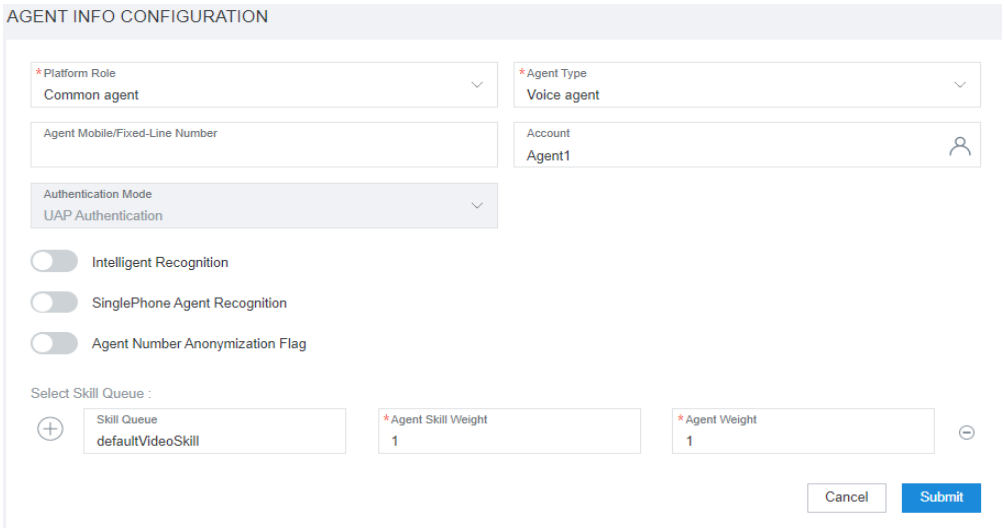


Table 4-3 Parameters for configuring agent information

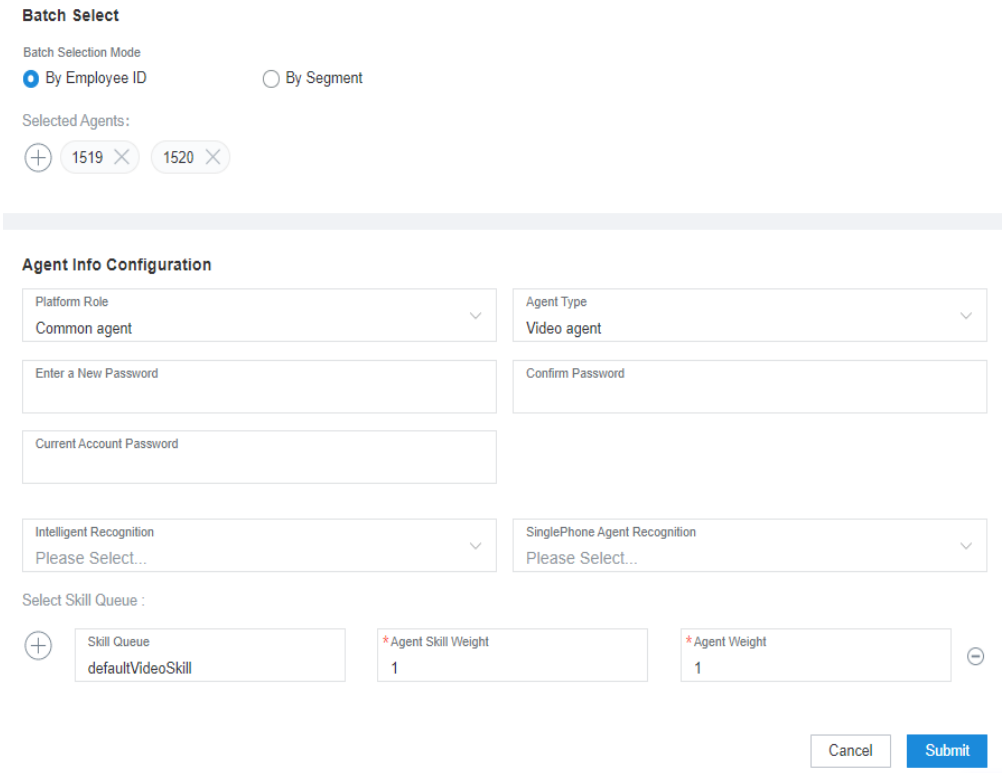
Parameter	Mandatory or Not	Description
Platform Role	Yes	Agent role. This parameter is mandatory. – **Common agent**: This role can answer or transfer inbound calls from customers. – **Quality checker**: This role can intervene in calls between common agents and customers. For example, this role can perform operations, such as insertion, interception, and forcible busy state setting, to coach and supervise agent's handling of inbound calls. – **Callout agent**: This role can answer, transfer, or reject inbound calls from customers.
Agent Type	Yes	Type of businesses that can be handled by an agent. This parameter is mandatory. – **Voice agent** – **Video agent** – **Multimedia agent** – **Versatile agent**
Agent Mobile/Fixed-Line Number	No	Mobile number or fixed-line phone number used by an agent.

Parameter	Mandatory or Not	Description
Account	Yes	Configure the created employee account. For details, see **User Guide > Tenant Administrator Guide > Managing Employees**.
Intelligent Recognition	No	Whether an agent is an intelligent agent. By default, this switch is turned off. In addition to basic voice control functions, intelligent agents support real-time transfer and related intelligent recommendation functions. Before turning on this switch, ensure that the number of agents for which intelligent recognition is enabled does not exceed the number of intelligent agents allocated when the tenant is created.
SinglePhone Agent Recognition	No	After this switch is turned on, an agent can dial a specified access code to access an IVR flow, press a key as prompted to enter the employee ID and password to sign in, and answer calls on a mobile phone. When this switch is turned on, system O&M personnel need to customize the single-phone agent process for the tenant based on the platform, and the tenant needs to provide number resources for accessing the single-phone agent process.
Agent Number Anonymization Flag	No	Flag for a third-party to mark whether an agent has the anonymization feature. This is not a feature switch. The anonymization feature enables agents to customize the calling number displayed on the user side (the calling number displayed to the user) and the calling number displayed on the agent side (the calling number displayed to the customer manager).

Parameter	Mandatory or Not	Description
Select Skill Queue	Yes	Skill queue of an agent. If multiple skill queues need to be added, ensure that the media types of all the skill queues are the same, except for versatile agents. For example, the media types of all the skill queues are voice and video, or multimedia. **NOTE** – If **Agent Type** is set to **Video Agent**, set the number of video agents allowed when applying for tenant resources. – If **Agent Type** is set to **Multimedia Agent**, set the number of multimedia agents allowed when applying for tenant resources. – If **Agent Type** is set to **Versatile Agent**, set the number of versatile agents allowed when applying for tenant resources. – To add more business accounts, choose  > **User Management** > **Employee**.

4. Click **Submit**. The business account and skill queue are associated with the agent ID.
5. (Optional) Click **Batch Configure**. On the **Batch Agent Info Configuration** page, configure agent information in batches, as shown in **Figure 4-6**.

Figure 4-6 Batch configuration



- **Batch Select**: Select agents to be configured by employee ID or employee ID segment.
- **Agent Info Configuration**: Set parameters by referring to **Step 4.2**.

----End

4.4 Integration Development

4.4.1 Importing AICC.Support.js

Step 1 Open the HTML page to be integrated into the Customer Service Cloud.

Step 2 Import **AICCSupport.js** on the page to be integrated. The following is an HTML example.

```
<script type="text/javascript" src="https://IP address:Port number or Domain name/service-cloud/resource.root/page/resource.root/js/AICCSupport.js"></script>
```

IP address:Port number indicates the IP address and port number configured when AICC is installed. *Domain name* indicates the accessible domain of the AICC.

Step 3 Develop the functions of registering listening events, removing listening events, and making one-click outbound calls.

----End

4.4.2 Developing the Function of Registering Listening Events

Step 1 Register the Init listening event.

```
AICCSupport.addHandler('Init',(eventData)=>{console.log('AICC Support Demo Page Init',eventData);});
```

Init is the value of the first parameter **eventName** of the addHandler method.

eventData is the parameter of the callback function, which is used to return event data to the integrator. The callback function is the second parameter of the addHandler method.

Table 4-4 Input parameters of the addHandler method

Parameter	Type	Name	Mandatory or Not	Description
eventName	String	Event name	Yes	Event name.
callback(eventData)	Function	Event handler	Yes	Callback function. (The **eventData** parameter is used to return data.)

Table 4-5 Event names exposed by eventName

eventName	Name	Description
Init	Workbench initialization event	Event triggered when the workbench is opened.
Ringing	Inbound call event	Event triggered when an inbound call is received.
Talking	Answering event	Event triggered when a call is answered.
Release	Release event	Event triggered when a call is released.
Switch	User switch event	Event triggered when an agent switches between multimedia users on the left of the online chat workbench.

Table 4-6 Return parameters in eventData of the callback function

Parameter	Type	Name	Description
callerNum	String	Calling number	Calling number.
calledNum	String	Called number	Called number.

Parameter	Type	Name	Description
callid	String	Session ID	Session ID. Unique ID of the current session in a call. The value must be the same as that of **Call S/N** on the **Customer Contact History** > **Contact Record** page.
mediaType	String	Media type	The options are **audio**, **video**, **wechat**, **email**, **whatsapp**, and **web**. **NOTE** The **mediaType** parameter can be returned properly only when you set **Type** and then **Subtype** for creating a multimedia skill queue.
callType	String	Call type	**0** indicates an inbound call. **1** indicates an outbound call.
workNo	String	Current agent ID	Initialization event. Only the agent ID is available.
calldata	String	Call-associated data	Call-associated data.

Step 2 Press **F12** to open the browser console. If the output log "[AICCSupport Log] -Info-addHandler - add **Init** handler" can be found, the listening event is registered successfully.

----End

4.4.3 Developing the Function of Removing Listening Events

Step 1 Remove the Init listening event.

```
AICCSupport.removeHandler('Init');
```

Init is the value of the **eventName** parameter of the removeHandler method.

Table 4-7 Input parameter of the removeHandler method

Parameter	Type	Name	Mandatory or Not	Description
eventName	String	Event name	Yes	Event name, such as the example in **Step 1**.

Step 2 Press **F12** to open the browser console. If the output log "[AICCSupport Log] -Info-removeHandler - remove **Init** handler" can be found, the listening event is removed successfully.

----End

4.4.4 Developing the Function of Making One-Click Outbound Calls

Step 1 Execute a one-click outbound call.

```
AICCSupport.clickToOutbound({mediaType: 'audio',calledNum: '12345678'});
```

audio is the value of the **mediaType** sub-attribute of the **outboundParam** object parameter of the clickToOutbound method.

12345678 is the value of the **calledNum** sub-attribute of the **outboundParam** object parameter of the clickToOutbound method.

Table 4-8 Input parameter of the clickToOutbound method

Parameter	Type	Name	Mandatory or Not	Description
outboundParam	Object	Outbound parameter	Yes	Outbound parameter.

Table 4-9 outboundParam parameters

outboundParam	Type	Name	Mandatory or Not	Description
mediaType	String	Media type	Yes	The options are **audio**, **video**, **email**, and **whatsapp**.
calledNum	String	Called number	Yes	Called number.

Step 2 Press **F12** to open the browser console. If the output log "[AICCSupport Log] -Info-clickToOutbound - execute One-click out call" can be found, the one-click outbound call function is developed successfully.

