

Custom Helpdesk Integration Guide

Yeastar P-Series Software Edition

Version: 1.0

Date: 2025-05-08



Contents

- About This Guide..... 1**
- Integration Template..... 3**
 - Add a Helpdesk Integration Template..... 3
 - Manage Helpdesk Integration Templates..... 22
- Integration Setup.....26**
 - Integrate Yeastar P-Series Software Edition with a Helpdesk using Template.....26
 - Set up Helpdesk Integration Features..... 30
- Reference..... 38**
 - XML Descriptions for Integration Template.....38

Custom Helpdesk Integration Overview

Yeastar P-Series Software Edition supports to integrate with a Helpdesk via API, so as to combine phone system with customer support service. The integration allows Helpdesk users to quickly get customers' profile and history call details on Helpdesk system while dealing with customer phone calls using their PBX extensions; it can also implement automatic ticket creation based on customer phone calls, so as to streamline the workflow. If your desired Helpdesk is not listed among the ready-made integrations provided by Yeastar, you can create a custom Helpdesk integration template to connect the Yeastar P-Series Software Edition with a Helpdesk that supports REST API.

Requirements

- **PBX Server**
 - **Firmware:** Version 83.18.0.102 or later.
 - **Plan:** Enterprise Plan (EP) or Ultimate Plan (UP)
- **Helpdesk System:** Supports REST API.

Key features

The integration of Yeastar P-Series Software Edition with a custom Helpdesk via API provides the following key features:



Note:

The availability of the features (excluding 'click to call' feature) depends on the capability of the Helpdesk and your template settings.

Click to Call

Users can launch calls by a single click on the detected numbers from the Helpdesk via Yeastar Linkus Web Client.



Note:

This feature requires users to install ['Yeastar Linkus for Google'](#) Chrome extension and set up Linkus Web Client to work with the Chrome extension.

Call Popup

Automatically bring up the contact's profile on the web browser when users (with their Linkus Web Client or Linkus Desktop Client logged in) receive / answer an inbound call from a Helpdesk contact, or supports to manually open the contact URL during a call from the call window.

Contact Synchronization

Synchronize Helpdesk contacts to an associated PBX phonebook when receiving inbound calls from / placing outbound calls to Helpdesk contacts.

Contact Creation

A new contact can automatically or manually be created in Helpdesk for unknown inbound calls or outbound calls.

Call Journal

All the call activities get logged automatically to the Helpdesk when a user ends calls with Helpdesk contacts.

Call Recording Playback

Helpdesk users can see and play recordings stored in the PBX from corresponding call log in Helpdesk interface.

Automatic Ticket Creation

A new ticket will automatically be created in Helpdesk for inbound calls or outbound calls.

Guidances

- For detailed information about the integration template, see the following topics:
 - [Add a Helpdesk Integration Template](#)
 - [Manage Helpdesk Integration Templates](#)
 - [XML Descriptions for Integration Template](#)
- For integration setup and feature configurations, see the following topics:
 - [Integrate Yeastar P-Series Software Edition with a Helpdesk using Template](#)
 - [Set up Helpdesk Integration Features](#)

Integration Template

Add a Helpdesk Integration Template

Yeastar P-Series Software Edition allows you to create and configure custom integration templates, which can be used to seamlessly integrate the PBX with a Helpdesk system via API, enabling data exchange and synchronization. This topic describes how to create a Helpdesk integration template and configure the necessary settings to meet specific operational requirements.

Requirements and restrictions

Requirements

- **PBX Server:** The PBX server should meet the following requirements.
 - **Firmware:** Version 83.18.0.102 or later.
 - **Plan:** Enterprise Plan (EP) or Ultimate Plan (UP)
- **Helpdesk System:** Supports REST API.

Restrictions

Supports to add up to **10** custom Helpdesk integration templates.

Prerequisites

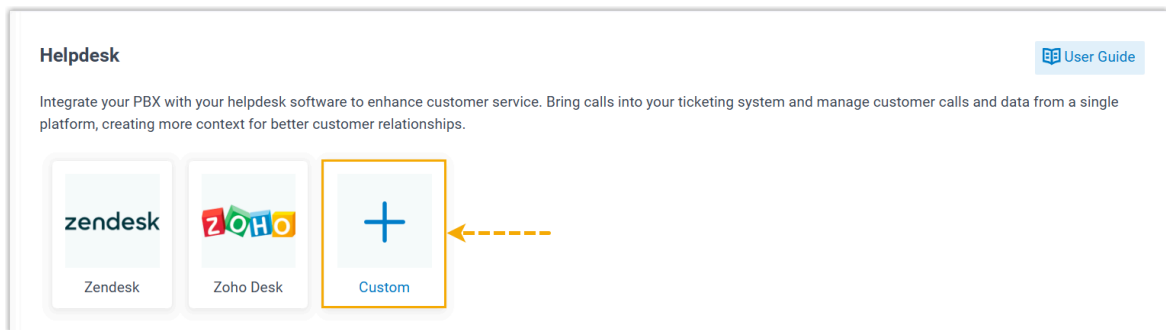
- You have basic coding skills
- You have obtained the REST API documentation of your desired Helpdesk for reference, and get the following information:
 - **Authentication method:** Determine the authentication method used by your desired Helpdesk (None, basic, or OAuth2 authentication method are supported).
 - **API request information:** You have gathered the necessary API request URLs and data specification for the following features according to the capabilities of the Helpdesk and your specific requirements.

API request	Description
Search contacts	The API request URLs to search contacts within the Helpdesk using specific criteria.

API request	Description
Get users	Optional. The API request URL to retrieve user information from the Helpdesk.
Create contacts	Optional. The API request URL to add new contacts to the Helpdesk.
Call journalling	Optional. The API request URL to create call records in the Helpdesk.
Create tickets	Optional. The API request URL to create new tickets in the Helpdesk.

Procedure

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. In the Helpdesk list, click **Custom**.



You are redirected to the Template Management page.


3. Click **Add** to add a template.
4. In the pop-window, complete the following template settings according to the capabilities the Helpdesk system and your specific requirements.
 - [Configure general settings](#)
 - [\(Optional\) Configure user association](#)
 - [Configure authentication method](#)
 - [Configure contacts searching for synchronization and call popup](#)
 - [\(Optional\) Configure new contact creation](#)
 - [\(Optional\) Configure call journaling](#)
 - [\(Optional\) Configure automatic new ticket creation](#)
5. Click **Save** to generate a custom Helpdesk integration template.

Configure general settings

In the **General** section, complete the basic settings for the template.

▼ **General**

* Logo



Upload Logo
 Drag and drop the image or click to upload. Supported file format: PNG, JPG, JPEG. PNG is recommended; Suggested Resolution: 150x150; File size: no more than 500KB.

* Helpdesk Name * Max Concurrent Requests

Remark

- **Logo:** Upload a logo for the Helpdesk, which will be displayed in the Helpdesk list on the Integration page.



Note:

The logo file should meet the following requirements:

- **File format:** PNG (recommended), JPG, or JPEG
- **Suggested Resolution:** 150 × 150
- **File size:** Less than 500KB

- **Helpdesk Name:** Enter the name of the Helpdesk.
- **Max Concurrent Request:** Specify the maximum number of concurrent HTTP requests allowed to the Helpdesk.
- **Remark:** Optional. Enter a description for the Helpdesk template, which will be displayed in the Template Management page.

(Optional) Configure user association

In the **User Association** section, enable and configure the feature to support associating Helpdesk users with PBX extensions.



Note:

If enabled, the system will require associating Helpdesk users with PBX extensions during integration, as shown in the following picture. Only the associated extension users can use the Helpdesk integration functionalities.

Custom Helpdesk User	Extension / Extension Group
user3 - user3@example.com	Please select at least one extension. ↗
user2 - user2@example.com	Please select at least one extension. ↗
user1 - user1@example.com	Please select at least one extension. ↗

1 **User Association**

* Get User

2 `https://www.api.example.com/v1/users?type=ActiveUsers`

User Field Mapping

PBX Field	CRM Field
User Unique ID	<code>data.#.id</code>
First Name	<code>data.#.First_Name</code>
Last Name	<code>data.#.Last_Name</code>
Email	<code>data.#.Email</code>

3

1. Turn on the switch of **User Association**.
2. In the **Get User** field, enter the Helpdesk API request URL for getting the list of Helpdesk users.



Note:

- The default HTTP request method is `GET`, if you need to use a different request method, you can manually change it in the template later. For more detailed descriptions about the template parameters in this scenario, see [XML Description - User Association Scenario](#).
- The values input by the users during the integration [authentication](#) process can be used as variables in the request if necessary.

For example,

```
https://www.api.example.com/v1/users?type=ActiveUsers
```


3. In the **User Field Mapping** section, map the Helpdesk fields to the required PBX fields by specifying the JSON path (e.g. `data.#(key=="value").field`) for the corresponding fields in the Helpdesk response.



Tip:

For more information about the path syntax, see [GJSON Path Syntax](#).

PBX Field	Description
User Unique ID	The unique ID of the user. For example, <code>users.#.id</code> .
First Name	The first name of the user, which will be displayed in the Helpdesk user list. For example, <code>users.#.First_Name</code> .
Last Name	The last name of the user, which will be displayed in the Helpdesk user list. For example, <code>users.#.Last_Name</code> .
Email	The email of the user, which will be displayed in the Helpdesk user list, and can be used for automatic association of Helpdesk users and PBX extensions. For example, <code>users.#.Email</code> .

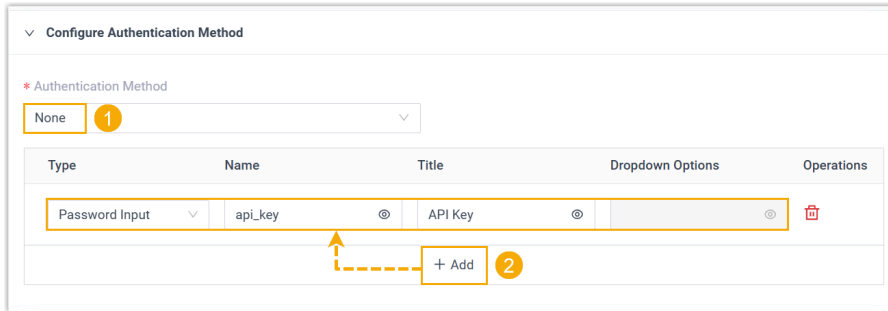
Configure authentication method

In the **Configure Authentication Method** section, configure the authentication method based on the Helpdesk system's requirements.

- [None authentication](#)
- [Basic authentication](#)
- [OAuth2 authentication](#)

None authentication

This method is used when there is no authentication required, or when the Helpdesk only requires relevant variables for the integration, such as an API Key for the request, or a webhook URL for real-time data delivery.



1. In the **Authentication Method** drop-down list, select **None**.
2. **Optional:** Click **Add** to add custom fields, which will be displayed as input field on the integration page, requiring users to provide the necessary information for the integration.




Note:

- If no fields are added, the Helpdesk will be integrated directly without requiring any additional settings or input from the users.
- Supports up to 5 custom fields.
- The values input by users during the integration authentication process can be used as variable values in the template.

Setting	Description
Type	Select the type of the custom field. <ul style="list-style-type: none"> • Text Input: A standard input field where users can enter plain text. • Password Input: A specialized input field for sensitive information, with text masked for privacy. • Dropdown List: A selectable list that allows users to choose from a pre-defined list of options.
Name	Specify the variable name for the field, which can be referenced within the template as needed.
Title	Specify the title for the field, which will be display above the input component.
Dropdown Options	If the Type is set to Dropdown List , enter the options in this field.

Note:

Setting	Description
	 Use commas to separate multiple options. For example, <code>option1,option2</code> .

Basic authentication

This method authenticates requests using credentials, such as username and password or an API key. In this case, every HTTP request sent to the Helpdesk will include a header field in the format of `Authorization: Basic {{.basic_string}}`, where `{{.basic_string}}` is the Base64-encoded combination of the username and password or API key specified during the integration.


1. In the **Authentication Method** drop-down list, select **Basic**.
2. In the **Credential Type** drop-down list, select **Username and Password** or **API Key** according to the Helpdesk system's requirements.
3. Configure the required fields for users to enter credentials, and add additional custom fields if needed.



Note:

- Supports up to 5 fields.
- The values input by users during the integration authentication process can be used as variable values in the template.

Setting	Description
Type	Select the type of the custom field. <ul style="list-style-type: none"> • Text Input: A standard input field where users can enter plain text.

Setting	Description
	<ul style="list-style-type: none"> • Password Input: A specialized input field for sensitive information, with text masked for privacy. • Dropdown List: A selectable list that allows users to choose from a pre-defined list of options.
Name	Specify the variable name for the field, which can be referenced within the template as needed.
Title	Specify the title for the field, which will be display above the input component.
Dropdown Options	<p>If the Type is set to Dropdown List, enter the options in this field.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note: Use commas to separate multiple options. For example, <code>option1,option2</code>.</p> </div>

4. In the **Base64 Encoded Credential** field, define the combination format of the credential variables, which will be used to generate the basic authentication string (*basic_string*).

For example, `{{.username:.password}}` or `{{.api_key:}}`.

OAuth2 authentication

This method uses tokens to grant access without sharing credentials. It requires obtaining the necessary header and other parameters (e.g. OAuth access token) using specific API requests to the Helpdesk.