----End

4.4.5 Developing the Function of Obtaining the Identity Authentication Result

Context

The agent workbench integrates the identity authentication function. When querying information or handling some businesses for a customer, an agent needs to authenticate the customer identity, and can perform subsequent operations only after the authentication is successful.

Figure 4-7 shows a typical business process of identity authentication.

After answering a customer's call, an agent initiates an identity authentication process to the customer on the voice and video workbench. The identity authentication process is determined by the configured IVR flow. A third-party identity authentication API needs to be configured in the IVR flow. The API authenticates the customer identity based on the parameters passed in call-associated data and the entered information, and returns the authentication result to the voice and video workbench. The voice and video workbench determines whether the agent has the permission to query information or handle some businesses for the customer based on the returned authentication result.

Figure 4-7 Business process

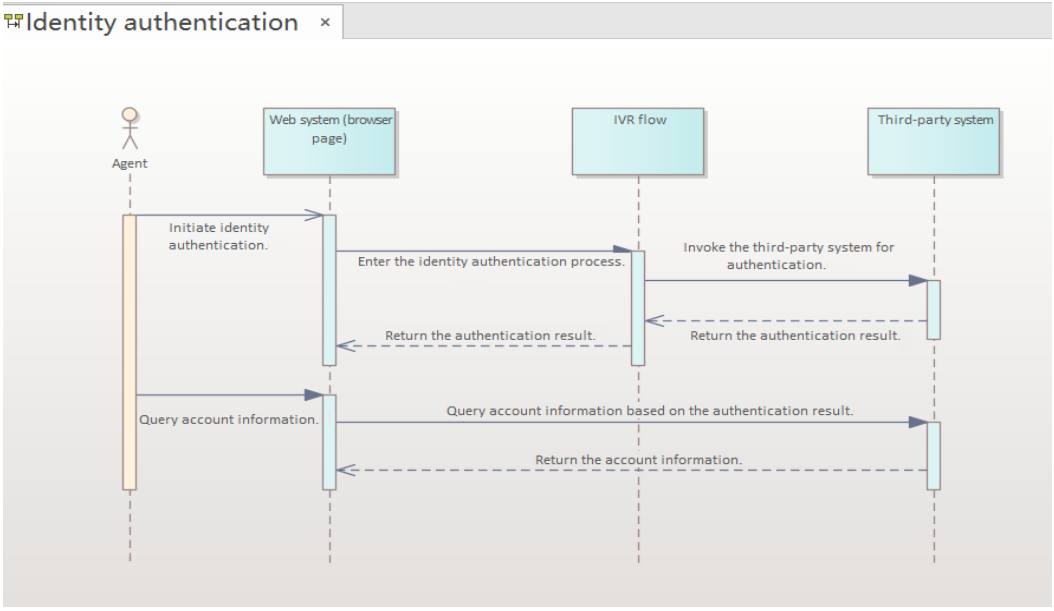
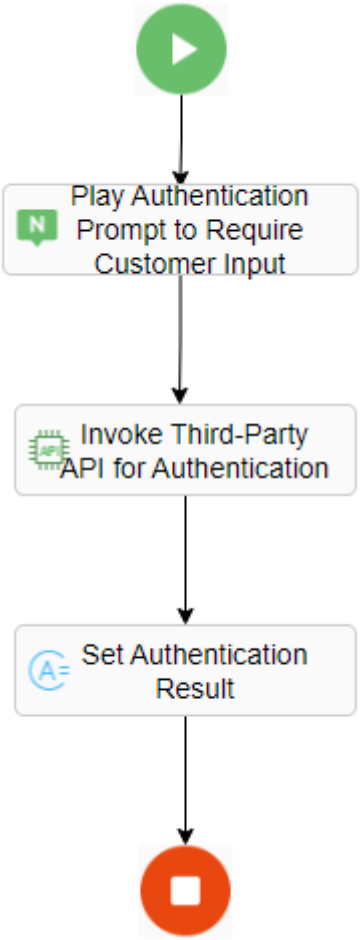


Figure 4-8 shows a typical IVR flow for identity authentication.

Figure 4-8 IVR flow



Prerequisites

1. The role has the identity authentication permission.

   The tenant administrator of a new tenant has this permission by default. The tenant administrator of an old tenant has this permission after the old tenant is upgraded using the upgrade script. A tenant administrator with this permission can assign this permission to other roles.

2. The role has the permission to configure identity authentication processes.

   The tenant administrator of a new tenant has this permission by default. The tenant administrator of an old tenant has this permission after the old tenant is upgraded using the upgrade script.

3. The role has the permission to query identity authentication records.

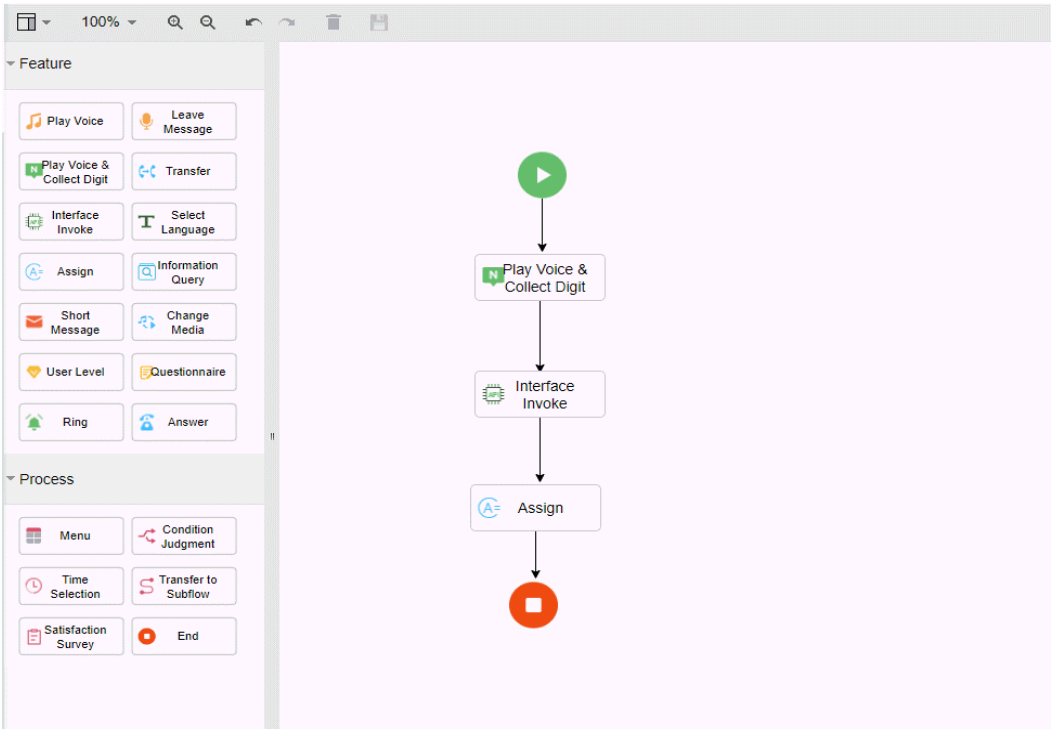
   The tenant administrator of a new tenant has this permission by default. The tenant administrator of an old tenant has this permission after the old tenant is upgraded using the upgrade script. A tenant administrator with this permission can view all identity authentication records of the tenant space by default. A tenant administrator with this permission can assign this permission to other roles.

4. An enabled identity authentication process is available.

Configuring an Identity Authentication Process

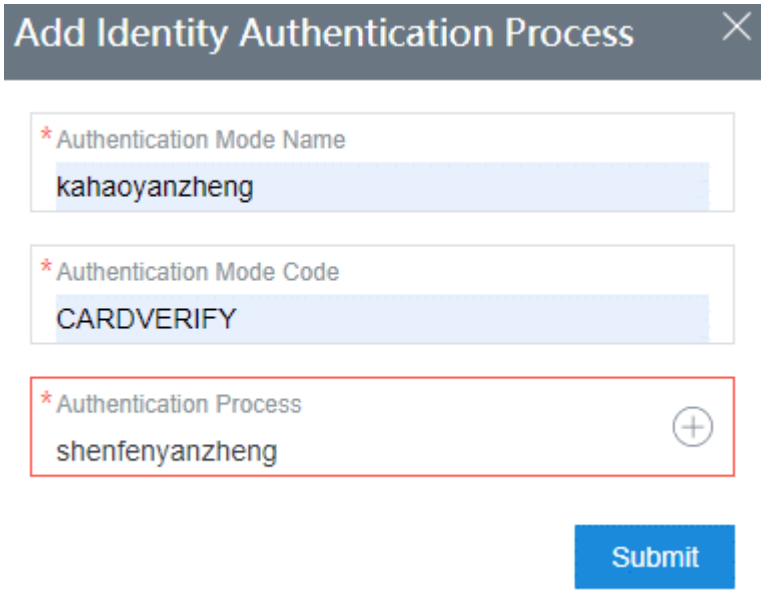
Step 1 On the flow management page, configure the IVR flow to be used by an identity authentication process.

Figure 4-9 IVR flow



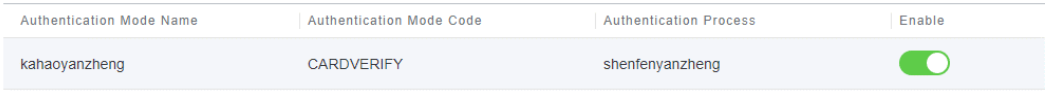
Step 2 Add an identity authentication process. Set **Authentication Process** to the IVR flow.

Figure 4-10 Identity authentication process



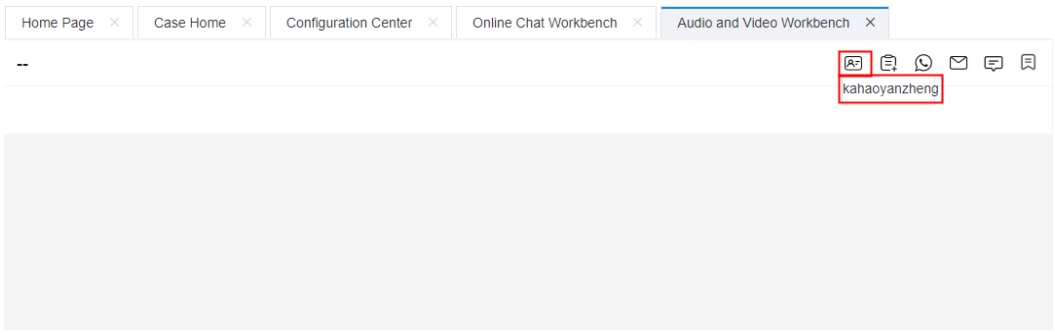
Step 3 Enable the identity authentication process.

Figure 4-11 Enabled identity authentication process



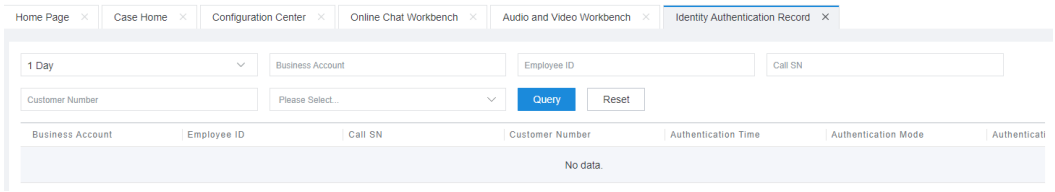
Step 4 Initiate identity authentication on the voice and video workbench.

Figure 4-12 Voice and video workbench



Step 5 Query identity authentication records.

Figure 4-13 Identity authentication records



----End

Secondary Development API

Body parameter content passed by the IVR flow to the third-party API:

{

"calldata":{ // Call-associated data

"verifyServiceNo":"88880011", // Handled number for identity authentication

"verifyWorkNo":"371" // Agent ID for identity authentication

}

}

Body parameter content returned by the third-party API:

calldata:{ // Call-associated data

"data":{

"verifyResult":0, // Identity authentication result. **0**: not returned; **1**: successful; **2**: failed; **3**: abnormal.

"verifyReturn1":"", // Identity authentication data 1

"verifyReturn2":"",//Identity authentication data 2

"verifyReturn3":"","" //Identity authentication data 3

}

}

Enterprise developers develop enterprise business system pages based on JS OpenAPIs and integrate the pages into the AICC agent workbench. On the pages, the following API can be invoked to obtain the authentication result:

AICCSupport.getVerifyResult(verifyTypeCode)

Format of the result object:

{

"verifyResult":0, // Identity authentication result. **0**: not returned; **1**: successful; **2**: failed; **3**: abnormal.

}

Obtaining the Authentication Result of an Identity Authentication Mode

AICCSupport.getVerifyResult(verifyTypeCode)

The **verifyTypeCode** parameter of the getVerifyResult method indicates the identity authentication mode for which the authentication result needs to be obtained.

Table 4-10 Input parameter of the getVerifyResult method

Parameter	Type	Meaning	Mandatory or Not	Description
verifyTypeCode	String	Identity authentication mode code	Yes	Only letters, digits, and underscores (_) are allowed.

Table 4-11 Output parameter of the getVerifyResult method

Parameter	Type	Meaning	Mandatory or Not	Description
verifyResult	String	Identity authentication result	Yes	The options are as follows: • **0**: not returned • **1**: successful • **2**: failed • **3**: abnormal An empty string indicates that the identity authentication mode code is invalid.

4.4.6 Development Example

Environment Requirement	-
Reference Library	AICCSupport.js
Download Link	**index.html**

NOTICE

- The demo described in this document may involve the use of personal data. You are advised to comply with relevant laws and regulations and take sufficient measures to ensure that personal data is fully protected.
- The demo described in this document is for demonstration only. Commercial use of the demo is prohibited.
- Information in this document is for reference only and does not constitute any offer or commitment.

index.html

```
<!DOCTYPE html>
<html>
<head>
```



```
<title>AICC Support Demo Page</title>
<meta charset="UTF-8">
<script type="text/javascript" src="AICCSupport.js"></script>
</head>
<body>
<h1>AICC Support Demo Page</h1>
<br/>
<div>
    <button id="initBtn">Remove an initialization event.</button>
    <button id="ringingBtn">Remove an inbound call event.</button>
    <button id="talkingBtn">Remove an answering event.</button>
    <button id="releaseBtn">Remove a release event.</button>
    <button id="switchBtn">Switch a multimedia user.</button>
    <button id="verifyBtn">Register an identity authentication result return event.</button>
</div>
<br/>
<div>
    <input id="outboundNum" type="text" value="88880085" />
    <input id="outboundType" type="text" value="audio" />
    <button id="audioBtn">Audio</button>
    <button id="videoBtn">Video</button>
    <button id="emailBtn">Email</button>
    <button id="whatsappBtn">WhatsApp</button>
    <button id="oneKeyOutbound">One-click outbound call</button>
</div>
<br/>
<div>
    <button id="clearMessage">Clear messages.</button>
</div>
<br/>
<div id="recMessage" style="font-size:8px;height:400px;overflow-y:auto">
</div>
</body>
<script>
    // Execute the method of opening the page.
    window.onload = function() {
        // Obtain and display the elements of calldata.
        var messageEle = document.getElementById('recMessage');

        // Open the page and directly register the Init event.
        AICCSupport.addHandler('Init',
            (eventData)=>{
                // Callback method
                // Output event data to the console.
                console.log('AICC Support Demo Page Init',data);
                // Display event data (div whose ID is recMessage) on the page.
                let oldMessage = document.getElementById('recMessage').innerHTML
                messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Initialization event '+
JSON.stringify(eventData);
            }
        );

        // Open the page and directly register the Ringing event.
        AICCSupport.addHandler('Ringing',
            (eventData)=>{
                // Callback method
                // Output event data to the console.
                console.log('AICC Support Demo Page Ringing',data);
                // Display event data (div whose ID is recMessage) on the page.
                let oldMessage = document.getElementById('recMessage').innerHTML
                messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Inbound call event '+
JSON.stringify(eventData);
            }
        );

        // Open the page and directly register the Talking event.
        AICCSupport.addHandler('Talking',
            (eventData)=>{
                // Callback method
```



```
            // Output event data to the console.
            console.log('AICC Support Demo Page Talking',data);
            // Display event data (div whose ID is recMessage) on the page.
            let oldMessage = document.getElementById('recMessage').innerHTML
            messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Answering event '+
JSON.stringify(eventData);
        }
    );

    // Open the page and directly register the Release event.
    AICCSupport.addHandler('Release',
        (eventData)=>{
            // Callback method
            // Output event data to the console.
            console.log('AICC Support Demo Page Release',data);
            // Display event data (div whose ID is recMessage) on the page.
            let oldMessage = document.getElementById('recMessage').innerHTML
            messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Release event '+
JSON.stringify(eventData);
        }
    );

    // Obtain the outbound number.
    var outboundNum = document.getElementById('outboundNum')
    // Obtain the outbound call type.
    var outboundType = document.getElementById('outboundType');
    // Listen for the audio button click event.
    var audioAndVideoBtn = document.getElementById('audioBtn');
    audioAndVideoBtn.addEventListener('click', function (e) {
        // Change the outbound call type to audio.
        outboundType.value = 'audio'
    });
    // Listen for the video button click event.
    var audioAndVideoBtn = document.getElementById('videoBtn');
    audioAndVideoBtn.addEventListener('click', function (e) {
        // Change the outbound call type to video.
        outboundType.value = 'video'
    });
    // Listen for the email button click event.
    var audioAndVideoBtn = document.getElementById('emailBtn');
    audioAndVideoBtn.addEventListener('click', function (e) {
        // Change the outbound call type to email.
        outboundType.value = 'email'
    });
    // Listen for the WhatsApp button click event.
    var audioAndVideoBtn = document.getElementById('whatsappBtn');
    audioAndVideoBtn.addEventListener('click', function (e) {
        // Change the outbound call type to whatsapp.
        outboundType.value = 'whatsapp'
    });

    // Listen for the click event of the one-click outbound call button.
    var oneKeyOutbound = document.getElementById('oneKeyOutbound');
    oneKeyOutbound.addEventListener('click', function (e) {
        var num = outboundNum.value;
        var type = outboundType.value;
        // Execute a one-click outbound call.
        AICCSupport.clickToOutbound({mediaType: type,calledNum: num})
    });

    // Listen for the click event of the button for registering or removing initialization events.
    var initBtn = document.getElementById('initBtn');
    initBtn.addEventListener('click', function (e) {
        var val = initBtn.innerHTML;
        // Remove an event.
        if(val === 'Remove an initialization event.'){
            initBtn.innerHTML = 'Register an initialization event.';
            AICCSupport.removeHandler('Init');
        }
```



```
        // Register an event.
        if(val === 'Register an initialization event.'){
            initBtn.innerHTML = 'Remove an initialization event.';
            AICCSupport.addHandler('Init',
                (eventData)=>{
            // Callback method
            // Output event data to the console.
                    console.log('AICC Support Demo Page Init',data);
            // Display event data (div whose ID is recMessage) on the page.
                    let oldMessage = document.getElementById('recMessage').innerHTML
            messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Initialization event '+
JSON.stringify(eventData);
                }
            );
        }
    });

    // Listen for the click event of the button for registering or removing an inbound call event.
    var ringingBtn = document.getElementById('ringingBtn');
    ringingBtn.addEventListener('click', function (e) {
        var val = ringingBtn.innerHTML;
        // Remove an event.
        if(val === 'Remove an inbound call event.'){
            ringingBtn.innerHTML = 'Register an inbound call event.';
            AICCSupport.removeHandler('Ringing');
        }
        // Register an event.
        if(val === 'Register an inbound call event.'){
            ringingBtn.innerHTML = 'Remove an inbound call event.';
            AICCSupport.addHandler('Ringing',
                (eventData)=>{
            // Callback method
            // Output event data to the console.
                    console.log('AICC Support Demo Page Ringing',data);
            // Display event data (div whose ID is recMessage) on the page.
                    let oldMessage = document.getElementById('recMessage').innerHTML
            messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Inbound call event '+
JSON.stringify(eventData);
                }
            );
        }
    });

    // Listen for the click event of the button for registering or removing an answering event.
    var talkingBtn = document.getElementById('talkingBtn');
    talkingBtn.addEventListener('click', function (e) {
        var val = talkingBtn.innerHTML;
        // Remove an event.
        if(val === 'Remove an answering event.'){
            talkingBtn.innerHTML = 'Register an answering event.';
            AICCSupport.removeHandler('Talking');
        }
        // Register an event.
        if(val === 'Register an answering event.'){
            talkingBtn.innerHTML = 'Remove an answering event.';
            AICCSupport.addHandler('Talking',
                (eventData)=>{
            // Callback method
            // Output event data to the console.
                    console.log('AICC Support Demo Page Talking',data);
            // Display event data (div whose ID is recMessage) on the page.
                    let oldMessage = document.getElementById('recMessage').innerHTML
            messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Answering event '+
JSON.stringify(eventData);
                }
            );
        }
    });
```

```
        // Listen for the click event of the button for registering or removing a release event.
        var releaseBtn = document.getElementById('releaseBtn');
        releaseBtn.addEventListener('click', function (e) {
            var val = releaseBtn.innerHTML;
            // Remove an event.
            if(val === 'Remove a release event.'){
                releaseBtn.innerHTML = 'Register a release event.';
                AICCSupport.removeHandler('Release');
            }
            // Register an event.
            if(val === 'Register a release event.'){
                releaseBtn.innerHTML = 'Remove a release event.';
                AICCSupport.addHandler('Release',
                    (eventData)=>{
                // Callback method
                // Output event data to the console.
                        console.log('AICC Support Demo Page Release',data);
                // Display event data (div whose ID is recMessage) on the page.
                        let oldMessage = document.getElementById('recMessage').innerHTML
                messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Release event '+
JSON.stringify(eventData);
                    }
                );
            }
        });

        // Listen for the click event of the button for registering or removing a multimedia user switch event.
        var switchBtn = document.getElementById('switchBtn');
        switchBtn.addEventListener('click', function (e) {
            var val = switchBtn.innerHTML;
            // Remove an event.
            if(val === 'Remove a multimedia user switch event.'){
                switchBtn.innerHTML = 'Register a multimedia user switch event.';
                AICCSupport.removeHandler('Switch');
            }
            // Register an event.
            if(val === 'Register a multimedia user switch event.'){
                switchBtn.innerHTML = 'Remove a multimedia user switch event.';
                AICCSupport.addHandler('Switch',
                    (eventData)=>{
                        console.log('AICC Support Demo Page Switch',data);
                        // Display event data (div whose ID is recMessage) on the page.
                        let oldMessage = document.getElementById('recMessage').innerHTML
                        messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Multimedia user switch event
'+ JSON.stringify(eventData);
                    }
                );
            }
        });

        // Listen for the click event of the button for registering or removing an identity authentication result
return event.
        var verifyBtn = document.getElementById('verifyBtn');
        verifyBtn.addEventListener('click', function (e) {
            var val = verifyBtn.innerHTML;
            if(val === 'Remove an identity authentication result return event.'){
                verifyBtn.innerHTML = 'Register an identity authentication result return event.';
                AICCSupport.removeHandler('IvrReturn');
            }
            if(val === 'Register an identity authentication result return event.'){
                verifyBtn.innerHTML = 'Remove an identity authentication result return event.';
                AICCSupport.addHandler('IvrReturn',
                    (data)=>{
                        console.log('AICC Support Demo Page IvrReturn',data);
                        let oldMessage = document.getElementById('recMessage').innerHTML
                        messageEle.innerHTML = oldMessage +'<br/>'+new Date()+' Identity authentication result
return event '+ JSON.stringify(data).replace(/\\/g, '');
                    }
                );
```



```
        }
    });

    // Listen for the click event of the message clearing button.
    var clearMessage = document.getElementById('clearMessage');
    clearMessage.addEventListener('click', function (e) {
        // Clear the display content (div whose ID is recMessage).
        document.getElementById('recMessage').innerHTML = '';
    });
}
</script>
</html>
```