The screenshot shows the 'Configure Authentication Method' interface with the following configuration:


- 1 Authentication Method:** A dropdown menu is set to 'OAuth2'.
- 2 Fields:** Two 'Password Input' fields are defined:

Type	Name	Title	Dropdown Options
Password Input	client_id	Client ID	
Password Input	client_secret	Client Secret	
- 3 Endpoints:**
 - Authorization Endpoint:** `https://api.example.com/oauth2/auth?client_id={{.client_id}}`
 - Token Endpoint:** `https://api.example.com/oauth2/token?client_id={{.client_id}}&client_secret={{.client_secret}}`
 - Scope (Optional):** `contacts.read,contacts.write,calls.read,calls.write,user.read,user.write`
 - Additional Query String (Optional):** (Empty field)

1. In the **Authentication Method** drop-down list, select **OAuth2**.
2. Configure the required fields for users to enter credentials.

**Note:**

- If more fields are required, you need to manually add custom fields in the template later. For more detailed descriptions about the template parameters in this scenario, see [XML Description - Authentication Scenario](#).
- The values input by users during the integration authentication process can be used as variable values in the template.

Setting	Description
Type	Select the type of the custom field. <ul style="list-style-type: none"> • Text Input: A standard input field where users can enter plain text. • Password Input: A specialized input field for sensitive information, with text masked for privacy. • Dropdown List: A selectable list that allows users to choose from a pre-defined list of options.
Name	Specify the variable name for the field, which can be referenced within the template as needed.
Title	Specify the title for the field, which will be display above the input component.
Dropdown Options	If the Type is set to Dropdown List , enter the options in this field. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>Note: Use commas to separate multiple options. For example, <code>option1,option2</code>.</p> </div>

3. In the **Authorization Endpoint** field, enter the Helpdesk authorization URL, to which users will be redirected for authorization during integration.

For example,

```
https://www.api.example.com/auth/v2/authorize?client_id={{.client_id}}
```

4. In the **Token Endpoint** field, enter the Helpdesk API request URL for getting access token and refresh token.

For example,

```
https://www.api.example.com/oauth/v2/token?client_id={{.client_id}}&client_secret={{.client_secret}}
```

5. **Optional:** In the **Scope (Optional)** field, enter the scope to specify the data in the Helpdesk that the PBX is allowed to access.



Note:

Use comma to separate multiple fields.

For example,

```
contacts.read,contacts.write,calls.read,calls.write,user.read,user.write
```

6. **Optional:** If the Helpdesk system authenticates requests using query parameters instead of scope, you can add the query parameters in the **Additional Query String (Optional)** field.

For example, `client_id={{.client_id}}&redirect_uri={{.redirect_uri}}`.

After the template setting is saved, the authentication page of the Helpdesk will automatically display the PBX authentication information, which can be used to create an application in the Helpdesk system and retrieve the required authorization information from the Helpdesk for the integration.

Network

Please go to your Custom Helpdesk API console and do the following settings:

- **Home Page URL:** enter your PBX domain name
- **Authorized Redirect URI:** enter your PBX address followed by the path `"/integration/callback/api/helpdesk/74a843885f564c2d8ea47bd47545adf2/"`. You can click the icon below to copy the address directly.

<p><small>* Homepage URL</small></p> <input style="width: 95%;" type="text" value="https://example.yeastar.com"/>	<p><small>* Authorized Redirect URI</small></p> <input style="width: 95%;" type="text" value="https://example.yeastar.com/integration/callback/api/helpdesk/74a843885f564c2d8ea47bd47545adf2/"/>
---	--

Authorization

<p><small>* Client ID</small></p> <input style="width: 95%;" type="text"/>	<p><small>* Client Secret</small></p> <input style="width: 95%;" type="text"/>
--	--

Configure contacts searching for synchronization and call popup

In the **Synchronize Contacts Automatically** section, specify the contacts type available for users to select when configuring integration features, and specify the API requests to search for the specified types of contacts. These configurations can be used to implement Helpdesk contacts searching, automatic contact synchronization and call popup.

1. Add and configure contact types and specify the corresponding API requests.

The screenshot shows a configuration panel titled "Synchronize Contacts Automatically". At the top left, there is a blue button with a plus sign and the word "Add". Below this, there is a list of contact types. The first item is "Leads" with a right-pointing chevron and a trash icon. The second item is "Contacts" with a down-pointing chevron and a trash icon. The "Contacts" item is expanded, showing two fields: "Contact Type:" with a text input containing "Contacts", and "Contact Match Query URL:" with a text input containing the following URL: `https://www.api.example.com/v1/Contacts/search?criteria=((Phone:equals:{{.Phone}})or(Home_Phone:equals:{{.Phone}})or(Mobile:equals:{{.Phone}})or(Asst_Phone:equals:{{.Phone}}))`. A dashed orange arrow points from the "Add" button to the "Contacts" item in the list.

- **Contact Type:** Enter the name of the contact type. For example `Contacts`.

The name set here will appear as an option for automatic contact synchronization when users configure the integration features.

- **Contact Match Query URL:** Enter the Helpdesk API request URL used to search for contacts with specific criteria.

In this scenario, the following variables can be used as the values of criteria parameters:

- The variables obtained from user input during the integration [authentication process](#).
- `{{.Phone}}`: The phone number associated with inbound or outbound calls, or the number entered by users for contact searching.

For example,

```
https://www.api.example.com/v1/Contacts/search?criteria=((Phone:equals:{{.Phone}})or(Home_Phone:equals:{{.Phone}})or(Mobile:equals:{{.Phone}})or(Asst_Phone:equals:{{.Phone}}))
```

2. Configure the URL for call popup using one of the following methods.

Specify URL Format

Configure the call popup URL by specifying the URL format, where you can use variables to dynamically construct the URL.

The screenshot shows a configuration window for the 'Contact Popup URL'. It contains two main sections:

- * Contact Popup URL:** A dropdown menu with the selected option 'Specify URL Format'.
- * URL Format:** A text input field containing the template: `{{.domain_url}}/{{.CustomValue}}/tab/{{Capitalize .ContactSyncType}}/{{.ContactId}}`.

- In the **Contact Popup URL** drop-down list, select **Specify URL Format**.
- In the **URL Format** field, enter the expression of the Contact URL.

In this scenario, the following variables can be used, the values of which are obtained from the Helpdesk contact search results.

- The variables obtained from user input during the integration authentication process.
- `{{ .ContactSyncType }}`: The [contact type](#).
- `{{ .ContactId }}`: The [contact ID](#).
- `{{ .CustomValue }}`: A [custom field](#) mapped in the **Contact Field Mapping** section.



Note:

If the provided variables do not meet your requirements, you can manually edit the template later to define additional custom variables and retrieve the corresponding values. For more detailed descriptions about the template parameters in this scenario, see [XML Description - Retrieve Additional Variables](#).

For example,

```
{{.doman_url}}{{.CustomValue}}/  
tab/{{Capitalize .ContactSyncType}}/{{.ContactId}}
```


Retrieve from Contact Fields

Extract the Contact URL directly from a specific field in the contact search response, where the contact URL can be retrieved.

The screenshot shows two configuration fields. The first is labeled '* Contact Popup URL' and has a dropdown menu with 'Retrieve from Contact Fields' selected. The second is labeled '* Contact Field for URL' and has a text input field containing 'data.#.contactUrl'.

- a. In the **Contact Popup URL** drop-down list, select **Retrieve from Contact Fields**.
- b. In the **Contact Field for URL** field, specify the JSON path for the corresponding field in the response. For example, `data.#.contactUrl`.

3. In the **Contact Field Mapping** section, map the Helpdesk fields to the required PBX fields by specifying the JSON path (e.g. `data.#(key=="value").field`) for the corresponding fields in the Helpdesk response.


 **Note:**
 If more than one variable need to be retrieved for a field, use commas to separate the variables. For example, `data.#(phone=="{{.Phone}}")#.street,data.#(mobile=="{{.Phone}}")#.street`.

a. Map the following required fields.

Contact Field Mapping	
<input type="checkbox"/> PBX Field	Helpdesk Field
<input checked="" type="checkbox"/> Contact ID	<code>data.#.id</code>
<input checked="" type="checkbox"/> First Name	<code>data.#.First_Name</code>

- **Contact ID:** The ID of the searched contact. For example, `data.#.id`.
- **First Name:** The first name of the searched contact. For example, `data.#.first_name`.

b. Enable and map the respective PBX fields according to your needs.

 **Note:**
 You need to enable and map at least one phone number.

Contact Field Mapping	
<input type="checkbox"/> PBX Field	Helpdesk Field
<input checked="" type="checkbox"/> Contact ID	<input type="text" value="data.#.id"/>
<input checked="" type="checkbox"/> First Name	<input type="text" value="data.#.First_Name"/>
<input checked="" type="checkbox"/> Last Name	<input type="text" value="data.#.Last_Name"/>
<input checked="" type="checkbox"/> Company	<input type="text" value="data.#.Account_Name"/>
<input checked="" type="checkbox"/> Email	<input type="text" value="data.#.Email"/>
<input checked="" type="checkbox"/> Business Number	<input type="text" value="data.#.Phone"/>

- c. If the existing PBX fields do not provide the variables you need, enable and map the desired response field to the **Custom Value** field, which can be referenced in the template with the variable name `{{.CustomValue}}`.



Note:

This variable is specially used for [constructing the Contact URL](#), and will not appear in the PBX contact details.

<input checked="" type="checkbox"/> Custom Value	<input type="text" value="data.#.Owner"/>
--	---

(Optional) Configure new contact creation

If you want to enable the creation of a new contact in the Helpdesk when the caller’s number does not match any existing Helpdesk contact, enable and configure the **Create New Contact** settings.

1. Turn on the switch of **Create New Contact**.
2. Add and configure the contact types available for users to select when configuring integration features.

- **Contact Type:** Specify the type of the contact. For example, `Contacts`.
- **Contact Creation URL (POST):** Enter the Helpdesk API request URL used to create a new contact.
For example,

```
https://www.api.example.com/v1/Contacts
```

- **Contact Creation Request Body:** Enter the request body (in JSON format) to pass the contact information.

In this scenario, the following variables can be used:

- The variables obtained from user input during the integration authentication process.
- `{{.LastName}}`: Required. The last name of the contact.
- `{{.FirstName}}`: Required. The first name of the contact.
- `{{.BusinessNumber}}`: Required. The phone number of the contact.



Note:



If the provided variables do not meet your requirements, you can manually edit the template later to define additional custom variables and retrieve the corresponding values. For more detailed descriptions about the template parameters in this scenario, see [XML Description - Retrieve Additional Variables](#).

For example,

```
{
  "data": [
    {
      "Last_Name": "{{.LastName}}",
      "First_Name": "{{.FirstName}}",
      "Phone": "{{.BusinessNumber}}"
    }
  ]
}
```

(Optional) Configure call journaling

If you want to enable call journaling to automatically log call activities and details in the Helpdesk, enable and configure the **Call Journal** feature.

1. Turn on the switch of **Call Journal**.

2. Configure the API request and body for call journaling.

- Create Call Journal URL (POST):** Enter the Helpdesk API request URL used to create a call log in the Helpdesk.
For example,
`https://www.api.example.com/v2/Calls`
- Call Journal Creation Request Data:** Enter the request body (in JSON format) to pass the call details.

3. Configure the call details:

- Subject:** Extension Call
- Description:** Call: {{Time}} {{Call_Log_Status}} from {{Call_From}} to {{Call_To}} {{Talk_Duration}}
- Play Call Recording:** Enabled

1. Turn on the switch of **Call Journal**.

2. Configure the API request and body for call journaling.

- **Create Call Journal URL (POST):** Enter the Helpdesk API request URL used to create a call log in the Helpdesk.

For example,

```
https://www.api.example.com/v2/Calls
```

- **Call Journal Creation Request Data:** Enter the request body (in JSON format) to pass the call details.



Note:

For the variables supported in the request body, see [XML Description - Call Journal Variables](#).

For example,

```

{
  "data": [
    {
      "{{.Owner}}>{{.WhoModule}}": {
        "Description": "{{.Description}}",
        "Voice_Recording__s": "{{.RecordPath}}",
        "Call_Start_Time": "{{TimeFormat .StartTimeUnix
'yyyy-MM-ddTHH:mm:ss-z' 'l'}}",
        "Subject": "{{.Subject}}",
        "Call_Type": "Inbound",
        "Call_Result": "{{.Call_Log_Status}}",
        "Call_Duration": "{{.Talk_Duration_Sec}}"
      }
    }
  ]
}

```

3. If you want to allow users to customize specific settings for call journaling, including key attributes such as logging details and call recording playback option, enable the following options and configure the default values as needed.



Note:

- The values set in these fields will also be captured as variable values if necessary.
- For more information about the supported variables, see [XML Description - Call Journal Variables](#).

- **Subject:** If enabled, enter a default subject for call journal. All call logs will be created with the specified subject.
- **Description:** If enabled, specify the details of the call logs.
For example,

```

Call: {{.Time}} {{.Call_Log_Status}} from {{.Call_From}} to
{{.Call_To}} {{.Talk_Duration}}

```

- **Play Call Recording:** Decide whether to allow users to configure the option for playing call recordings in the Helpdesk.

(Optional) Configure automatic new ticket creation

If you want to enable the automatic creation of new tickets based on calls, enable and configure the **Create New Ticket Automatically** settings.