4.4.7 Post-development Configuration

4.4.7.1 Configuring the Address Trustlist

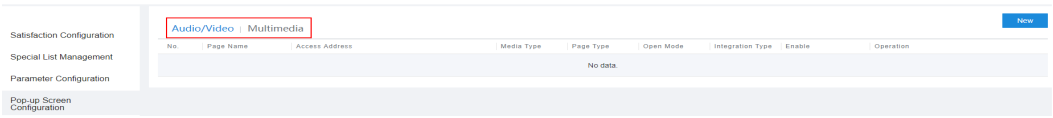
Provide the URL of the developed page to Huawei O&M personnel for them to add it to the address trustlist on the **Call Center Management** > **Address Whitelist Management** page.

4.4.7.2 Configuring a Screen Pop-up

Step 1 Choose **Configuration Center** > **Expansion and Integration Management** > **Screen Pop-up**.

Step 2 In the upper left corner, click the **Audio/Video** or **Multimedia** tab. For details, see **Figure 4-14**.

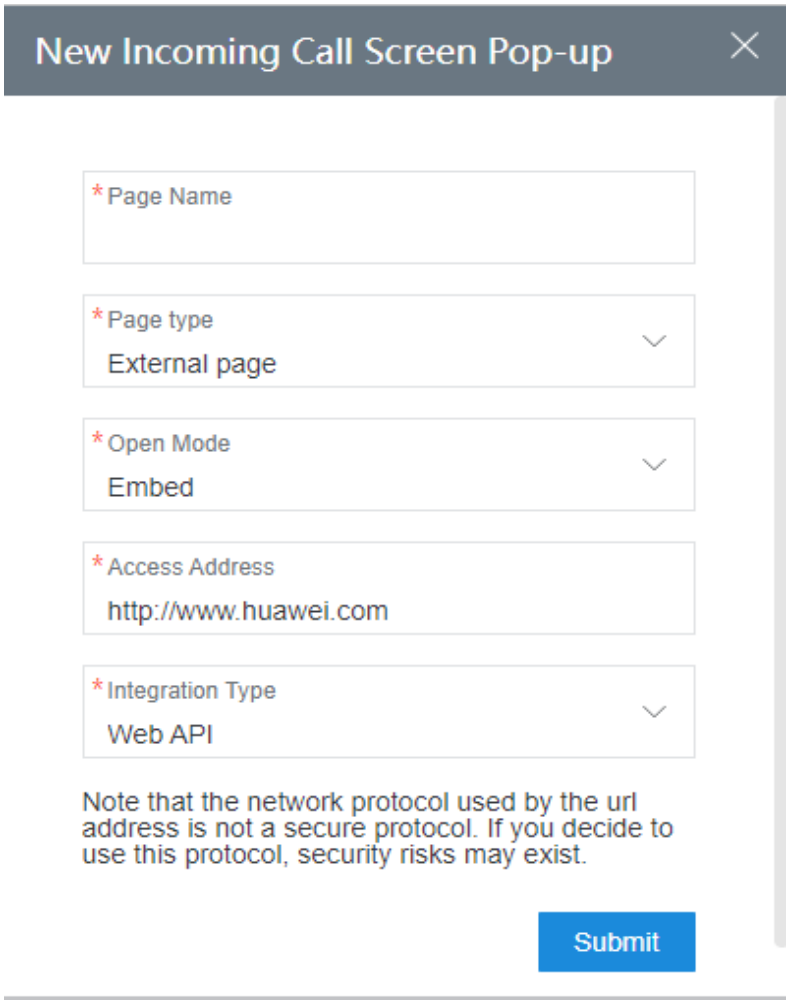
Figure 4-14 Clicking a screen pop-up tab



Step 3 Configure a screen pop-up.

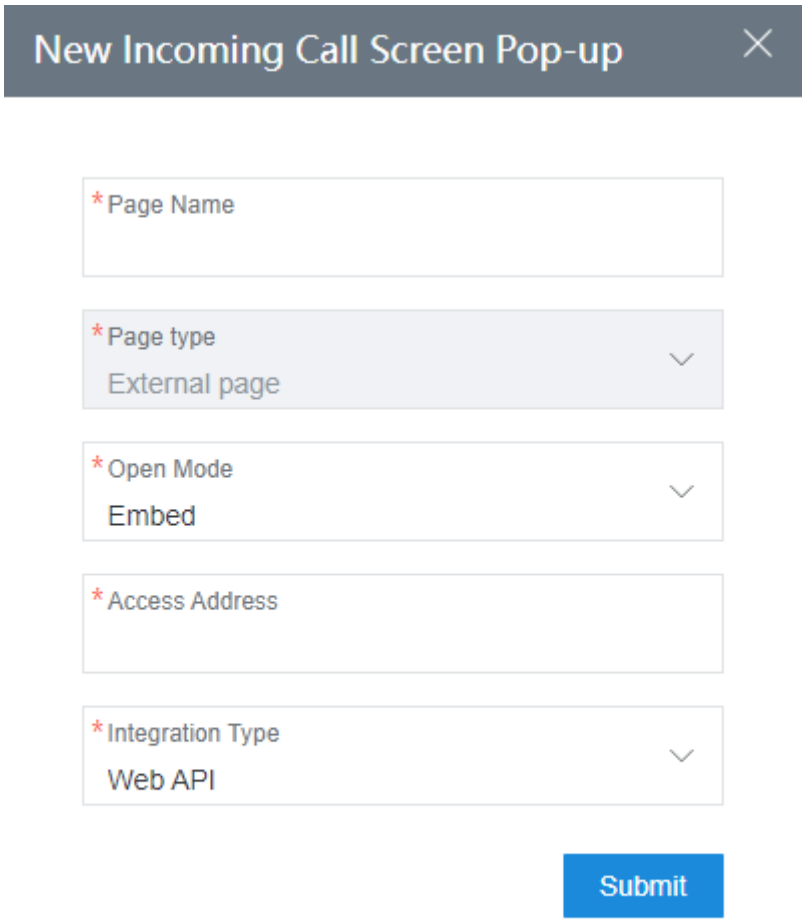
1. If you use a voice or video agent, click **New** on the **Audio/Video** tab page, and set **Page Type** to **External page**, **Open Mode** to **Embed**, and **Integration Mode** to **Web API**, as shown in **Figure 4-15**.

Figure 4-15 Audio/Video screen pop-up configuration page



2. If you use a multimedia agent, click **New** on the **Multimedia** tab page, and set **Open Mode** to **Embed** and **Integration Mode** to **Web API**, as shown in **Figure 4-16**.

Figure 4-16 Multimedia screen pop-up configuration page



NOTE

Registering listening events, removing listening events, and the event handler (callback function) depend on the audio and video workbench or online chat workbench. The integrated screen pop-up must be an external page that is embedded in the workbench and integrated by a web API. In other cases, JavaScript integration is not supported.

Step 4 After the screen pop-up is configured, turn on the switch in the **Enable** column.

Step 5 If the agent and skill queue are of the voice and video type, choose **Audio and Video Workbench** to access the screen pop-up.

NOTE

The one-click voice and video outbound call function depends on the connection bar on the top. The function can be used only when a voice or video skill queue is available and an agent has signed in. The one-click voice and video outbound call function can be used on the screen pop-ups of the audio and video workbench and online chat workbench.

If the agent and skill queue are of the multimedia type, choose **Online Chat Workbench** to access the screen pop-up.

NOTE

The multimedia one-click outbound call function supports only email outbound calls and WhatsApp outbound calls. The function depends on the online chat workbench and can be used only when a multimedia skill queue is available and an agent has signed in. Email outbound calls and WhatsApp outbound calls cannot be used on the screen pop-ups of the audio and video workbench.

----End

4.5 Test and Verification

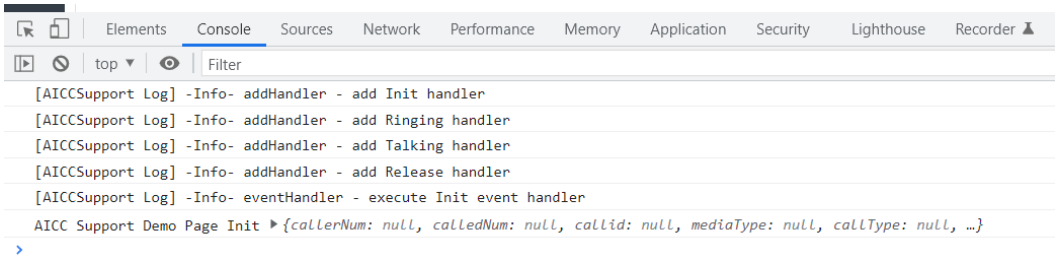
4.5.1 Registration of Listening Events and Execution Verification of the Event Handler

After page integration is complete, you need to test and verify whether the functions of registering listening events and removing listening events, the callback function, and the one-click outbound call function are normal. The following uses the Chrome browser as an example. During the verification, you can use the Nginx server to simulate a third party to invoke the basic page or use other access methods that you are familiar with.

Step 1 Enable the Nginx server on the local PC (for details about the Nginx version, see **nginx/Windows-1.22.0**), and configure the integration page in the HTML file in the **nginx** directory. You can also configure the path of the integration page in the **nginx.conf** file.

Step 2 Open the AICC page, add the integration page access address configured in Step 1 to the address trustlist by referring to **4.4.7.1 Configuring the Address Trustlist**, and configure a screen pop-up by referring to **4.4.7.2 Configuring a Screen Pop-up**.

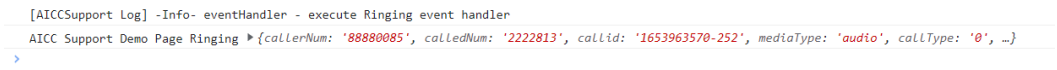
Step 3 Open the audio and video workbench and register an event on the integration page. (The page demonstrated here automatically registers the corresponding event when it is opened. Therefore, the Init event handler is executed as soon as the page is opened.)



The AICC Support Demo Page Init log is output by the event handler that registers a listening event. For details, see **Registering Listening Events**.

Step 4 After signing in as an agent and registering a ringing listening event, dial the configured called number through the OpenEye client (for details, see **OpenEye Installation Guide**).

Step 5 When receiving a call, check that the integration page executes the ringing event handler and returns information.



The AICC Support Demo Page Ringing log is output by the event handler that registers a listening event. For details, see **Registering Listening Events**.