**Note:**

When the feature is set to create a ticket before the call (i.e. when the phone rings) in later use, it might be missing some information. In this case, you need to manually edit the template to update the information for the created ticket. For more information, see [XML Description - Extended Configurations](#),

1 Create New Ticket Automatically

* Create New Ticket URL (POST)

https://www.api.example.com/api/v1/tickets

2 * New Ticket Creation Request Body

```
{
  "subject": "{{Subject}}",
  "contactId": "{{ContactId}}",
  "phone": "{{ContactNumber}}",
  "description": "{{Description}}"
}
```

3 * Subject

{{Communication_Type}} {{Call_Status}} - from {{Call_From}} to {{Call_To}}

* Description

{{Time}} {{Communication_Type}} {{Call_Status}} - from {{Call_From}} to {{Call_To}} {{Talk_Duration}}

1. Turn on the switch of **Create New Ticket Automatically**.

2. Configure the API request and body for new ticket creation.

- **Create New Ticket URL (POST):** Enter the Helpdesk API request URL used to create a new ticket in the Helpdesk.

For example,

```
https://www.api.example.com/api/v1/tickets
```

- **New Ticket Creation Request Body:** Enter the request body (in JSON format) to pass the ticket details.

**Note:**

For the variables supported in the request body, see [XML Description - New Ticket Creation Variables](#).

For example,

```
{
```

```
"subject": "{{.Subject}}",
"contactId": "{{.ContactId}}",
"phone": "{{.ContactNumber}}",
"description": "{{.Description}}"
```

- If you want to allow users to customize subjects and descriptions for the new ticket, enable the following options and configure the default values as needed.



Note:

- The values set in these fields will also be captured as variable values if necessary.
- For more information about the supported variables, see [XML Description - New Ticket Creation Variables](#).

- Subject:** If enabled, enter a default subject for the ticket.

For example,

```
{{.Communication_Type}} {{.Call_Status}} - from {{.Call_From}} to
{{.Call_To}}
```

- Description:** If enabled, specify the details of the ticket.

For example,

```
{{.Time}} {{.Communication_Type}} {{.Call_Status}} - from
{{.Call_From}} to {{.Call_To}} {{.Talk_Duration}}
```

Related information

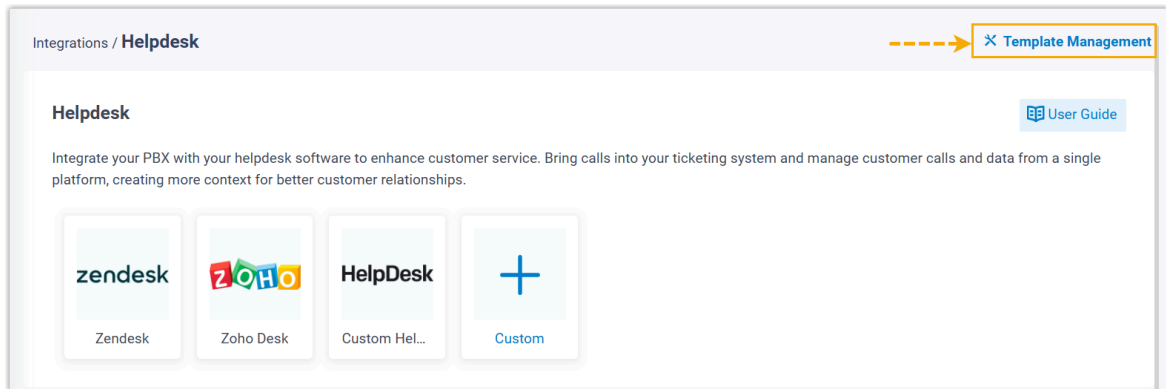
[Integrate Yeastar P-Series Software Edition with a Helpdesk using Template](#)

Manage Helpdesk Integration Templates


This topic describes how to manage the Helpdesk templates.

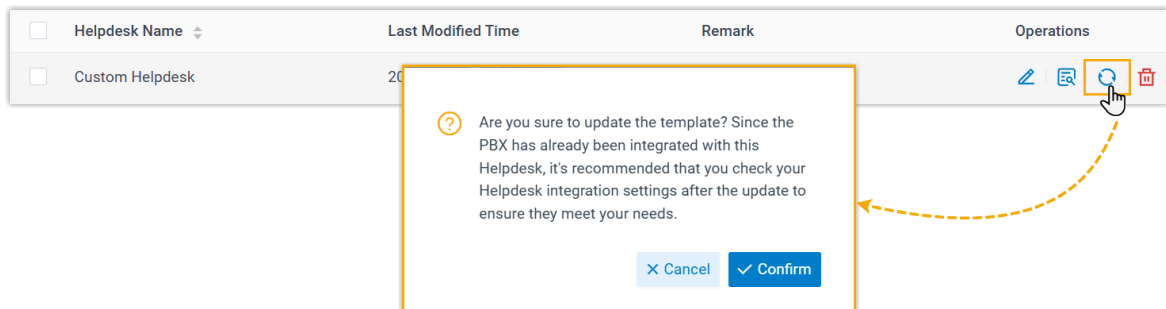
Edit an integration template


- Log in to PBX web portal, go to **Integrations > Helpdesk**.
- At the top-right corner of the page, click **Template Management**.



You are redirected to the Template Management page.

3. Go to the **Custom Templates** tab, then click  beside the desired template.
4. In the pop-up window, edit the template settings according to your needs.
5. Click **Save**.
6. If you have integrated the custom Helpdesk with PBX, do as follows to directly apply the changes to the current integration.

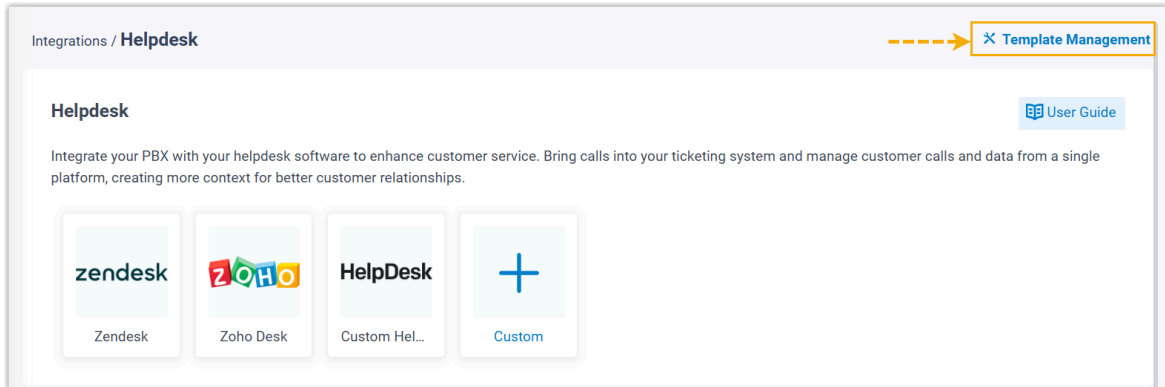


- a. Click  beside the template.
- b. In the pop-up window, click **Confirm**.

Download integration templates

You can download the integration template to upload to other PBX systems for integration or manually modify the template parameters to customize settings for specific needs.

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. At the top-right corner of the page, click **Template Management**.



You are redirected to the Template Management page.

3. Select the desired template(s), then click **Download**.

The template files are downloaded to your local computer.

Upload an integration template

You can upload an integration template to the PBX systems for quick integration.

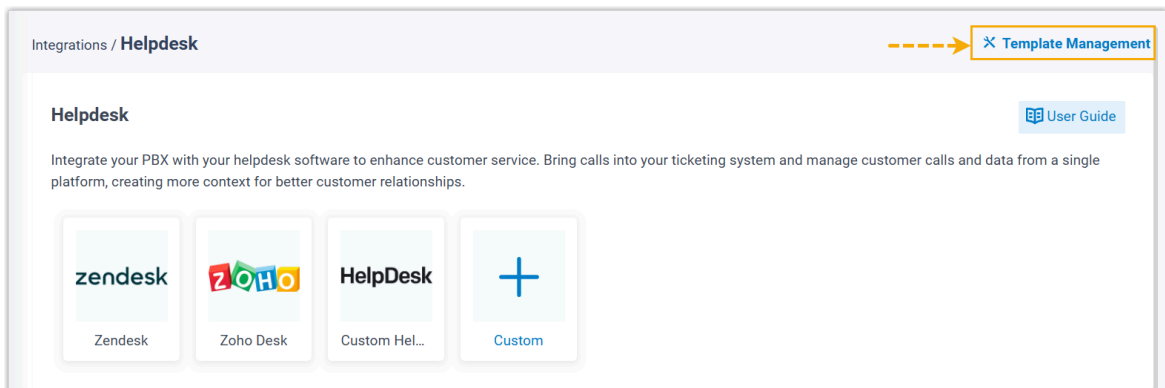


Note:

- Supports up to **10** custom Helpdesk integration templates.
- The file size should be less than 1 MB.

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.

2. At the top-right corner of the page, click **Template Management**.



You are redirected to the Template Management page.

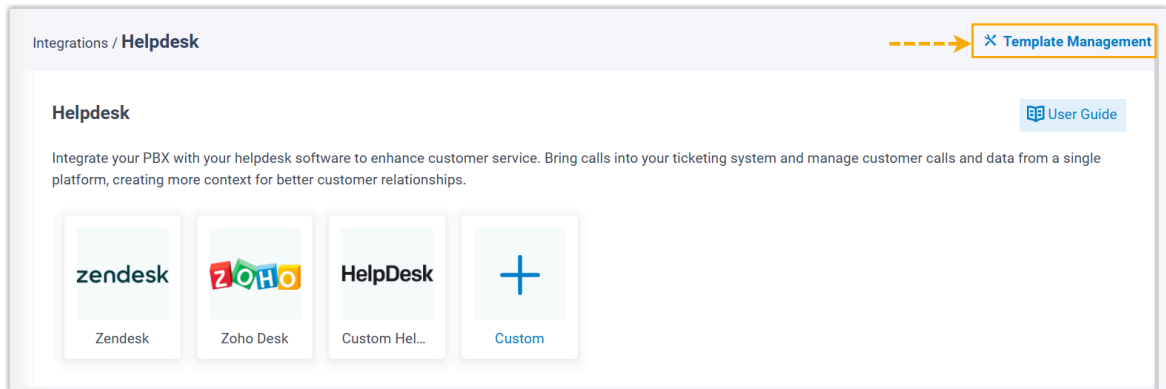
3. At the top of the template list, click **Upload**.

4. In the pop-up window, click **Browse**, then select the desired template file to upload.
5. Click **Save**.


The template file is uploaded to the PBX system, and displayed in the template list.

Delete integration templates

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. At the top-right corner of the page, click **Template Management**.



You are redirected to the Template Management page.

3. To delete a single template, do as follows:
 - a. Click  beside the template.
 - b. In the pop-up window, click **OK**.
4. To delete multiple templates, do as follows:
 - a. Select the desired templates.
 - b. At the top of the template list, click **Delete**.
 - c. In the pop-up window, click **OK**.

Integration Setup

Integrate Yeastar P-Series Software Edition with a Helpdesk using Template

If your desired Helpdesk is not in the list of ready-made integrations, you can implement a custom integration via a Helpdesk integration template. This topic describes how to establish the integration.

Requirements

- **Firmware:** Version 83.18.0.102 or later.
- **Plan:** Enterprise Plan (EP) or Ultimate Plan (UP)

Prerequisites

- You have [added a Helpdesk integration template](#).
- You have obtained the necessary authorization information from the Helpdesk system according to its authentication requirements.
 - **None authentication:** Obtain the required integration information (such as API key or Webhook URL) if needed.
 - **Basic authentication:** Obtain the credential for authentication (such as username and password, or API key).
 - **OAuth2 authentication:** Obtained the PBX authentication information first (Path: **Integrations > Helpdesk > Custom Helpdesk**), then use the information to create an application on the Helpdesk system, and obtain the corresponding authorization information (such as client ID and client secret).

Network

Please go to your Custom Helpdesk API console and do the following settings:

- **Home Page URL:** enter your PBX domain name
- **Authorized Redirect URI:** enter your PBX address followed by the path `/integration/callback/api/helpdesk/74a843885f564c2d8ea47bd47545adf2/`. You can click the icon below to copy the address directly.

* Homepage URL

* Authorized Redirect URI

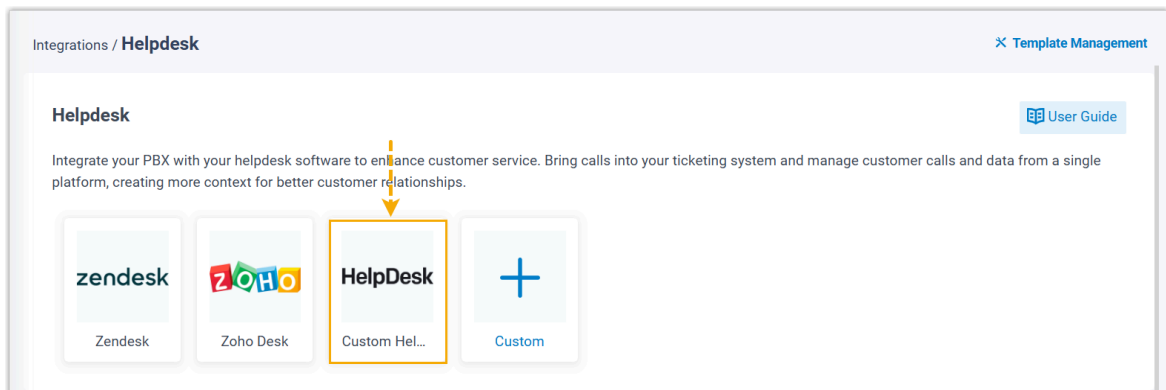
Authorization

* Client ID

* Client Secret

Procedure

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. In the Helpdesk list, click the Helpdesk that you have added via a template.



3. Complete the authentication according to the Helpdesk system's requirements.

None authentication

Settings

* Api Key

- a. In the **Settings** section, enter the required information for integration.
- b. Click **Save**.



Note:



If no additional information is required, the Helpdesk integration will be completed once you click the Helpdesk.

Basic authentication

Settings

* Username

* Password

- a. In the **Settings** section, enter the required credentials.
- b. Click **Save**.

OAuth2 authentication

Network

Please go to your Custom Helpdesk API console and do the following settings:

- **Home Page URL:** enter your PBX domain name
- **Authorized Redirect URI:** enter your PBX address followed by the path "/integration/callback/api/helpdesk/74a843885f564c2d8ea47bd47545adf2/". You can click the icon below to copy the address directly.

* Homepage URL

* Authorized Redirect URI

Authorization

* Client ID

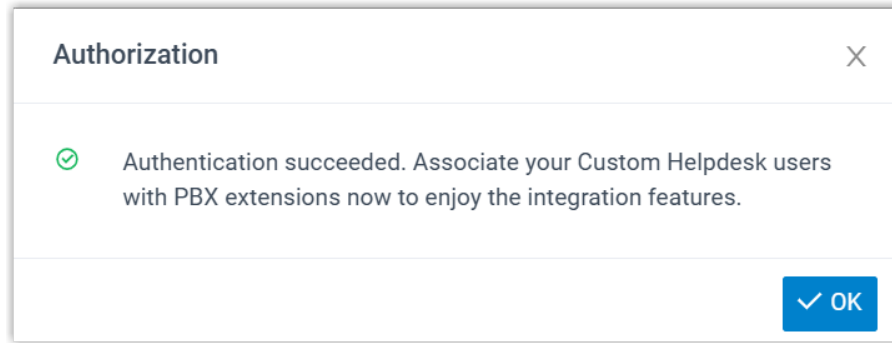
* Client Secret

- a. In the **Authorization** section, enter the authorization information obtained from the application created in the Helpdesk system.
- b. Click **Save**.

A new browser page will be launched to request for Helpdesk data access permission.

- c. Accept the request to allow the PBX to access data in your Helpdesk account.

On the PBX configuration page, a pop-up window displays the authentication result.




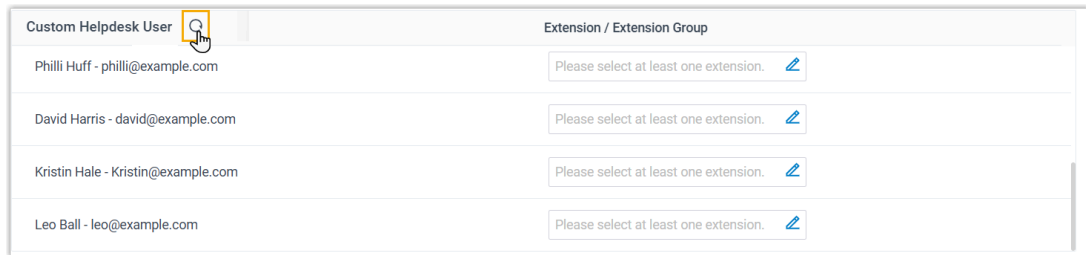
d. Click **OK** to confirm.

The **Status** field displays **Connected**, indicating that the Helpdesk integration is successfully set up.

4. Associate Helpdesk users with PBX extensions.

If the Helpdesk requires the association of Helpdesk users with PBX extensions to access the integration functionalities, complete the following settings.

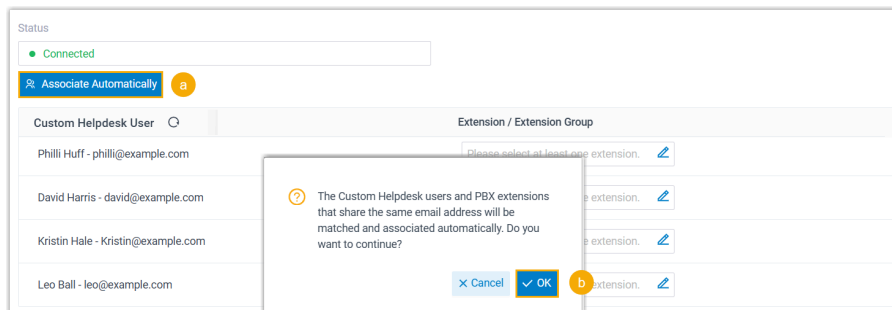
a. On the Helpdesk integration page, click  to synchronize the latest list of Helpdesk users.



b. Associate the Helpdesk users with PBX extensions.

Associate automatically

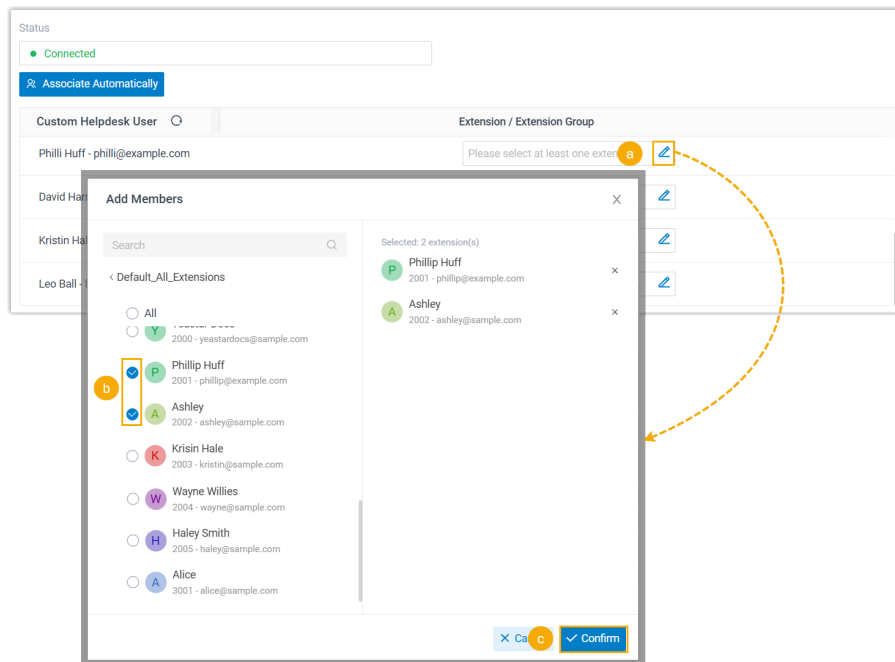
If users bind the same email address to their Helpdesk accounts and PBX extensions, you can implement automatic association of their Helpdesk accounts and PBX extensions as follows:




- i. Click the **Associate Automatically** button.
- ii. On the pop-up window, click **OK**.

Associate manually

If the user binds different email addresses to their Helpdesk account and PBX extension, you need to manually associate the user's Helpdesk user account and PBX extension.



- i. In the **Extension** field beside the Helpdesk user, click .
 - ii. In the pop-up window, select the desired user's extension(s).
 - iii. Click **Confirm**.
- c. Click **Save**.

Related information

[Set up Helpdesk Integration Features](#)

Set up Helpdesk Integration Features

This topic describes how to set up the integration features after you integrate the PBX with a custom Helpdesk, including contact synchronization, contact creation, call journal, etc.

Prerequisites

You have [integrated the PBX with a custom Helpdesk via template](#).

Set up contact synchronization

If this feature is enabled, inbound calls and outbound calls will trigger a contact lookup in the Helpdesk system. If there exists the same number in Helpdesk, the matched contact will be synchronized to the associated PBX phonebook, and the synchronized contact is read-only on PBX.

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. In the **Synchronize Contacts Automatically** section, turn on the switch.
3. In the **Synchronize Contacts From** drop-down list, select the type(s) of contacts to be synchronized from Helpdesk.

* Synchronize Contacts From

Leads × Contacts × ^

Contacts	✓
Leads	✓
Accounts	

4. In the **Synchronize to Phonebook** drop-down list, select an existing empty phonebook or create a new phonebook to store the contacts that will be synchronized from Helpdesk.



Note:

The contacts can only be synchronized to an empty phonebook.

* Synchronize to Phonebook * Name

[Create New] v Helpdesk_Synchronization

5. On the **Always Query Helpdesk** option, decide whether to search contacts in the Helpdesk real-time.

Always Query Helpdesk

When enabled, PBX will query Helpdesk in real time for contact information and support real-time dial search for Helpdesk contacts in the Linkus client. When disabled, Helpdesk will only be queried if no matching information is found in the PBX company contacts, and the real-time dial search for Helpdesk contacts in the Linkus client will also be disabled.
 Note: Enabling "Always Query" will increase API credit consumption. Please monitor your quota. For details about Helpdesk platform API quotas, refer to the [official documentation](#).

- If enabled, PBX will query Helpdesk in real time for contact information, and support real-time dial search for Helpdesk contacts in the Linkus clients of the associated extension users.
- If disabled, PBX will only query Helpdesk when there is no matching information found in the PBX company contacts.



Note:

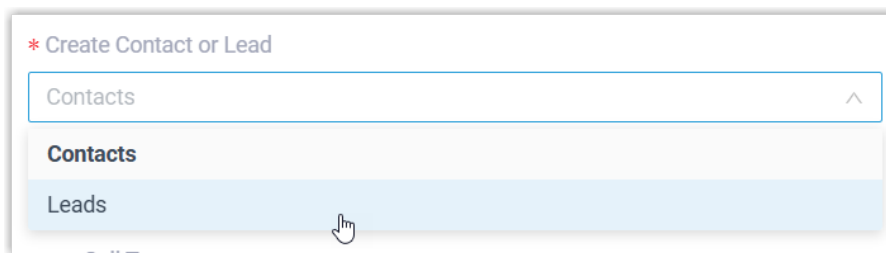
- Enabling this feature will increase API usage, so make sure to monitor your quota accordingly.
- The real-time search is not supported when [masked number](#) is enabled.

6. Click **Save**.

Set up contact creation

If this feature is enabled, a new contact can be created in the Helpdesk, automatically or manually depending on your settings, when the associated extensions call or receive calls from an unknown number.

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. In the **Create New Contact** section, turn on the switch.
3. In the **Create Contact or Lead** drop-down list, select the desired type of contact.



4. In the **Create Method** section, select the desired creation method according to your needs.
 - If you want the system to automatically create contacts in the Helpdesk based on specified types of calls, do as follows:
 - a. Select **Automatically**.

- b. In the **Call Type** drop-down list, select when will a contact be automatically created in the Helpdesk.

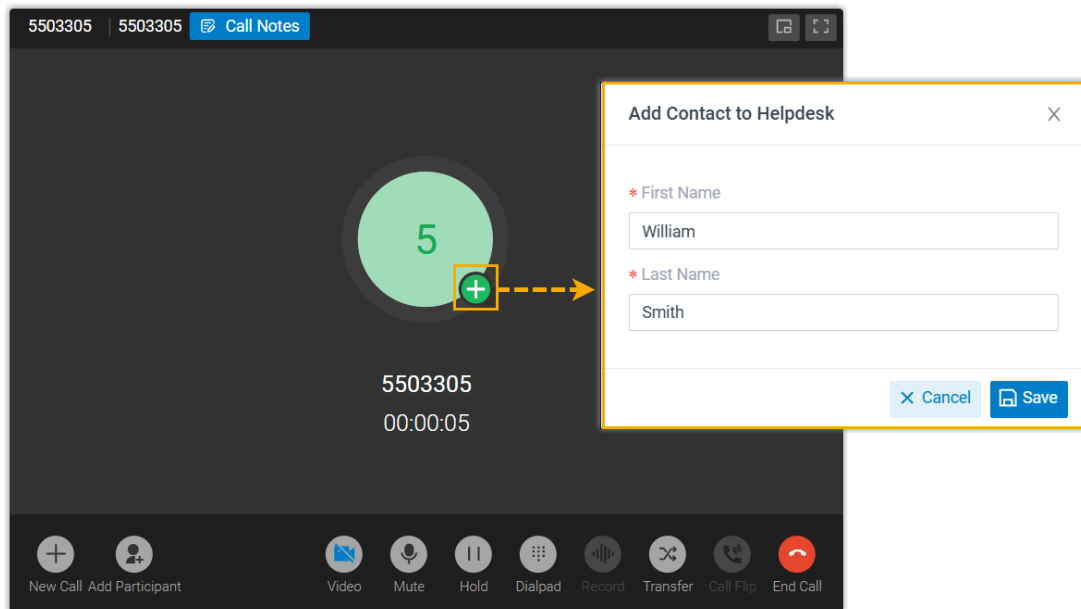
* Call Type

Inbound × Outbound × |

Inbound	✓
Outbound	✓

- **Inbound:** Inbound call from an unknown number that doesn't match a Contact already in the Helpdesk.
- **Outbound:** Outbound call to an unknown number that doesn't match a Contact already in the Helpdesk.
- If you want to allow associated extension users to manually create contacts in the Helpdesk during a call, select **Manually**.


During a call with an unknown number, users can add the contact to the Helpdesk directly on the call window.