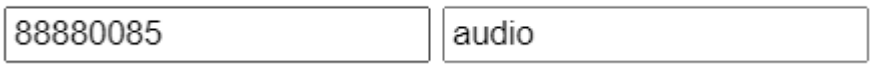
----End

4.5.2 Verification of the One-Click Outbound Call Function

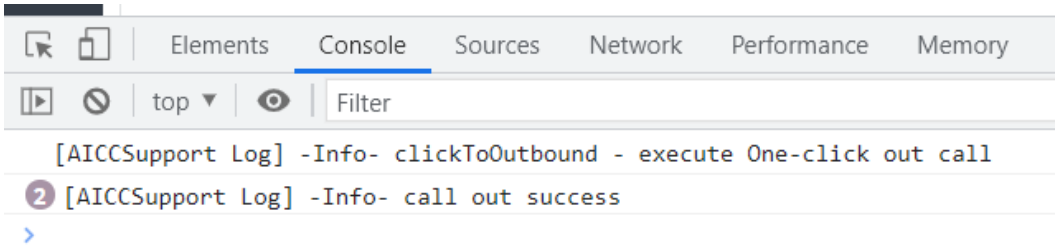
Step 1 Enable the Nginx server on the local PC (for details about the Nginx version, see **nginx/Windows-1.22.0**), and configure the integration page in the HTML file in the **nginx** directory. You can also configure the path of the integration page in the **nginx.conf** file.

Step 2 Open the AICC page, add the integration page access address configured in Step 1 to the address trustlist by referring to **4.4.7.1 Configuring the Address Trustlist**, and configure a screen pop-up by referring to **4.4.7.2 Configuring a Screen Pop-up**.

Step 3 After signing in as an agent and opening the audio and video workbench, enter the outbound number and outbound call type on the integration page and invoke the one-click outbound call method.



Step 4 Check that the one-click outbound call is initiated successfully.



----End

4.5.3 Verification of the Page Integration

Step 1 Enable the Nginx server on the local PC (for details about the Nginx version, see **nginx/Windows-1.22.0**), and configure the integration page in the HTML file in the **nginx** directory. You can also configure the path of the integration page in the **nginx.conf** file.

Step 2 Open the AICC page, add the integration page access address configured in **Step 1** to the address trustlist, and configure a screen pop-up.

Step 3 Configure an identity authentication process.

Step 4 Sign in as an agent, open the voice and video workbench, and initiate the identity authentication process during a call.

Step 5 Enter the identity authentication mode code on the integration page and invoke the API for obtaining the identity authentication result.

Step 6 The API for obtaining the identity authentication result returns a success response.

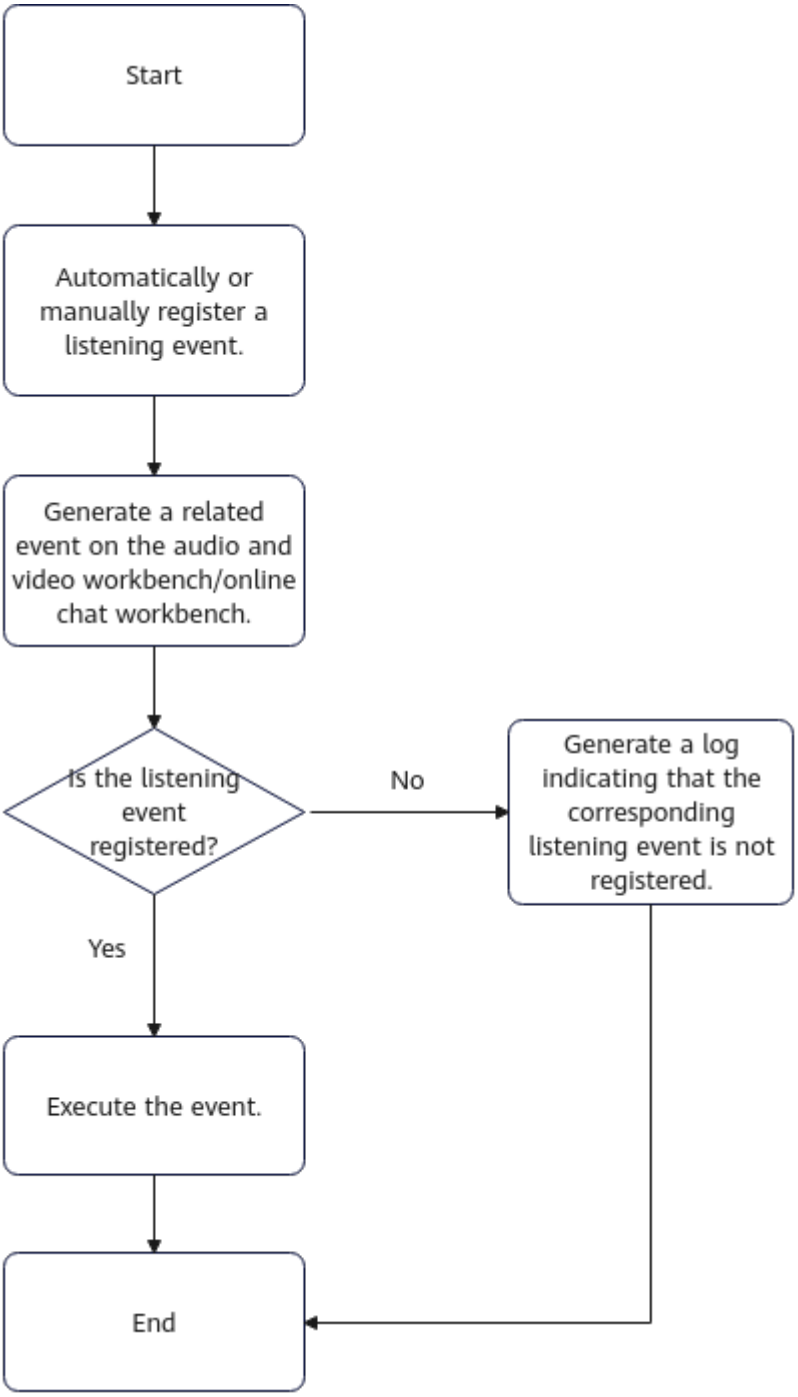
----End

4.6 Appendix - Process Description

4.6.1 Overall Working Process

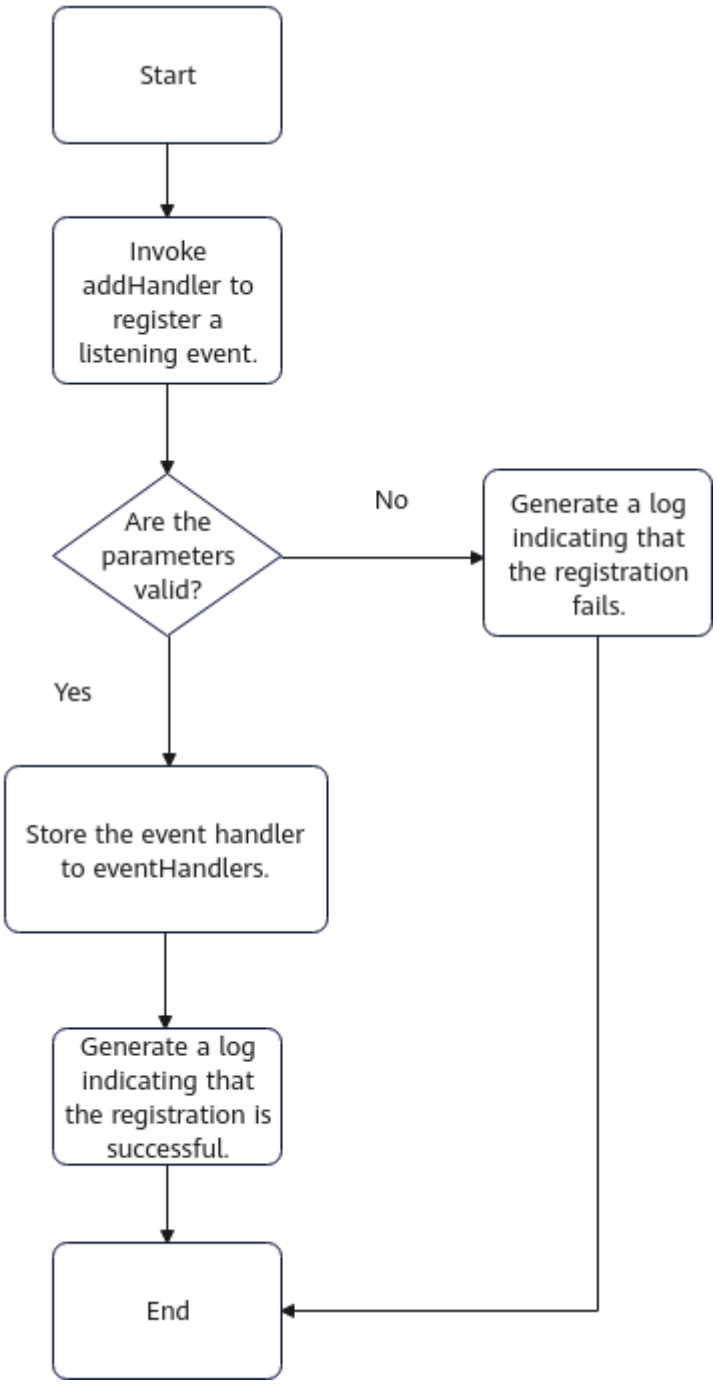
If you want to know the working process of each event or function, read the following sections.

Figure 4-17 Working process



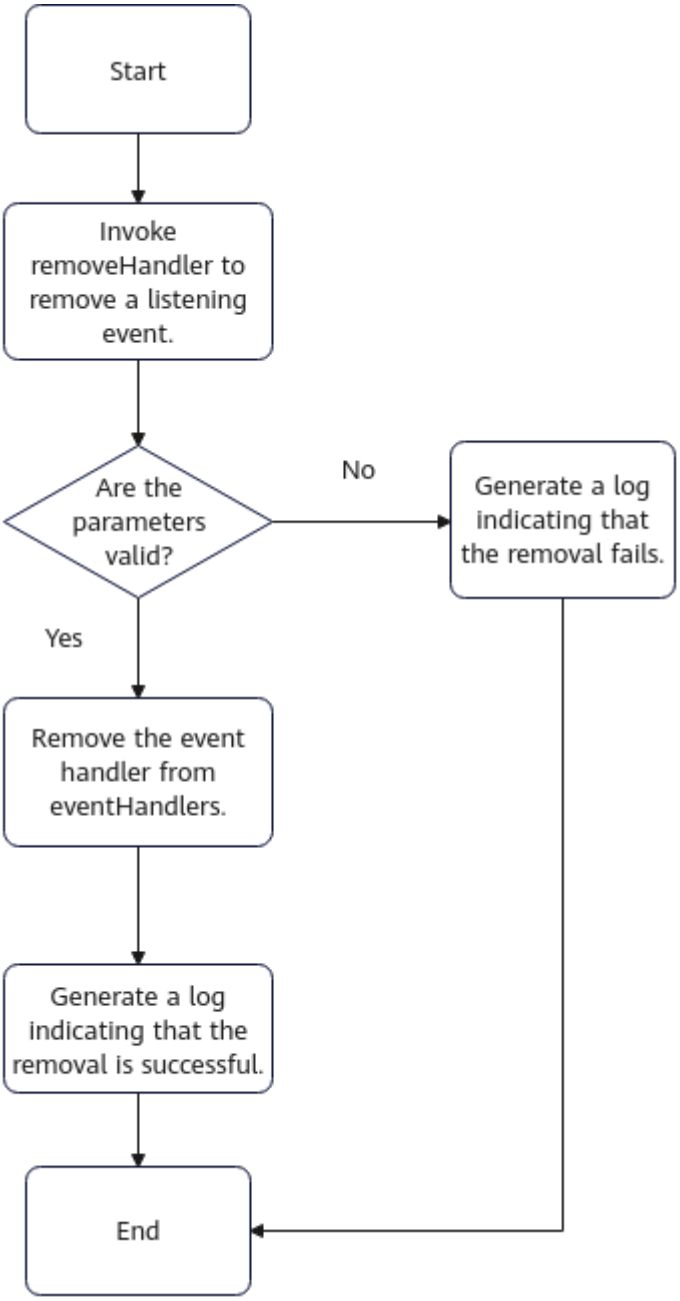
4.6.2 Process of Registering a Listening Event (addHandler)