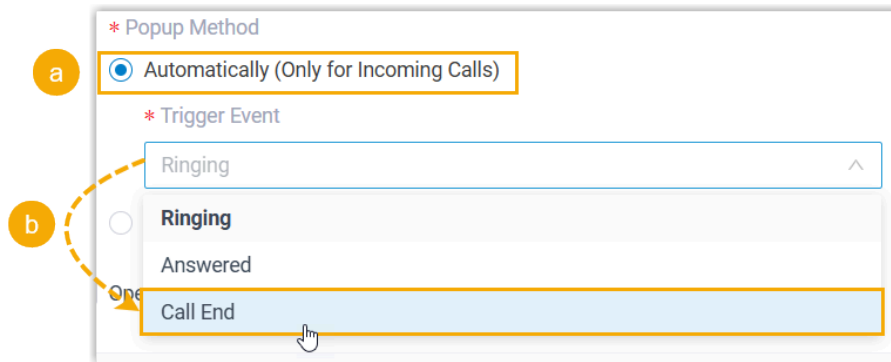


5. Click **Save**.

Set up call popup

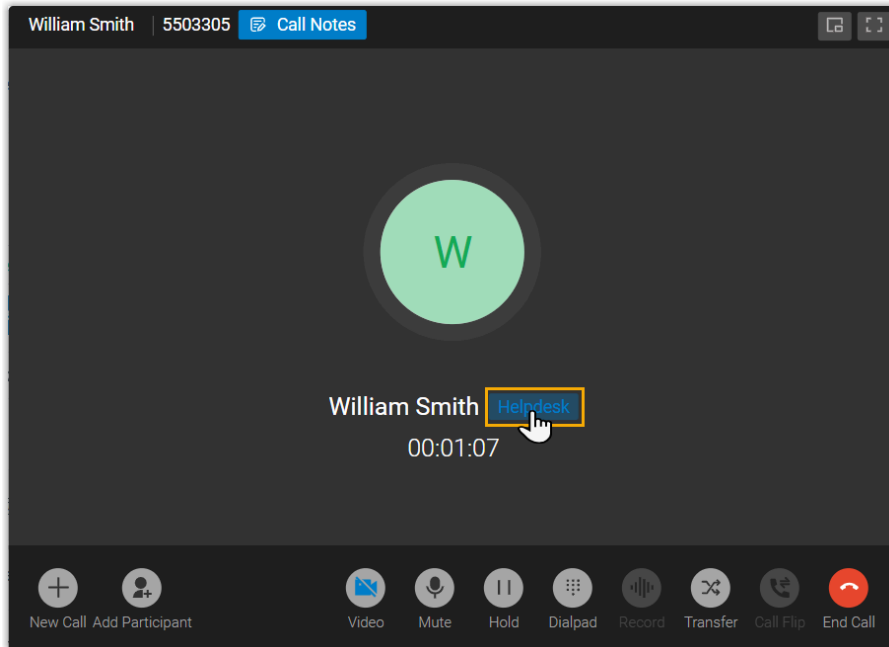
You can set up call popup for an extension user's Linkus Web Client or Desktop Client, which allows the user to receive automatic popup URL of the contact's profile based on specified call-related events, or manually open the contact's URL from the call window directly during the call with a Helpdesk contact.

1. Log in to PBX web portal, go to **Extension and Trunk > Extension**.
2. Click  to edit the extension associated with a Helpdesk user.
3. In the **Linkus Clients** tab, scroll down to your desired Linkus client, then select the checkbox of **Open Contact URL Using System-Integrated Helpdesk**.
4. In the **Popup Method** section, decide the method of call popup.
 - If you want the system to automatically open the Helpdesk contact details page, do as follows:



- a. Select **Automatically (Only for Incoming Calls)**.
- b. In the **Trigger Event** drop-down list, set when the call popup will be automatically triggered.
 - **Ring End:** A call popup will be triggered when a user receives an inbound call from a Helpdesk contact.
 - **Answered:** A call popup will be triggered when a user answers an inbound call from a Helpdesk contact.
 - **Call End:** A call popup will be triggered when a user finishes a call with a Helpdesk contact.
- If you want the associated extension user to manually open the contact's URL during a call with a CRM contact, select **Manually**.

During a call with a Helpdesk contact, users can manually open the contact URL from the call window.



5. Click **Save**.

Set up call journaling

If this feature is enabled, all outbound calls, incoming calls, and missed call records will be logged to Helpdesk automatically, which helps users to keep track of every conversation.

1. Log in to PBX web portal, go to **Integrations > Helpdesk**.
2. In the **Call Journal** section, turn on the switch.
3. Configure the following call log information according to your needs.

*** Subject**

Description

ⓘ The supported variables include {{.Time}}, {{.Call_Log_Status}}, {{.Call_From}}, {{.Call_To}}, and {{.Talk_Duration}}, among others. For a complete list, please refer to the user manual.

- **Subject:** The subject of the call log.
- **Description:** The description of the call log.




Note:

The contents can be composed of variables. For the supported variables, see [XML Description - supported variables](#).

- If you want to play call recordings stored in the PBX within the Helpdesk, select the checkbox of **Play Call Recording**.

Play Call Recording

If enabled, you can play call recordings directly within the Helpdesk system. When you click to play a recording in the Helpdesk, the system will request the recording file from the PBX. For details on how the Helpdesk platform protects the privacy of recording data, please contact your Helpdesk provider.

 **Note:**
When you click to play recording in the Helpdesk, the system will request recording file from the PBX. For details on how the Helpdesk platform protects the privacy of recording data, please contact your Helpdesk provider.

- Click **Save**.

Set up automatic ticket creation

You can set up automatic ticket creation for Helpdesk contacts, so as to automatically convert inbound calls and outbound calls into support tickets.

- Log in to PBX web portal, go to **Integrations > Helpdesk**.
- On the Helpdesk integration page, turn on **Create New Ticket Automatically**.
- Set up the time point and call type(s) of automatic ticket creation.

When to Create New Ticket	Description
Before the Call	<p>A ticket will be created in Helpdesk when the Helpdesk user receives an inbound call from, or places an outbound call to a Helpdesk contact.</p> <p>In the The Type of Call to Create Ticket drop-down list, select specific call type(s) based on which a new ticket will be created.</p> <ul style="list-style-type: none"> • Inbound: Inbound calls. • Outbound: Outbound calls.
After the Call	<p>A ticket will be created in Helpdesk when the user ends the call with a Helpdesk contact.</p> <p>In the The Type of Call to Create Ticket drop-down list, select specific call type(s) based on which a new ticket will be created.</p> <ul style="list-style-type: none"> • Inbound Answered: Inbound calls that are answered. • Inbound No Answer: Inbound calls that are missed. • Outbound Answered: Outbound calls that are answered.

When to Create New Ticket	Description
	<ul style="list-style-type: none"> • Outbound No Answer: Outbound calls that are not answered.

4. Configure the following ticket information according to your needs.

* Subject

{{Communication_Type}} {{Call_Status}} - from {{Call_From}}

Description

{{Communication_Type}} {{Call_Status}} - from {{Call_From}} to {{Call_To}}

ⓘ The supported variables include {{Time}}, {{Call_Log_Status}}, {{Call_From}}, {{Call_To}}, and {{Talk_Duration}}, among others. For a complete list, please refer to the user manual.

- **Subject:** The subject of the ticket.
- **Description:** The description of the ticket.



Note:

The contents can be composed of variables. For the supported variables, see [XML Description - supported variables](#).

5. Click **Save**.

Related information

[Custom Helpdesk Integration - Key features](#)

Reference

XML Descriptions for Integration Template

Yeastar P-Series Software Edition allows you to create CRM/Helpdesk integration templates, where you can define the necessary settings, including specific API endpoints, data fields, and functional scenarios required for the integration. This topic provides detailed description of the setting elements in the XML templates.


Template conventions

The template follows the **XML 1.1 standard** specification. This section describes the syntax and conventions of the template.

Variable

The template supports the use of variables for dynamic data. When the template is executed, the variables referenced within the template will be replaced with their respective values.

The following table shows the source of variable values and how to reference the variables in the template.

Item	Description
Source	<p>Variables in this template can be obtained from the following sources:</p> <ul style="list-style-type: none">• Pre-defined variables in the PBX system• Parameter values input by users during the integration• Values retrieve from the request response <p>You can retrieve values from the response body by specifying a JSON path, as shown in the examples below.</p> <div data-bbox="479 1501 1299 1648"><p> Tip: For more information about the path syntax, see GJSON Path Syntax.</p></div> <ul style="list-style-type: none">◦ To retrieve a response value, use the following syntax: <pre data-bbox="565 1726 1299 1806"><Output Name="UserUniqueId" Path="data.#.id"></Output></pre>◦ For retrieve a value that meets specific conditions, use following syntax:

Item	Description
	<pre><Output Name="BusinessNumber2" Path="identities.#{type=="phone_number"}# 1.value">< /Output></pre>
Reference format	<p>Variables in this template follow the text/template syntax specification of Go Language, as shown below:</p> <ul style="list-style-type: none"> • Standard variable: <code>{{ .varname }}</code> • Conditional variable: <code>{{ if .varname }} Welcome, member! {{ else }} Please sign up. {{ end }}</code> • Function calls with variable: <code>{{ Capitalize .varname }}</code>

Function

The template supports the following function, which can be used in conjunction with variables to dynamically manipulating or formatting the variable value.

Function	Description
TimeFormat	<p>The function can be used to customize the format of a timestamp.</p> <p>Format: <code>{{ TimeFormat <i>Timestamp_variable</i> "format_string" "Whether_to_convert_to_UTC_time" }}</code></p> <p>Example: <code>{{ TimeFormat .StartTimeUnix "yyyy-MM-ddTHH:mm:ss.000Z" "1" }}</code></p> <p>Assuming the value of <code>{{ .StartTimeUnix }}</code> is 1672531199, the final output would be 2023-01-01T00:59:59.000Z.</p>
ToMillis	<p>The function can be used to convert a seconds timestamp to a milliseconds timestamp.</p> <p>Format: <code>{{ ToMillis <i>Timestamp_variable</i> }}</code></p> <p>Example: <code>{{ ToMillis .StartTimeUnix }}</code></p> <p>Assuming the value of <code>{{ .StartTimeUnix }}</code> is 1672531199, the final output would be 1672531199000.</p>
Capitalize	<p>The function can be used to capitalizes the first letter of a string.</p> <p>Format: <code>{{ Capitalize <i>variable_string</i> }}</code></p> <p>Example: <code>{{ Capitalize .variable_string }}</code></p> <p>Assuming the value of <code>{{ .variable_string }}</code> is hello world, the final output would be Hello world.</p>
UrlEncode	<p>The function can be used to URL encode a string.</p> <p>Format: <code>{{ UrlEncode <i>.variable</i> }}</code></p>

Function	Description
	<p>Example: <code>{{ UrlEncode .variable}}</code></p> <p>Assuming the value of <code>{{.variable_string}}</code> is <code>hello world!</code>, the final output would be <code>hello%20world%21</code>.</p>


Elements

The templates offers the following main elements that help define the integration interactions, ensuring that the template can seamlessly connect with the third-party system and process data based on integration requirements.

The following example shows the main structure of the template.

```
<Information>
  <Scenarios>
    <Scenario Id="1" Type="REST">
      <Parameters>
        <Parameter Name="example" Value="example"></Parameter>
        ...
      </Parameters>
      <Requests>
        <Request Name="ExampleRequest" Method="GET" URL="https://api.example.com/getuser">
          <Outputs>
            <Output Name="ContactId" Path="data.#.id" Type="string"></Output>
            ...
          </Outputs>
        </Request>
      </Requests>
    </Scenario>
  </Scenarios>
</Information>
```

Main Element	Description
Information	<p>The <code><Information></code> element is a root element of the XML template, providing the basic information and configuration overview through its attributes. It also contains the <code><Scenarios></code> child element to describe the functional scenarios and integration capabilities of the template.</p> <p>For the details of the attributes, see Template property.</p>
Scenarios	<p>The <code><Scenarios></code> element provides a collection of <code><Scenario></code> child elements, which contains further child elements, such as</p>

Main Element	Description
	<p><Parameters>, <Requests>, and <Outputs>, to describe specific functional scenarios for the integration. The PBX will load the configured scenarios and execute the HTTP requests or operations defined within it.</p> <p>The template supports the following scenarios.</p> <ul style="list-style-type: none"> • Authentication scenario • User association scenario • Automatic contact synchronization scenario • Automatic contact creation scenario • Automatic ticket creation scenario (for Helpdesk system only) • Call journal scenario <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Note: If the specified scenarios do not meet your requirements, you can manually edit the template configurations for supplementary usage. For more information, see Extended configurations. </div>
Parameters	<p>The <Parameters> element is used to define a collection of parameters that are associated with the scenarios or requests.</p> <p>These parameters typically include dynamic values or configuration settings required for the execution of the scenario. They can be passed in the request URL, body, or headers, depending on the use case.</p>
Requests	<p>This <Requests> element provides a collection of requests for the scenarios to implement specific features.</p>
Outputs	<p>The <Outputs> element contains a list of <Output> elements that define the data to be extracted from the API response.</p>

Template property

The <Information> element provides the properties of the template, including the basic template information and configuration overview, as listed below:

Attribute	Description
Provider	The type of the template, with a fixed value of <code>crm</code> or <code>helpdesk</code> .
Name	The name of the template.
Key	Unique identifier for the template.
Logo	The filename of the logo.

Attribute	Description
Remark	A description or remark about the template.
Version	The version of the template. E.g. 1.0.0.
AuthType	The authentication method used in the integration. <ul style="list-style-type: none"> • none: None authentication. • basic: Basic authentication. • oauth2: OAuth2 authentication.
MaxConcurrentRequest	The maximum number of concurrent HTTP requests allowed to the CRM/Helpdesk.
UserAssociation	Whether the User Association feature is enabled or not. <ul style="list-style-type: none"> • 0: Disabled. • 1: Enabled.
CallJournal	Whether the Call Journal feature is enabled or not. <ul style="list-style-type: none"> • 0: Disabled. • 1: Enabled.
CreateNewContact	Whether the Create New Contact feature is enabled or not. <ul style="list-style-type: none"> • 0: Disabled. • 1: Enabled.
CreateNewTicket	Whether the Create New Ticket Automatically feature is enabled or not. <ul style="list-style-type: none"> • 0: Disabled. • 1: Enabled.

Authentication scenario

This scenario defines the authentication method used by the CRM/Helpdesk system. The structure of the scenario and its key elements are outlined as below:

```
<Scenario Id="AuthMethod" Type="AUTH">
  <Presets></Presets>
  <Parameters>
    <Parameter Name="AuthMethod" Value="oauth2"></Parameter>
    <Parameter Name="AuthEndPoint"
Value="https://api.example.com/oauth/v2/auth?client_id={{ .client_id }}"></Parameter>
    <Parameter Name="TokenEndPoint"
Value="https://api.example.com/oauth/v2/token?client_id={{ .client_id }}&
;client_secret={{ .client_secret }}"></Parameter>
```

```

<Parameter Name="AdditionalQueryString"></Parameter>
<Parameter Name="Scope"
Value="contacts.read,contacts.write,calls.read,calls.write,user.read,user.write"></Parameter>
<Parameter Name="CredentialType"></Parameter>
<Parameter Name="Base64EncodedCredential"></Parameter>
<Parameter Father="CustomFieldList" Name="client_id" Editor="password"
Title="Client ID"></Parameter>
<Parameter Father="CustomFieldList" Name="client_secret"
Editor="password" Title="Client Secret"></Parameter>
</Parameters>
<Requests></Requests>
</Scenario>

```


Here is a detailed explanation of the specific elements within the scenario.

Scenario element

The `<Scenario>` element has the following attributes:



Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>AuthMethod</code> .
Type	The type of the scenario, the value is <code>AUTH</code> , which means the scenario will execute authentication operations.





Parameter element

Attribute	Description
Name	The name of the authentication parameter .
Value	The value of the authentication parameter .
Father	<p>The name of a parent parameter (should be fixed as <code>CustomFieldList</code>), which serves as a container for the following child parameters, organizing custom fields that allow users to input values for the associated parameters in the configuration UI.</p> <div style="border: 1px solid #007bff; padding: 10px; margin-top: 10px;"> <p> Note: The variables captured in the custom fields are accessible for all the subsequent requests in the template.</p> <ul style="list-style-type: none"> • <code>Name</code>: The variable name of the custom field, which can be reference in the template. • <code>Editor</code>: The type of the custom field. </div>

Attribute	Description
	<p>Valid value:</p> <ul style="list-style-type: none"> ◦ <code>string</code>: A standard input field where users can enter plain text. ◦ <code>password</code>: A specialized input field for sensitive information, with text masked for privacy. ◦ <code>list</code>: A selectable list that allows users to choose from a pre-defined list of options. • <code>Title</code>: The title of the custom field. • <code>Value</code>: The pre-defined list of options.

Authentication parameters

Parameter	Description
AuthMethod	<p>Authentication method.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • <code>none</code>: This method is used when there is no authentication required, or when the CRM/Helpdesk only requires relevant variables for data exchange, such as an API Key or a webhook URL. • <code>basic</code>: This method authenticates requests using credentials, such as username and password or an API key. These credentials are base64-encoded into a basic string and included in a header field of every HTTP request sent to the CRM/Helpdesk. • <code>oauth2</code>: This method uses tokens to grant access without sharing credentials. It requires obtaining the necessary header and other parameters (e.g. OAuth access token) using specific API requests to the CRM/Helpdesk.
AuthEndPoint	<p>The CRM/Helpdesk authorization URL that needs to be opened to initiate the authorization process.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #0070C0;"> <p> Note: This parameter is only available for <code>oauth2</code> authentication method.</p> </div>
TokenEndPoint	<p>The CRM/Helpdesk API request URL for getting access token and refresh token.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #0070C0;"> <p> Note: This parameter is only available for <code>oauth2</code> authentication method.</p> </div>

Parameter	Description
AdditionalQueryString	<p>If the CRM/Helpdesk system authenticates requests using query parameters instead of scope, you can add the query variables in this parameter.</p> <p> Note: This parameter is only available for <code>oauth2</code> authentication method.</p>
Scope	<p>Specify the scope of CRM/Helpdesk data that the PBX is allowed to access.</p> <p> Note: This parameter is only available for <code>oauth2</code> authentication method.</p>
CredentialType	<p>The type of the credential.</p> <p> Note: This parameter is only available for <code>basic</code> authentication method.</p> <p>Valid value:</p> <ul style="list-style-type: none"> <code>username_password</code> <code>api_key</code>
Base64EncodedCredential	<p>Define the combination format of the credential variables, which will be used to generate the basic authentication string.</p> <p> Note: This parameter is only available for <code>basic</code> authentication method.</p> <p>E.g. <code>{{.username:.password}}</code> or <code>{{.api_key:}}</code>.</p>

User association scenario

This scenario is used to retrieve the list of users from the CRM/Helpdesk system. The structure of the scenario and its key elements are outlined as below:

```
<Scenario Id="UserAssociation" Type="REST">
  <Parameters></Parameters>
  <Requests>
```

```

<Request Name="UserAssociation" Method="GET"
ResponseType="application/json" RequestEncoding=" "
URLFormat="https://www.api.example.com/v1/users?type=ActiveUsers">
  <Parameters></Parameters>
  <Outputs>
    <Output Name="UserUniqueId" Path="users.#.id" Type=""></Output>
    <Output Name="FirstName" Path="users.#.First_Name" Type=""></Output>
    <Output Name="LastName" Path="users.#.Last_Name" Type=""></Output>
    <Output Name="Email" Path="users.#.Email" Type=""></Output>
  </Outputs>
</Request>
</Requests>
</Scenario>

```

Here is a detailed explanation of the specific elements within the scenario.

Scenario element

The `<Scenario>` element has the following attributes:

Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>UserAssociation</code> .
Type	The type of the scenario, the value is <code>REST</code> , which means the scenario will execute a REST API using an HTTP request.

Request element

The `<Request>` element has the following attributes:

Attribute	Description
Name	The name of the request, which helps to identify the request within the scenario. The value should be fixed as <code>UserAssociation</code> .
Method	The HTTP method to be used for the request. In this scenario, the default HTTP method is <code>GET</code> , you can modify it if necessary.
ResponseType	The expected response format for the request, which should be in JSON format (<code>application/json</code>).
RequestEncoding	How the request should be encoded. In this scenario, the value is empty.
URLFormat	The request URL of the API endpoint.

Output element

The `<Output>` element has the following attributes:

Attribute	Description
Name	The name of the output variable, which represents the variables need to be retrieved from the response data.
Path	The location of the desired data field within the API response, using a path syntax to access nested data (e.g. <code>users.#.id</code>).
Type	The data type of the output variable. In this template, the type is fixed as <code>string</code> .

Variables

Variable	Description
UserUniqueld	The unique ID of the CRM/Helpdesk user.
FirstName	The first name of the user.
LastName	The last name of the user.
Email	The email of the user.

Automatic contact synchronization scenario

This scenario is used to search for contacts in the CRM/Helpdesk system, and the returned information can be used to implement CRM contacts searching, automatic contact synchronization and call popup. The structure of the scenario and its key elements are outlined as below:

```
<Scenario Id="SyncContactAuto" Type="REST">
  <Presets></Presets>
  <Parameters>
    <Parameter Name="ContactUrlType"
Value="retrieve_from_contact"></Parameter>
    <Parameter Name="URLFormat"></Parameter>
    <Parameter Name="ContactFieldForUri"
Value="data.#.contactUrl"></Parameter>
    <Parameter Name="ContactsIdEnable" Value="1"></Parameter>
    <Parameter Name="FirstNameEnable" Value="1"></Parameter>
    <Parameter Name="BusinessNumberEnable" Value="1"></Parameter>
    <Parameter Name="CustomValueEnable" Value="0"></Parameter>
  </Parameters>
  <Requests>
    <Request Name="Contacts" Method="GET" ResponseType="application/json"
RequestEncoding=" "
URLFormat="https://www.api.example.com/v1/Contacts/search?criteria=((Phone:equals:{{.Phone}})or(Home_Phone:equals:{{.Phone}})or(Mobile:equals:{{.Phone}})or(Asst_Phone:equals:{{.Phone}}))">
```

```

<Parameters></Parameters>
<Outputs>
  <Output Name="ContactsId" Path="data.#.id" Type=""></Output>
  <Output Name="FirstName" Path="data.#.First_Name" Type=""></Output>
  <Output Name="BusinessNumber" Path="data.#.Phone" Type=""></Output>
  ...
  <Output Name="CustomValue" Path="data.#.Owner" Type=""></Output>
</Outputs>
</Request>
</Requests>
</Scenario>

```

Here is a detailed explanation of the specific elements within the scenario.

Scenario element

The `<Scenario>` element has the following attributes:

Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>SyncContactAuto</code> .
Type	The type of the scenario, the value is <code>REST</code> , which means the scenario will execute a REST API using an HTTP request.


Parameter element


The `<Parameter>` element provides has the following attributes:

Attribute	Description
Name	The name of the configuration settings used in the scenario.
Value	The value of the configuration settings.

Configuration settings

Setting	Description
ContactUrlType	<p>Specify the method for retrieving the contact URL for call popup.</p> <p>Valid value:</p> <ul style="list-style-type: none"> <code>specify_url_format</code>: Configure the call popup URL by specifying the URL format, where you can use variables to dynamically construct the URL.

Setting	Description
	<ul style="list-style-type: none"> • <code>retrieve_from_contact</code>: Extract the contact URL directly from a specific field in the contact search response, where the contact URL can be retrieved.
URLFormat	<p>The expression of the Contact URL. The following variables can be used to construct the URL.</p> <ul style="list-style-type: none"> • The variables captured from the custom field inputs in the authentication process. • <code>ContactSyncType</code>: The contact type, which is retrieve from the name of the request used to search specific types of contact. • <code>ContactId</code>: The contact ID, which is retrieved from the response data in this scenario. • <code>CustomValue</code>: A custom field, which is retrieved from the response data in this scenario. <p>For example,</p> <pre> {{.crm_url}}/crm/{{.CustomValue}}/ tab/{{[Capitalize .ContactSyncType]}}/{{.ContactId}}. </pre> <div style="border: 1px solid #007bff; padding: 10px; margin-top: 10px;">  <p>Note: If the provided variables do not meet your requirements, you can define additional requests and retrieve the corresponding values for your desired variables. For more information, see Extended Configurations - Retrieve Additional Variables.</p> </div>
ContactFieldForUri	<p>The JSON path used to retrieve the corresponding field for the contact URL from the response data.</p> <p>For example, <code>.data.#.contactUrl</code>.</p>
ContactsIdEnable	<p>Whether to retrieve contact's ID from the response data. The value should be set to <code>1</code>.</p>
FirstNameEnable	<p>Whether to retrieve the contact's first name from the response data. The value should be set to <code>1</code>.</p>
LastNameEnable	<p>Whether to retrieve the contact's last name from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • <code>0</code>: Do not retrieve the data. • <code>1</code>: Retrieve the data.
CompanyEnable	<p>Whether to retrieve the contact's company from the response data.</p>


Setting	Description
	<p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
EmailEnable	<p>Whether to retrieve the contact's email from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
BusinessNumberEnable	<p>Whether to retrieve the contact's business phone number from the response data.</p> <div data-bbox="586 743 1299 852" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px;"> <p> Note: At least one phone number is required.</p> </div> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
BusinessNumber2Enable	<p>Whether to retrieve the contact's second business phone number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
BusinessFaxNumberEnable	<p>Whether to retrieve the contact's business fax number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
MobileNumberEnable	<p>Whether to retrieve the contact's mobile number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
MobileNumber2Enable	<p>Whether to retrieve the contact's second mobile number from the response data.</p> <p>Valid value:</p>

Setting	Description
	<ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
HomeNumberEnable	<p>Whether to retrieve the contact's home number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
HomeNumber2Enable	<p>Whether to retrieve the contact's second home number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
HomeFaxNumberEnable	<p>Whether to retrieve the contact's home fax number from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
OtherNumberEnable	<p>Whether to retrieve the contact's other type of number (other than business number or home number) from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
ZipCodeEnable	<p>Whether to retrieve the zip code from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
StreetEnable	<p>Whether to retrieve the street from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.
CityEnable	<p>Whether to retrieve the city from the response data.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Do not retrieve the data. • 1: Retrieve the data.

Setting	Description
StateEnable	Whether to retrieve the state/province from the response data. Valid value: <ul style="list-style-type: none"> 0: Do not retrieve the data. 1: Retrieve the data.
CountryEnable	Whether to retrieve the country from the response data. Valid value: <ul style="list-style-type: none"> 0: Do not retrieve the data. 1: Retrieve the data.
RemarkEnable	Whether to retrieve the remark for the contact from the response data. Valid value: <ul style="list-style-type: none"> 0: Do not retrieve the data. 1: Retrieve the data.
CustomValueEnable	Whether to enable a custom variable, which can be used to construct the Contact's URL . Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.

Request element

The <Request> element has the following attributes:

Attribute	Description
Name	The name of the request, which helps to identify the request within the scenario. <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;">  Note: The value should be the actual name of the specific contact type to be searched in the CRM/Helpdesk system, and will be referenced as a <code>ContactSyncType</code> variable. </div>
Method	The HTTP method to be used for the request. In this scenario, the default HTTP method is <code>GET</code> , you can modify it if necessary.
ResponseType	The expected response format for the request, which should be in JSON format (<code>application/json</code>).


Attribute	Description
RequestEncoding	How the request should be encoded. In this scenario, the value is empty.
URLFormat	The request URL of the API endpoint.