Figure 4-18 Registration process



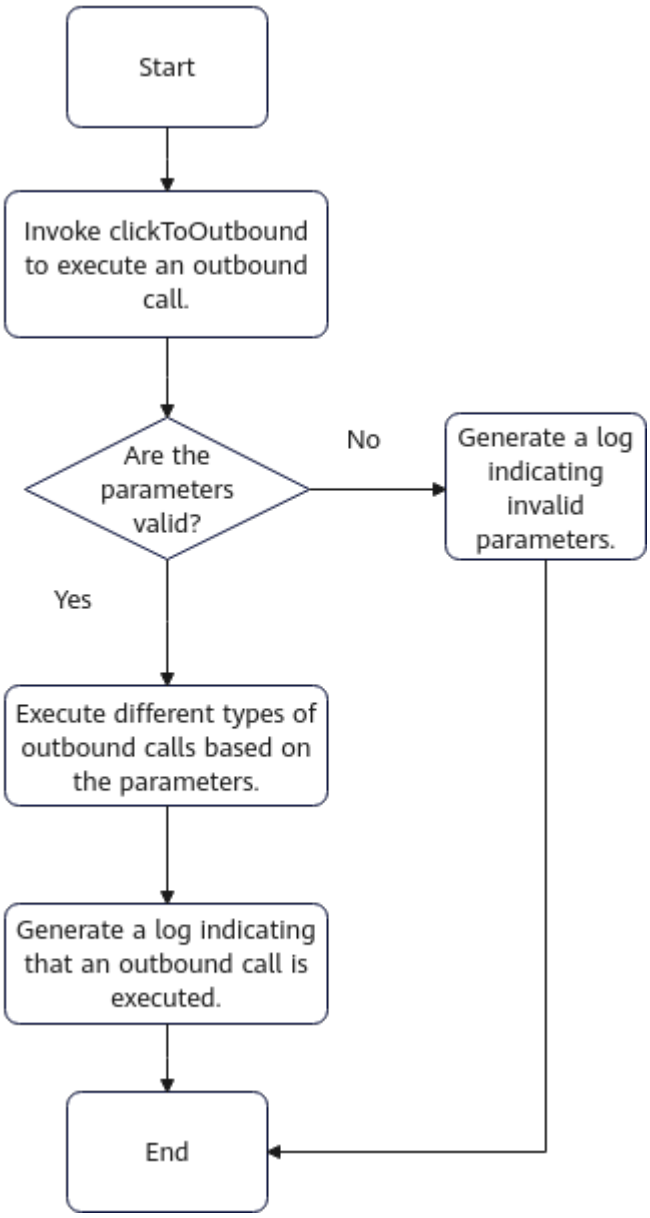
4.6.3 Process of Removing a Listening Event (removeHandler)

Figure 4-19 Removal process



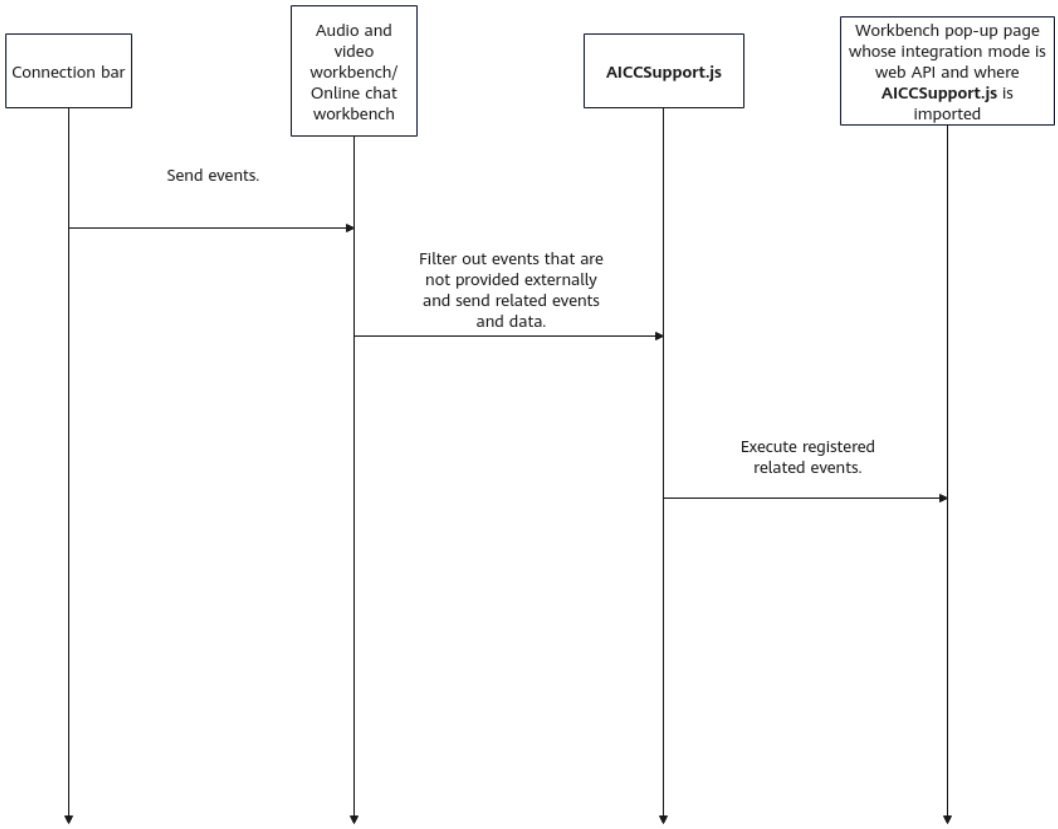
4.6.4 Process of Executing a One-Click Outbound Call (clickToOutbound)

Figure 4-20 Process of executing a one-click outbound call



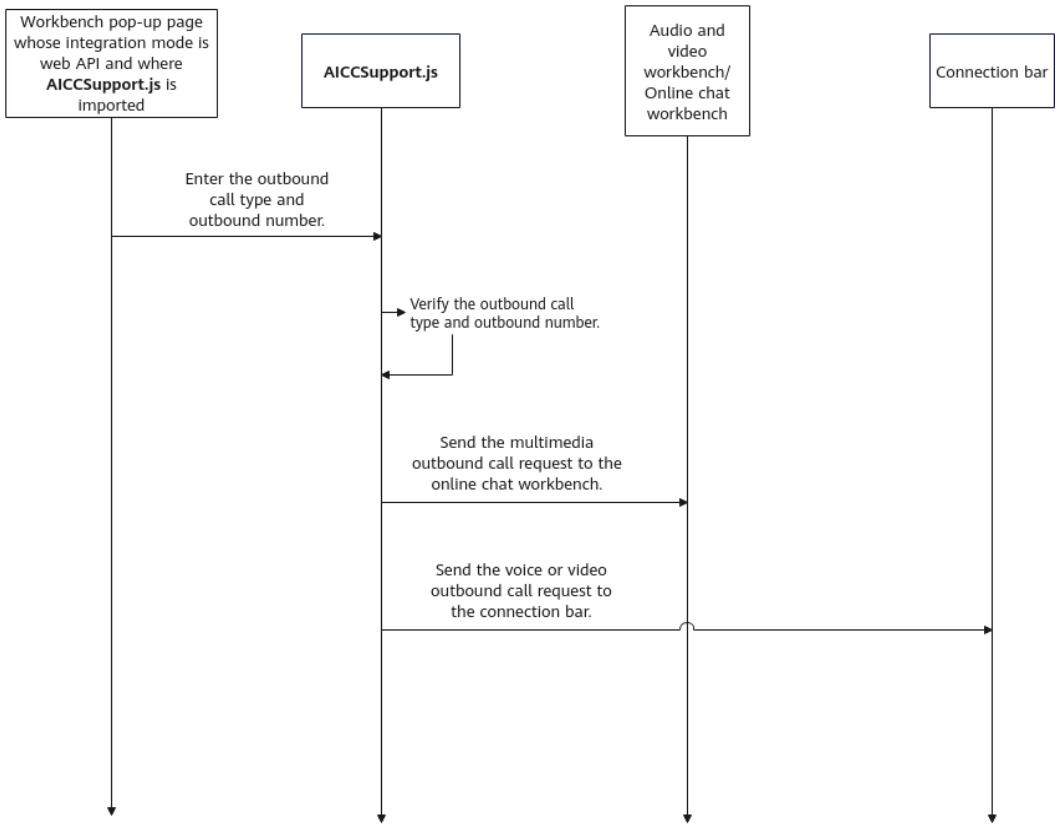
4.6.5 Process of Executing the Handling Method of a Registered Event