Output element

The `<Output>` element has the following attributes:

Attribute	Description
Name	The name of the output variable, which represents the variables need to be retrieved from the response data.
Path	The location of the desired data field within the API response, using a path syntax to access nested data (e.g. <code>users.#.id</code>).
Type	The data type of the output variable. In this template, the type is fixed as <code>string</code> .

Variables

Variable	Description
ContactsId	(Required) The unique ID of the contact.
FirstName	(Required) The first name of the contact.
LastName	The last name of the contact.
Company	The company of the contact.
Email	The email of the contact.
BusinessNumber	The business phone number of the contact. <div style="border-left: 2px solid #0070c0; padding-left: 10px; margin-top: 10px;">  Note: At least one phone number is required. </div>
BusinessNumber2	The second business phone number of the contact.
BusinessFaxNumber	The business fax number of the contact.
MobileNumber	The mobile number of the contact.
MobileNumber2	The second mobile number of the contact.
HomeNumber	The home number of the contact.

Variable	Description
HomeNumber 2	The second home number of the contact.
HomeFaxNumber	The home fax number of the contact.
OtherNumber	The other type of phone number of the contact.
ZipCode	Zip code.
Street	Street.
City	City.
State	State or province.
Country	Country.
Remark	Remark.
CustomValue	A custom variable, which is used to construct the Contact's URL .

Automatic contact creation scenario

This scenario is used to create a new contact in the CRM/Helpdesk system when the caller's number does not match any existing contact. The structure of the scenario and its key elements are outlined as below:

```
<Scenario Id="CreateNewContact" Type="REST">

  <Parameters></Parameters>
  <Requests>
    <Request Name="Contacts" Method="POST" ResponseType="application/json"
    RequestEncoding="" URLFormat="https://www.api.example.com/v1/Contacts">
      <Parameters>
        <Parameter Name="Data" Type="Body" Value="{ "data":
        [{"Last_Name": "{ .LastName} }", "First_Name": "{ .FirstName} }", "Phone": "{ .BusinessNumber} }"; ] } "></Parameter>
      </Parameters>
      <Outputs></Outputs>
    </Request>
  </Requests>
</Scenario>
```

Here is a detailed explanation of the specific elements within the scenario.


Scenario element

The `<Scenario>` element has the following attributes:

Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>CreateNewContact</code> .
Type	The type of the scenario, the value is <code>REST</code> , which means the scenario will execute a REST API using an HTTP request.


Request element


The `<Request>` element has the following attributes:

Attribute	Description
Name	The name of the request, which helps to identify the request within the scenario. <div style="border-left: 2px solid #007bff; padding-left: 10px; margin-top: 10px;">  Note: The value should be the actual name of the specific contact type to be created in the CRM/Helpdesk system. </div>
Method	The HTTP method to be used for the request. In this scenario, the HTTP method is <code>POST</code> .
ResponseType	The expected response format for the request, which should be in JSON format (<code>application/json</code>).
RequestEncoding	How the request should be encoded. In this scenario, the value is empty.
URLFormat	The request URL of the API endpoint.

Parameter element

The `<Parameter>` element nested within the `<Request>` element defines the parameters included in the request body.

Attribute	Description
Name	The name of the parameter, which should be fixed as <code>Data</code> .
Type	The type of the parameter, which should be fixed as <code>Body</code> .
Value	Defines the required parameters to be passed through the request body, within a <code>data</code> structure. The parameter values can be retrieved from the pre-defined variables in the PBX system. <div style="border-left: 2px solid #007bff; padding-left: 10px; margin-top: 10px;">  Note: </div>

Attribute	Description
	 The format should follows XML parameter standards. For example, if a line break is required, you must use the escape character <code>&#xA;</code> to represent a newline.

Variables



Note:

Only required variables are listed here. If you need additional parameters and variables, see [Extended Configurations - Retrieve Additional Variables](#).

Variable	Description
{{.LastName}} }	The last name of the contact.
{{.FirstName}} }}	The first name of the contact.
{{.BusinessNumber}}	The phone number of the contact.

Automatic ticket creation scenario (for Helpdesk system only)

This scenarios is used to create a new ticket in the Helpdesk system. The structure of the scenario and its key elements are outlined as below:

```

<Scenario Id="CreateNewTicket" Type="REST" >
  <Presets></Presets>
  <Parameters>
    <Parameter Name="EnableSubject" Value="1"></Parameter>
    <Parameter Name="Subject" Value="{{.Communication_Type}}
    {{.Call_Status}} - from {{.Call_From}} to {{.Call_To}}"></Parameter>
    <Parameter Name="EnableDescription" Value="1"></Parameter>
    <Parameter Name="Description" Value="{{.Time}} {{.Communication_Type}}
    {{.Call_Status}} - from {{.Call_From}} to {{.Call_To}}
    {{.Talk_Duration}}"></Parameter>
  </Parameters>
  <Requests>
    <Request Name="CreateNewTicket" Method="POST" Weight="0"
    ResponseType="application/json" RequestEncoding=""
    URLFormat="https://{{.domain}}/api/v1/tickets">
      <Parameters>
    
```

```

    <Parameter Name="Data" Type="Body"
    Value="{ "subject": "{{ .Subject}}", "contactId": "{{ .ContactId}}", "phone": "{{ .
ContactNumber}}", "description": "{{ .Description}}"; }"></Parameter>
  </Parameters>
  <Outputs></Outputs>
</Request>
</Requests>
</Scenario>

```

Here is a detailed explanation of the specific elements within the scenario.

Scenario element


The `<Scenario>` element has the following attributes:


Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>CreateNewTicket</code> .
Type	The type of the scenario, the value is <code>REST</code> , which means the scenario will execute a REST API using an HTTP request.

Parameter element

Attribute	Description
Name	The name of the configuration settings used in the scenario.
Value	The value of the configuration settings.

Configuration settings

Settings	Description
EnableSubject	Whether to allow users to customize the subject of the ticket. Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.
Subject	Specify the default value of the subject. <div style="border-left: 2px solid #007bff; padding-left: 10px; margin-top: 10px;"> <p> Note: The default value is <code>{{ .Communication_Type }}</code> <code>{{ .Call_Status }}</code> - from <code>{{ .Call_From }}</code> to <code>{{ .Call_To }}</code>. For more supported variables, see the parameter variable list.</p> </div>

Settings	Description
EnableDescription	<p>Whether to allow users to customize the description of the ticket.</p> <p>Valid value:</p> <ul style="list-style-type: none"> • 0: Disable. • 1: Enable.
Description	<p>Specify the default value of the description.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note: The default value is <code>{{.Time}} {{.Communication_Type}} {{.Call_Status}} - from {{.Call_From}} to {{.Call_To}} {{.Talk_Duration}}</code>. For more supported variables, see the parameter variable list.</p> </div>

Request element


The `<Request>` element has the following attributes:

Attribute	Description
Name	The name of the request, which helps to identify the request within the scenario. The value should be fixed as <code>CreateNewTicket</code> .
Method	The HTTP method to be used for the request. In this scenario, the HTTP method is <code>POST</code> .
ResponseType	The expected response format for the request, which should be in JSON format (<code>application/json</code>).
RequestEncoding	How the request should be encoded. In this scenario, the value is empty.
URLFormat	The request URL of the API endpoint.

Parameter element

The `<Parameter>` element nested within the `<Request>` element defines the parameters included in the request body.

Attribute	Description
Name	The name of the parameter, which should be fixed as <code>Data</code> .
Type	The type of the parameter, which should be fixed as <code>Body</code> .
Value	Defines the required parameters to be passed through the request body. The parameter values can be retrieved from the pre-defined variables in the PBX system.

Attribute	Description
	 Note: The format should follows XML parameter standards. For example, if a line break is required, you must use the escape character <code>&#xA;</code> to represent a newline.


Variables



Note:

Only required variables are listed here. If you need additional parameters and variables, see Extended Configurations - Retrieve Additional Variables.

Variable	Description
Variables obtained from the AuthMethod scenario	The variables captured from the user input for authentication process.
{{.Subject}}	The subject of the ticket. This variable value is retrieved from the Subject parameter .
{{.Description}}	The description of the ticket. The variable value is retrieved from the Description parameter .
{{.Time}}	The time the call was made or received (in PBX system's time format).
{{.StartTime Unix}}	The Unix timestamp of the call start time.
{{.EndTime Unix}}	The Unix timestamp of the call end time.
{{.Call_From}}	The name and number of the calling party that initiated the call. Format: Caller's name <Caller's number>.
{{.CallerName}}	Caller's name.
{{.CallerNumber}}	Caller's number.
{{.CalleeName}}	Callee's name.
	Callee's number.

Variable	Description
{{.CalleeNumber}}	
{{.CrmlId}}	The unique ID of the CRM/Helpdesk user associated with a PBX extension.
{{.ExtensionFirstName}}	The first name of the extension user.
{{.ExtensionLastName}}	The last name of the extension user.
{{.ExtensionEmail}}	The email of the extension user.
{{.ContactSyncType}}	The type of the contact.
{{.ContactNumber}}	The number of the contact.
{{.ContactId}}	The unique ID of a contact.
{{.TicketId}}	The unique ID of the ticket created based on a call. <div style="border: 1px solid #ccc; background-color: #f0f8ff; padding: 10px; margin-top: 10px;">  <p>Note: To get the ticket ID, you need to configure additional settings in the template. For more information, see Extended Configuration - Retrieve Ticket ID.</p> </div>
{{.CallDuration}}	The time between the call started and the call ended.
{{.Talk_Duration}}	The time between the call answered and the call ended (in a format of <code>hh:mm:ss</code>).
{{.Talk_Duration_Sec}}	The time between the call answered and the call ended (in seconds).
{{.Communication_Type}}	The direction of the call. Possible value: <ul style="list-style-type: none"> • <code>Inbound</code>: Inbound call. • <code>Outbound</code>: Outbound call.
{{.Call_Status}}	The processing status of the call. Possible value: <ul style="list-style-type: none"> • <code>Missed</code>: A call that was not answered.

Variable	Description
	<ul style="list-style-type: none"> • Completed: A call that was successfully connected and completed.
{{.Call_Log_Status}}	<p>The status of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • Missed Call: A call that was not answered, • Outgoing Call: An outbound call that was initiated by the system. • Incoming Call: An inbound call that was received by the system. • Voicemail: A call that was not answered and was directed to voicemail.
{{.CallDisposition}}	<p>The disposition of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • MISSED: The call was not answered. • VOICEMAIL: The call was sent to voicemail. • BUSY: The recipient was on another call and could not answer. • ANSWERED: The call was successfully answered. • NO ANSWER: The recipient did not answer the call.

Call journal scenario

This scenario is used to log the call activities of associated users to the CRM / Helpdesk. The structure of the scenario and its key elements are outlined as below:

```
<Scenario Id="CallJournal" Type="REST">
  <Presets></Presets>
  <Parameters>
    <Parameter Name="EnableSubject" Value="1"></Parameter>
    <Parameter Name="Subject" Value="Extension Call"></Parameter>
    <Parameter Name="EnableDescription" Value="1"></Parameter>
    <Parameter Name="Description" Value="Call: {{.Time}}
    {{.Call_Log_Status}} from {{.Call_From}} to {{.Call_To}}
    {{.Talk_Duration}}"></Parameter>
    <Parameter Name="EnablePlayCallRecording" Value="1"></Parameter>
    <Parameter Name="PlayCallRecording" Value="0"></Parameter>
  </Parameters>
  <Requests>
    <Request Name="CallJournal" Method="POST" Weight="0"
    ResponseType="application/json" RequestEncoding=""
    URLFormat="https://www.api.example.com/crm/v2/Calls">
      <Parameters>
```

```

<Parameter Name="Data" Type="Body"
Value="{ "data"[{"{{ .Owner }}{{ .WhoModule }}":
{"Description": "{{ .Description}}", "Voice_Recording__s": "{{ .RecordPath}}", "
Call_Start_Time": "{{ TimeFormat .StartTimeUnix "yyyy-MM-ddTHH:mm:ss-z"
"1"}}", "Subject&": "{{ .Subject}}", "Call_Type":
"Inbound", "Call_Result": "{{ .Call_Log_Status}}", "Call_Duration":
"{{ .Talk_Duration_Sec}}"}]}]"></Parameter>
</Parameters>
<Outputs></Outputs>
</Request>
</Requests>
</Scenario>

```

Here is a detailed explanation of the specific elements within the scenario.

Scenario element

The <Scenario> element has the following attributes:


Attribute	Description
Id	The unique identifier of the scenario, the value should be fixed as <code>CallJournal</code> .
Type	The type of the scenario, the value is <code>REST</code> , which means the scenario will execute a REST API using an HTTP request.



Parameter element

Attribute	Description
Name	The name of the configuration settings used in the scenario.
Value	The value of the configuration settings.

Configuration settings

Settings	Description
EnableSubject	Whether to allow users to customize the subject of the call log. Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.
Subject	Specify the default value of the subject.

 **Note:**

Settings	Description
	 The default value is <code>Extension Call</code> , you can also specify the subject using the supported variables .
EnableDescription	Whether to allow users to customize the description of the call log. Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.
Description	Specify the default value of the description.  Note: The default value is <code>{{.Time}} {{.Communication_Type}} {{.Call_Status}} - from {{.Call_From}} to {{.Call_To}} {{.Talk_Duration}}</code> . For more supported variables, see the parameter variable list .
EnablePlayCallRecording	Whether to allow users to configure the option for playing call recordings in the CRM. Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.
PlayCallRecording	Specify the default status of the call recording playback option. Valid value: <ul style="list-style-type: none"> 0: Disable. 1: Enable.