Figure 4-21 Process of executing the handling method of a registered event



4.6.6 Process of Executing the One-Click Outbound Call Method

Figure 4-22 Process of executing the one-click outbound call method



4.6.7 Process of Obtaining the Identity Authentication Result

Figure 4-23 Business process of obtaining the identity authentication result

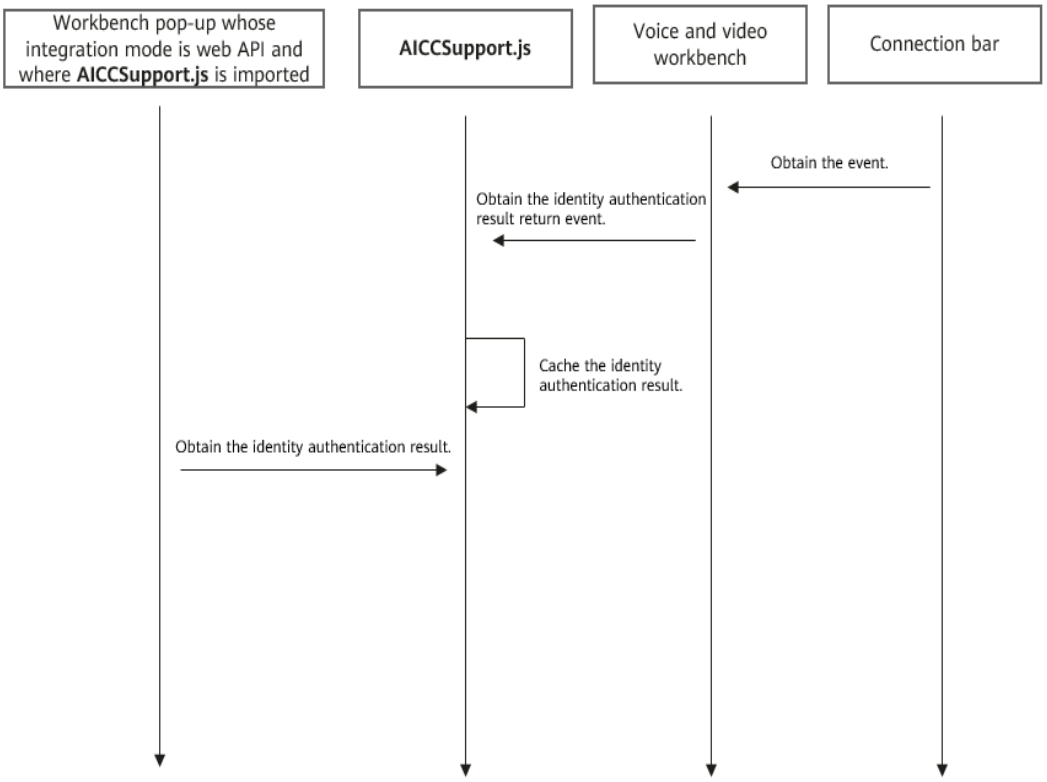
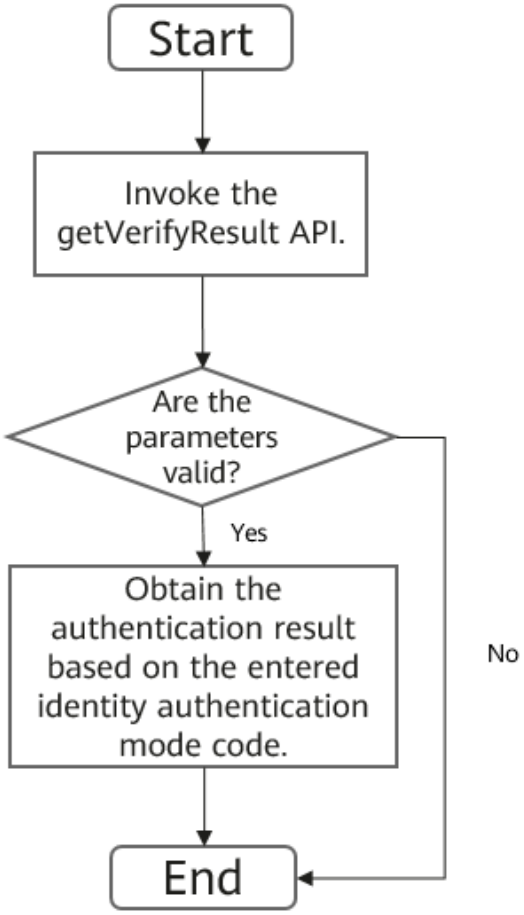


Figure 4-24 Process of invoking the API for obtaining the identity authentication result



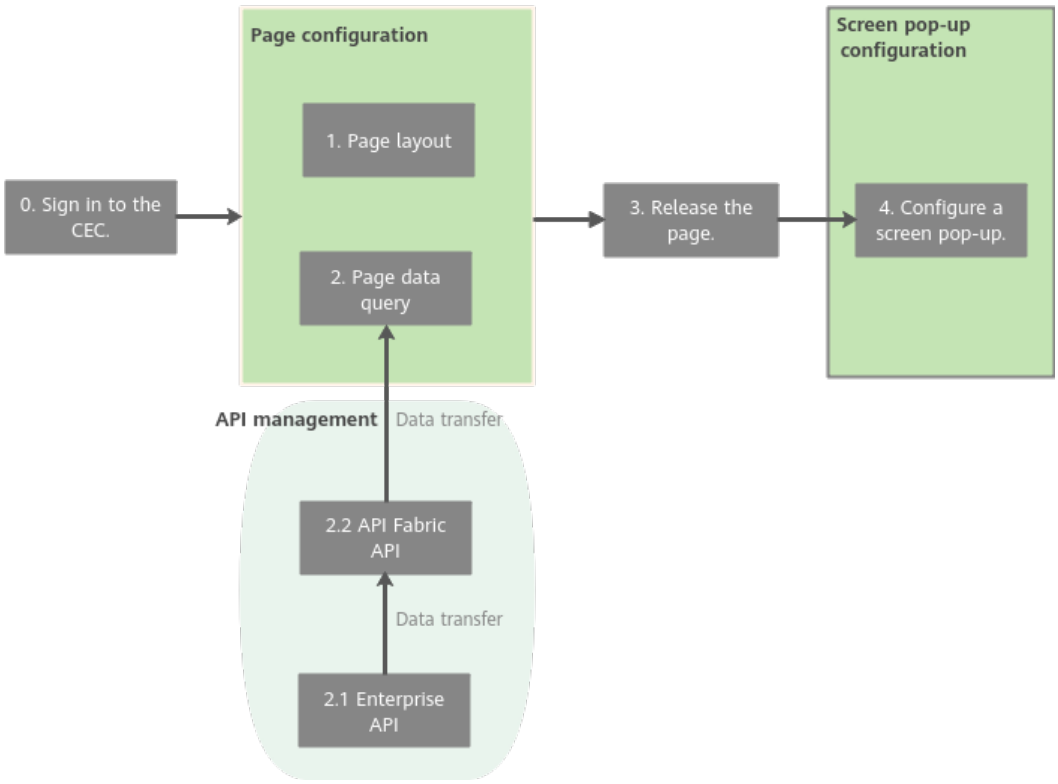
5 Scenario 4: Integration by Invoking an Enterprise API

5.1 Overview

When an enterprise directly uses the CEC agent framework as the call center system, the system can open a customized page and use the API provided by the third party to obtain information (for example, customer information) when an agent answers a call. This section describes the configuration.

The customized page embedded in the agent framework is implemented based on the screen pop-up configuration. **Figure 5-1** shows the integration principle.

Figure 5-1 Integration principle



5.2 Scenario Description

When answering a call, an agent in the enterprise call center invokes the customer information query interface provided by the enterprise to open the customer details page of the calling number (customer).

5.3 API Description

Version

V1.0.0

Function

This API is used to query customer information.

Method

The request method must be set to POST. This API supports only POST and does not support PUT, GET, DELETE, or other methods.

URL

Example: /rest/v1/queryCustomer

In the URL, v1 indicates the API version.

Request

- Request header

Content-Type:application/json; charset=UTF-8,

Connection:keep-alive

- Request body

The following provides an example of the request body of this API:

```
{
 "param":"152****1028"
}
```

Table 5-1 describes the parameter in the request body of this API.

Table 5-1 Request body parameter

Parameter	Type	Mandatory or Not	Description
param	String	Yes	Third-party query parameters.

Response

The following provides an example of the response body of this API:

```
[
  {
    "objectName": "Customer information",
    "values": [
      {
        "valueNo": "1",
        "length": "128",
        "type": "1",
        "value": "zengxiangyu01",
        "isMandatory": true,
        "name": "Name",
      },
      {
        "valueNo": "2",
        "dictItems": [
          {
            "value": "Prepaid",
            "key": "1"
          },
          {
            "value": "Postpaid",
            "key": "2"
          }
        ],
        "type": "5",
        "value": "1",
        "isMandatory": true,
        "name": "Payment mode",
      }]
  }
]
```

Table 5-2 describes the parameters in the response body of this API.

Table 5-2 Response body parameters

Parameter	Type	Mandatory or Not	Description
objectName	String	Yes	Returned information description.
values	Array	Yes	Returned information values. For details about the parameters, see **Table 5-3**.

Table 5-3 values parameters

Parameter	Type	Mandatory or Not	Description
valueNo	String	No	Sequence number.
name	String	Yes	Attribute name.
value	Object	Yes	Attribute value.
type	String	Yes	Type. The options are as follows: • **1**: string • **2**: number • **3**: date • **4**: time • **5**: data dictionary
length	String	No	Length. The maximum length is 256 characters.
isMandatory	Boolean	Yes	Whether an attribute is mandatory.
maskStyle	String	No	Mask display type. The options are as follows: • **1**: The last four digits are displayed. • **2**: The first four digits are displayed. If this parameter is empty or not transferred, all digits are displayed.
dictItems	Array	No	Dictionary items. This parameter is mandatory when **type** is set to **5**. For details about the parameters, see **Table 5-4**.

Table 5-4 dictItems parameters

Parameter	Type	Mandatory or Not	Description
key	String	Yes	Key.
value	String	Yes	Value.

Error Codes

None

5.4 Configuring a Page

Procedure

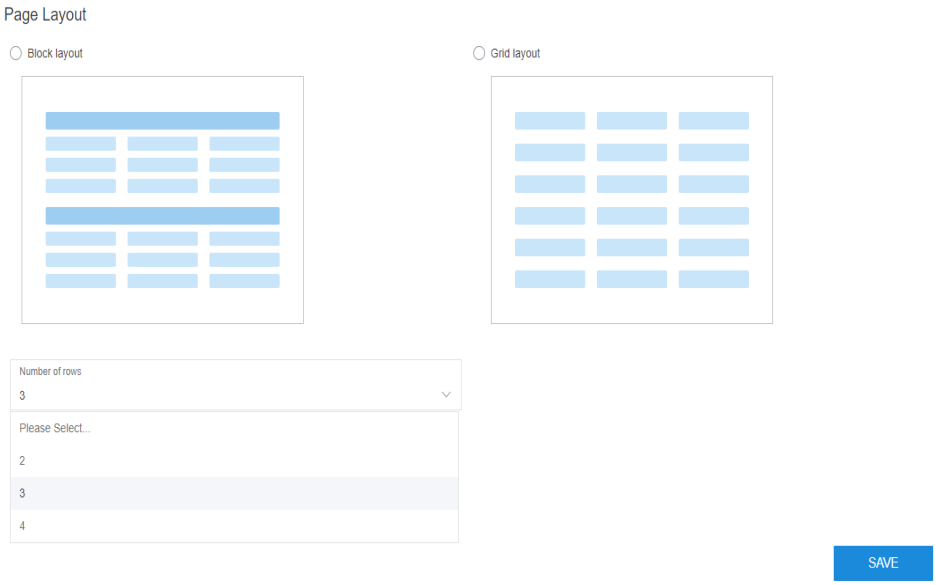
Step 1 Apply to the operations administrator for adding the preceding API to the dictionary value mapping for a specific tenant to use.

Step 2 Sign in to the CEC as a tenant administrator and configure a customized page.

1. Choose **Configuration Center** > **Expansion and Integration Management** > **Page Configuration**.
2. Click  to create a page. Enter a page name and click **OK**.

   Page Name: The value can contain a maximum of 256 characters and must be unique.
3. Configure the page layout.
   a. Click **Page Layout**.
   b. Select a layout type, as shown in **Figure 5-2**.

Figure 5-2 Page Layout



c. Click **SAVE**.

4. Configure the page API.
   a. Click **Page interface**. Select **External interface**.

      The external API needs to be configured on the **API Management** page. Contact the system operations administrator.

   b. Click **Save**.
5. Click the name of the customized page. In the **Page Status** area, click to release the page.

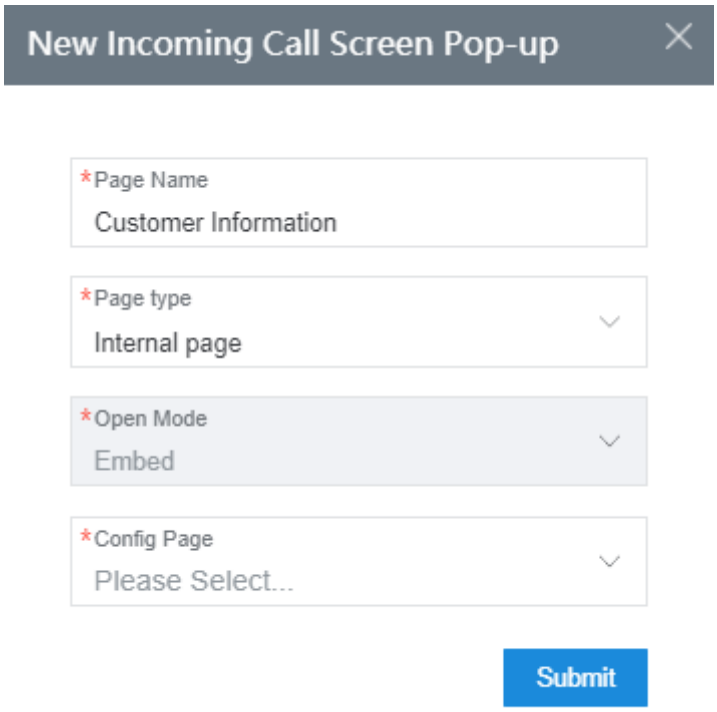
Step 3 Sign in to the CEC as a tenant administrator and configure a screen pop-up.

1. Choose **Configuration Center** > **Expansion and Integration Management** > **Screen Pop-up**.
2. Click the **Audio/Video** tab and configure a screen pop-up of the corresponding type. Multimedia screen pop-ups are not supported.

   Screen pop-ups can be integrated into the corresponding agent workbenches. For example, an **Audio/Video** screen pop-up can be integrated into the **Audio and Video Workbench**.
3. Click **New** in the upper right corner. The page shown in **Figure 5-3** is displayed.
4. Enter the page name for the screen pop-up, set **Page type** to **Internal page**, set **Config Page** to the customized page created in **2**, and click **Submit**. In the **Successful** dialog box that is displayed, click **OK**.

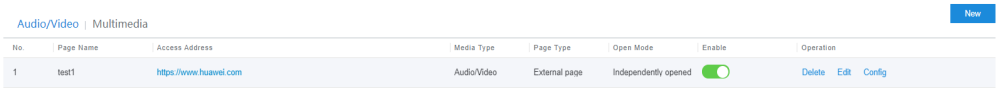
   A maximum of five screen pop-ups can be configured in total for the **Audio/Video** and **Multimedia** types.

Figure 5-3 Configuring an inbound call screen pop-up



5. In the inbound call screen pop-up list, enable the new page, as shown in **Figure 5-4** (green button).

   Figure 5-4 Enabling the page

   If multiple inbound call screen pop-ups are enabled, click **Up** or **Down** in the **Operation** column to adjust the pop-up sequence of a screen pop-up. The screen pop-up on the top of the list pops up first.

6. Sign out the agent and sign in again for the configuration to take effect.

----End

5.5 Test and Verification

Prerequisites

You have enabled the voice and video agent function. For details, see **4.3.1 Enabling the Voice, Video, or Multimedia Agent Function in the Customer Service Cloud**.

Procedure

Step 1 Sign in to the CEC, choose **Configuration Center** > **Access Configuration** > **Called Route**, and view access codes.

Step 2 Click 

on the top of the page to sign in an agent.

Step 3 Dial an access code using a test number.

Step 4 After the call is automatically connected, verify that the configured page can be automatically opened in the agent framework and can display customer information queried using the customer API.

----End

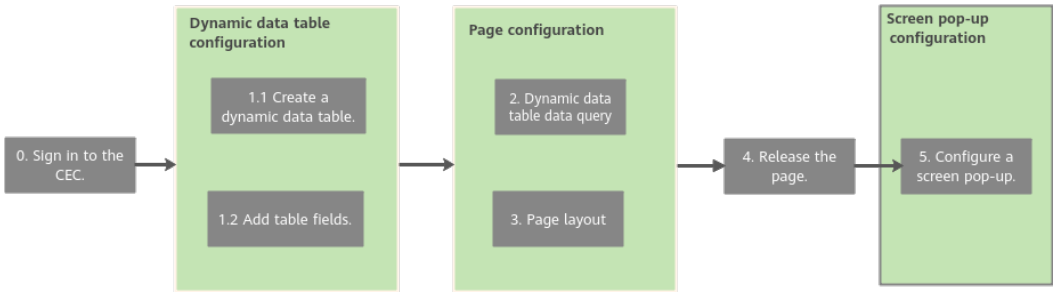
6 Scenario 5: Integration by Recording Information to the CEC

6.1 Overview

When an enterprise directly uses the CEC agent framework as the call center system, the system can open a customized page (for example, page for viewing customer information) when an agent answers a call. This section describes the configuration.

The customized page embedded in the agent framework is implemented based on the screen pop-up configuration. **Figure 6-1** shows the integration principle.

Figure 6-1 Integration principle



6.2 Scenario Description

An enterprise has a small amount of customer data and stores the data in the CEC. When answering a call, an agent in the call center can open the customer details page of the calling number (customer).

6.3 Configuring a Dynamic Data Table

Procedure

Step 1 Sign in to the CEC as a tenant administrator and configure a dynamic data table.

1. Choose **Configuration Center** > **Expansion Management** > **Dynamic Data Table**.
2. Click **New** in the upper right corner. In the dialog box that is displayed, enter the table name and description based on the data provided by the enterprise to create a dynamic data table.

   A maximum of five dynamic data tables can be created.
3. Click **Details** to add fields to the dynamic data table.
4. Click **New** in the upper right corner.
5. Set **Field Name**, **Used as index**, **Encrypted**, **Mandatory**, and **Field Type**. Depending on the value of **Field Type**, you may also need to set **Field length**, **Masked**, **Mask style**, or **Value range**. For details about the parameters, see **Configuring a Dynamic Data Table**.

   NOTE

   The **Field length**, **Masked**, and **Mask style** parameters are displayed only when **Field Type** is set to **String** or **Number**.

   The **Value range** parameter is displayed only when **Field Type** is set to **Dictionary**.
6. Click **Submit**.

   A maximum of one index field, 30 common fields, and 20 encrypted fields are allowed in a dynamic data table.
7. Return to the dynamic data table list, and click **Release** in the **Operation** column corresponding to the dynamic data table for which fields have been configured to release it.

Step 2 Sign in to the CEC as a tenant administrator and configure data for the dynamic data table.

1. Choose **Configuration Center** > **Expansion Management** > **Table Data**.
2. Click ∨ to select a released dynamic data table.

3. Configure table data.
   a. Click **Export Template** to download a template file. Set customer information in the template.

NOTE

- Only CSV files can be uploaded. The file size cannot exceed 20 MB. The file name can contain only Chinese characters, letters, digits, and underscores (_).
- The header of the uploaded CSV file must match that of the downloaded template file. Otherwise, the file processing fails.

b. Click **Import Data**. In the dialog box that is displayed, click to select the configured template, and click **Submit**.

c. Click **Import Result View**. On the page that is displayed, check the record status. If the record is in **Completed** state, click **Return**.

4. Repeat 2 and 3 to configure data for other dynamic data tables.

Step 3 Sign in to the CEC as a tenant administrator and configure a customized page.

1. Choose **Configuration Center** > **Expansion Management** > **Page Configuration**.
2. Click and enter the page name to create a customized page.

   A maximum of five customized pages can be created.
3. Click **Page Layout**, select **Block layout** or **Grid layout**, and set **Number of rows** to **2**, **3**, or **4**.
4. Click **SAVE**.
5. Click **Page interface** and select **Internal interface**.
6. Click **Add Object**, customize **Object Name**, set **Underlying Table Name** to the dynamic data table released in 1, select some or all fields from the selected dynamic data table in the **Selected Object Attributes** column, and click **Save** in the **Operation** column.

   A maximum of five objects can be added.
7. Click **Save** to save the page API.
8. Click the name of the customized page. In the **Page Status** area, click to release the page.

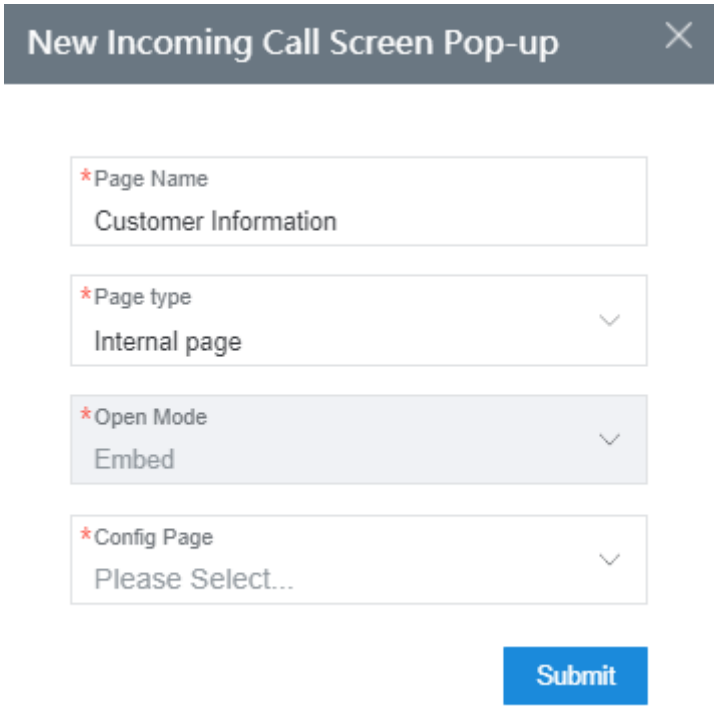
Step 4 Sign in to the CEC as a tenant administrator and configure a screen pop-up.

1. Choose **Configuration Center** > **Expansion Management** > **Screen Pop-up**.
2. Click the **Audio/Video** tab and configure a screen pop-up of the corresponding type.

   Screen pop-ups can be integrated into the corresponding agent workbenches. For example, an **Audio/Video** screen pop-up can be integrated into the **Audio and Video Workbench**.
3. Click **New** in the upper right corner. The page shown in **Figure 6-2** is displayed.
4. Enter the page name for the screen pop-up, set **Page type** to **Internal page**, set **Config Page** to the customized page released in 3, and click **Submit**. In the **Successful** dialog box that is displayed, click **OK**.

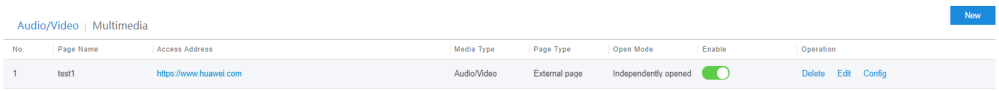
   A maximum of five screen pop-ups can be configured.

Figure 6-2 Configuring an inbound call screen pop-up



5. In the inbound call screen pop-up list, enable the new page, as shown in **Figure 6-3** (green button).

   Figure 6-3 Enabling the page

   If multiple inbound call screen pop-ups are enabled, click **Up** or **Down** in the **Operation** column to adjust the pop-up sequence of a screen pop-up. The screen pop-up on the top of the list pops up first.

6. Sign out the agent and sign in again for the configuration to take effect.

----End

6.4 Test and Verification

Prerequisites

You have enabled the voice and video agent function or multimedia agent function. For details, see **4.3.1 Enabling the Voice, Video, or Multimedia Agent Function in the Customer Service Cloud**.

Procedure

Step 1 Sign in to the CEC, choose **Configuration Center** > **Access Configuration** > **Called Route**, and view access codes.

Step 2 Click 

on the top of the page to sign in an agent.

Step 3 Simulate a customer to dial the access code of the agent using a test number.

Step 4 After the call is automatically connected, verify that the configured page can be automatically opened in the agent framework.

----End