Request element


The `<Request>` element has the following attributes:

Attribute	Description
Name	The name of the request, which helps to identify the request within the scenario. The value should be fixed as <code>CallJournal</code> .
Method	The HTTP method to be used for the request. In this scenario, the HTTP method is <code>POST</code> .
ResponseType	The expected response format for the request, which should be in JSON format (<code>application/json</code>).
RequestEncoding	How the request should be encoded. In this scenario, the value is empty.

Attribute	Description
URLFormat	The request URL of the API endpoint.

Parameter element

The `<Parameter>` element nested within the `<Request>` element defines the parameters included in the request body.

Attribute	Description
Name	The name of the parameter, which should be fixed as <code>Data</code> .
Type	The type of the parameter, which should be fixed as <code>Body</code> .
Value	<p>Defines the required parameters to be passed through the request body. The parameter values can be retrieved from the pre-defined variables in the PBX system.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note: The format should follow XML parameter standards. For example, if a line break is required, you must use the escape character <code>&#xA;</code> to represent a newline.</p> </div>


Variables



Note:

Only required variables are listed here. If you need additional parameters and variables, see [Extended Configurations - Retrieve Additional Variables](#).

Variable	Description
Variables obtained from the <code>AuthMethod</code> scenario	The variables captured from the user input for authentication process.
<code>{{.Subject}}</code>	<p>The subject of the call.</p> <p>This variable value is retrieved from the Subject parameter.</p>
<code>{{.Description}}</code>	<p>The description of the call.</p> <p>The variable value is retrieved from the Description parameter.</p>
<code>{{.Time}}</code>	The time the call was made or received (in PBX system's time format).

Variable	Description
{{.StartTime Unix}}	The Unix timestamp of the call start time.
{{.EndTimeU nix}}	The Unix timestamp of the call end time.
{{.Call_From }}	The name and number of the calling party that initiated the call. Format: Caller's name <Caller's number>.
{{.CallerNam e}}	Caller's name.
{{.CallerNum ber}}	Caller's number.
{{.CalleeNa me}}	Callee's name.
{{.CalleeNu mber}}	Callee's number.
{{.Crmlid}}	The unique ID of the CRM/Helpdesk user associated with a PBX extension.
{{.Extension FirstName}}	The first name of the extension user.
{{.Extension LastName}}	The last name of the extension user.
{{.Extension Email}}	The email of the extension user.
{{.ContactSy ncType}}	The type of the contact.
{{.ContactNu mber}}	The number of the contact.
{{.ContactId} }}	The unique ID of a contact.
{{.TicketId}}	The unique ID of the ticket created based on a call. <div style="border-left: 2px solid #007bff; padding-left: 10px; margin-top: 10px;">  Note: To get the ticket ID, you need to configure additional settings in the template. For more information, see Extended Configuration - Retrieve Ticket ID. </div>
{{.CallDurati on}}	The time between the call started and the call ended.

Variable	Description
{{.Talk_Duration}}	The time between the call answered and the call ended (in a format of hh:mm:ss).
{{.Talk_Duration_Sec}}	The time between the call answered and the call ended (in seconds).
{{.Communication_Type}}	<p>The direction of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • <code>Inbound</code>: Inbound call. • <code>Outbound</code>: Outbound call.
{{.Call_Status}}	<p>The processing status of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • <code>Missed</code>: A call that was not answered. • <code>Completed</code>: A call that was successfully connected and completed.
{{.Call_Log_Status}}	<p>The status of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • <code>Missed Call</code>: A call that was not answered, • <code>Outgoing Call</code>: An outbound call that was initiated by the system. • <code>Incoming Call</code>: An inbound call that was received by the system. • <code>Voicemail</code>: A call that was not answered and was directed to voicemail.
{{.CallDisposition}}	<p>The disposition of the call.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • <code>MISSED</code>: The call was not answered. • <code>VOICEMAIL</code>: The call was sent to voicemail. • <code>BUSY</code>: The recipient was on another call and could not answer. • <code>ANSWERED</code>: The call was successfully answered. • <code>NO ANSWER</code>: The recipient did not answer the call.
{{.EnableCallRecording}}	<p>Whether the call recording is enabled.</p> <p>Possible value:</p> <ul style="list-style-type: none"> • <code>0</code>: Disabled. • <code>1</code>: Enabled.

Extended configurations

In addition to the scenarios specified in the configuration page, you can also manually add custom configurations for supplementary use.



Important:

After modification, if the template is updated via the PBX web portal, the custom configurations might be overwritten unexpectedly.

Set custom request headers

If custom request headers are needed, you can add parameter(s) with the type set to `Header`. This will add the headers to the request body.

As shown in the following example, a `X-Auth-Token` header is used to pass the API key. The `Father` attribute (fixed as `PopulateTemplateString`) groups a set of related parameters, while the `Type` (fixed as `Header`) indicates that the parameter is intended to be passed as an HTTP header. The `Name` and `value` specify the header's name and value. In this case, the header included in the request would be `X-Auth-Token: {{ .api_key }}`.

```
<Requests>
  <Request Name="contacts" Method="GET" ResponseType="application/json" URLFormat="https://api.example.com/contacts/search?number={{ .Phone }}">
    <Parameters>
      <Parameter Father="PopulateTemplateString" Name="X-Auth-Token" Type="Header" Value="{{ .api_key }}" />
    </Parameters>
    <Outputs>
      ...
    </Outputs>
  </Request>
```

Identify token errors

In the `AuthMethod` scenario, you can add a parameter `TokenInvalid` to help identify whether the issue in the request is related to the token or other underlying problems.

You can refer to the following example, and change the `value` to the token-related error code or specific content. If the HTTP status code returned is not

200 and the response body contains the content specified in `Value`, the error will be recognized as **TokenInvalid**.

```
<Scenario Id="AuthMethod" Type="AUTH">
  <Presets />
  <Parameters>
    <Parameter Name="AuthMethod" Value="oauth2" />
    ...
    <Parameter Father="TokenErrorMap" Type="TokenInvalid" Name="TokenInvalid" Value="INVALID_TOKEN" />
    ...
  </Parameters>
  <Requests />
  <Responses />
</Scenario>
```

Retrieve additional variables for global use

If you require additional variables beyond the provided ones, if you want to use certain common information globally during the integration process, you can define a `GetGlobalInfo` scenario and set up the related requests to retrieve the necessary values.

When the integration is completed or the PBX is started, requests set within the `GetGlobalInfo` scenario will be sent to the CRM/Helpdesk system, and the required variables will be stored in the `Output` elements, with their names directly corresponding to the variable names, making them accessible for reference in other scenarios (excluding the authentication scenario).

As shown in the following example, the `GetGlobalInfo` scenario contains requests for retrieving essential information `DepartmentId` and `PortalName`.

```
<Scenario Id="GetGlobalInfo" Type="REST">
  <Presets />
  <Parameters />
  <Requests>
    <Request Name="GetDepartments" Method="GET" ResponseType="application/json" URLFormat="https://desk.example.{{.domain_suffix}}/api/v1/departments">
      <Parameters />
      <Outputs>
        <Output Name="DepartmentId" Path="data.#.id" />
      </Outputs>
    </Request>
```

```

        <Request Name="GetOrganizations" Method="GET" ResponseType="application/json" URLFormat="https://desk.example.{{.domain_suffix}}/api/v1/organizations">
            <Parameters />
            <Outputs>
                <Output Name="PortalName" Path="data.#(portalName!=").portalName" />
            </Outputs>
        </Request>
    </Requests>
</Scenario>

```

Supplement contact query information

If the initial query for contact information does not return all the necessary information you need, further queries may be required using the returned contact ID or other contact information.

As shown in the example below, in a scenario of contact synchronizing, the initial query request does not return the desired phone numbers.

```

<Scenario Id="SyncContactAuto" Type="REST">
    <Presets />
    <Parameters>
        <Parameter Name="ContactUrlType" Value="specify_url_format" />
        ...
    </Parameters>
    <Requests>
        <Request Name="contacts" Method="GET" ResponseType="application/json" URLFormat="https://{{.subdomain}}.example.com/api/v2/users?query=phone:{{.Phone}}&role[]=end-user">
            <Parameters />
            <Outputs Next="GetIdentities">
                <Output Name="ContactsId" Path="users.#.id" />
                <Output Name="FirstName" Path="users.#.name" />
                <Output Name="BusinessNumber" Path="users.#.phone" />
            >
            ...
            //The following phone number fields returns empty.

            <Output Name="BusinessNumber2" Path=""></Output>

            <Output Name="BusinessFaxNumber" Path=""></Output>
            <Output Name="MobileNumber" Path=""></Output>

```

```

    <Output Name="MobileNumber2" Path=""></Output>
    <Output Name="HomeNumber" Path=""></Output>
    <Output Name="HomeNumber2" Path=""></Output>
    ...
  </Outputs>
</Request>
</Requests>
</Scenario>

```

In this case, you can perform a further query to retrieve the desired numbers by chaining scenarios, as described in the following steps:

1. Add a Next Attribute in the original scenario

In the contact synchronizing scenario, add a `Next` attribute in the `<Outputs>` element of the initial query request. The attribute value must be the name of the desired request configured in the [Common scenario](#). In this example, the request name is `GetIdentities`.

```
<Outputs Next="GetIdentities">
```

2. Create a Common scenario

Create a scenario named `Common` with a type of `REST` (both the name and type are fixed), then add a request named `GetIdentities` to query additional contact information using the variable `{{ .ContactsId }}` obtained from the initial request.

```

<Scenario Id="Common" Type="REST">
  <Presets></Presets>
  <Parameters></Parameters>
  <Requests>
    <Request Name="GetIdentities" Method="GET" URLFormat="https://{{ .subdomain }}.example.com/api/v2/users/{{ .ContactsId }}/identities" ResponseType="application/json">
      <Parameters></Parameters>
      <Outputs>
        <Output Name="BusinessNumber2" Path="identities.#(type=="phone_number")#|1.value"></Output>
        <Output Name="BusinessFaxNumber" Path="identities.#(type=="phone_number")#|2.value"></Output>
        <Output Name="MobileNumber" Path="identities.#(type=="phone_number")#|3.value"></Output>
        <Output Name="MobileNumber2" Path="identities.#(type=="phone_number")#|4.value"></Output>
      </Outputs>
    </Request>
  </Requests>
</Scenario>

```



```

        <Output Name="HomeNumber" Path="identities.#(type=="phone_number")#|5.value"></Output>
        <Output Name="HomeNumber2" Path="identities.#(type=="phone_number")#|6.value"></Output>
        <Output Name="HomeFaxNumber" Path="identities.#(type=="phone_number")#|7.value"></Output>
        <Output Name="OtherNumber" Path="identities.#(type=="phone_number")#|8.value"></Output>
    </Outputs>
</Request>
</Requests>
</Scenario>

```

Since the request is referenced via the `NEXT` attribute in the original scenario, the variable from the initial query will be passed to the request. The request will execute and backfill the `Outputs` into the initial request's response to complete the information.

Retrieve TicketID for further operations

In a Helpdesk integration template, you can configure additional settings to retrieve the ticket ID for subsequent operations, such as updating the ticket or referencing the variable in call journal.

For example, when the ticket is created before the call (i.e. when the phone rings), it might be missing some information. In this case, you can update the information for the created ticket by using the `TicketId` in the corresponding call record later, as described in the following steps:

```

<Scenario Id="CreateNewTicket" Type="REST">
  <Presets />
  <Parameters>
    <Parameter Name="EnableSubject" Value="1" />
    ...
    <Parameter Name="NeedSyncContact" Value="1" />
  </Parameters>
  <Requests>
    <Request Name="CreateNewTicket" Method="POST" ResponseType="application/json" URLFormat="https://{{.domain}}/api/v1/tickets">
      <Parameters>
        <Parameter Name="Data" Type="Body" Value="{\"departmentId\":\"{{.DepartmentId}}\", \"subject\":\"{{.Subject}}\", \"contactId\":\"{{.ContactId}}\", \"phone\":\"{{.ContactNumber}}\", \"description\":\"{{.Description}}\"}" />
      </Parameters>
    </Request>
  </Requests>
</Scenario>

```

```

    </Parameters>
    <Outputs>
      <Output Name="TicketId" Path="id" />
    </Outputs>
  </Request>
  <Request Name="UpdateTicket" Method="POST" ResponseType=
"application/json" URLFormat="https://{{.domain}}/api/v1/tickets
/{{.TicketId}}/comments">
    <Parameters>
      <Parameter Name="Data" Type="Body" Value="{\"isPublic
\": \"true\", \"content\": \"{{.Description}}\"}" />
    </Parameters>
    <Outputs />
  </Request>
</Requests>
</Scenario>

```

1. Retrieve ticket ID

In the `CreateNewTicket` scenario, add an `<Output>` element to extract the ticket ID from the response of `CreateNewTicket` request.

```

<Outputs>
  <Output Name="TicketId" Path="id" />
</Outputs>

```

This allows the system to store the `TicketId` for future use.

2. Update the ticket

Add a request to update the ticket in the corresponding call record using the previously retrieved ticket ID.

```

<Request Name="UpdateTicket" Method="POST" ResponseType="ap
plication/json" URLFormat="https://{{.domain}}/api/v1/ticke
ts/{{.TicketId}}/comments">
  <Parameters>
    <Parameter Name="Data" Type="Body" Value="{\"isPublic\":
\"true\", \"content\": \"{{.Description}}\"}" />
  </Parameters>
  <Outputs />
</Request>

```

The ticket is updated with all necessary data after the initial creation.