How to Create or Modify Halo Tickets using a Rest API-based Bot

What's in this page:

Prerequisites

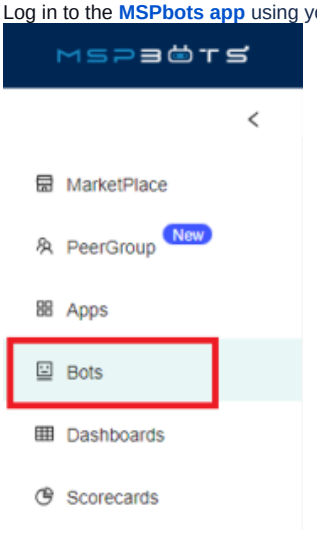
- (Optional) When you need to **update** tickets, it is crucial to clarify the following before proceeding with the configuration:
    - Determine which tickets meet the criteria for modification.
    - Due to Halo supports modifications for over 200 attribute fields, please specify which parameters and attribute names you want to modify for these tickets. Please refer to https://halo.haloservicedesk.com/apidoc/resources/tickets.
- (Optional) When you need to **create** tickets, it is essential to clarify the following before proceeding with the configuration:
    - Identify the field parameters and attribute names for the tickets you are creating. Please refer to https://halo.haloservicedesk.com/apidoc/resources/tickets.
- Create a widget or dataset to filter the data needed for the Halo tickets you want to create or modify.

Required Permissions

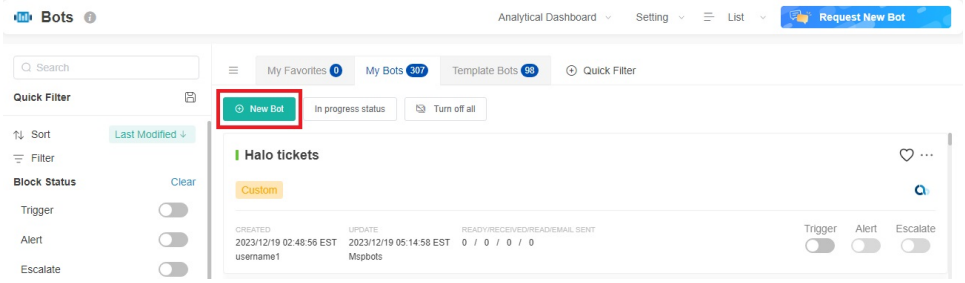
If you need to configure this guide, you need to have **admin** permissions.

How to Create a Bot to Create or Modify Halo Tickets

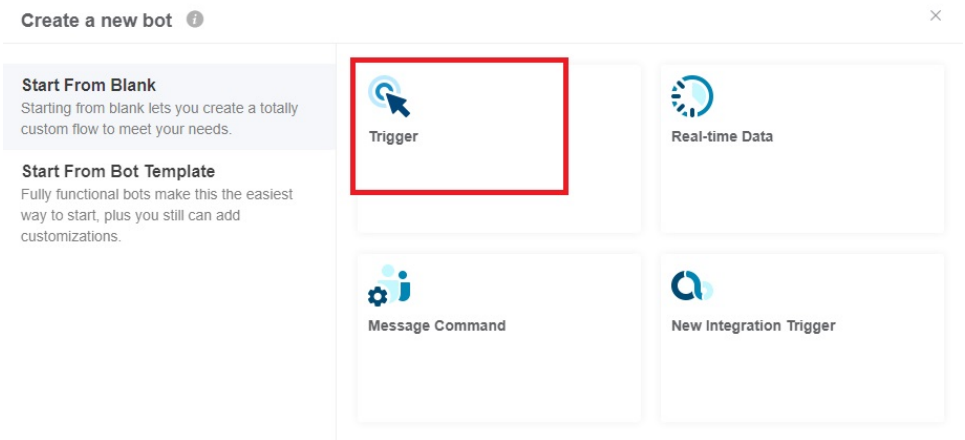
1. Log in to the **MSPbots app** using your Office 365 account and go to **Bots**.



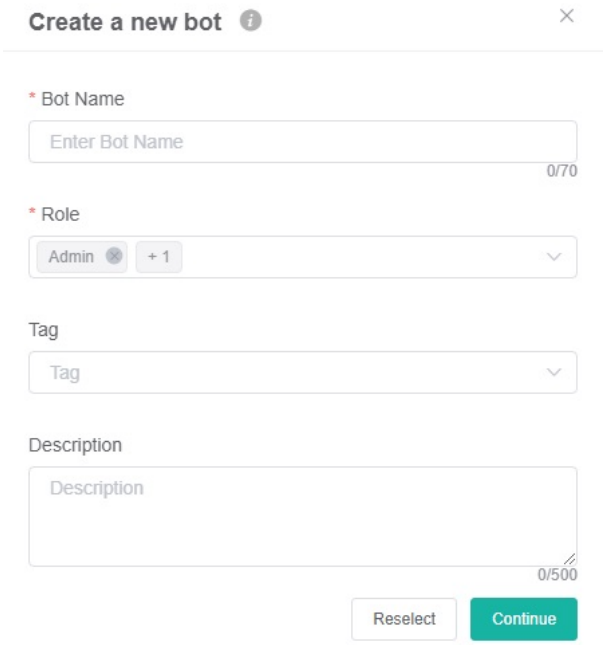
2. Click **New Bot**.



3. In the Start From Blank tab, select **Trigger**.

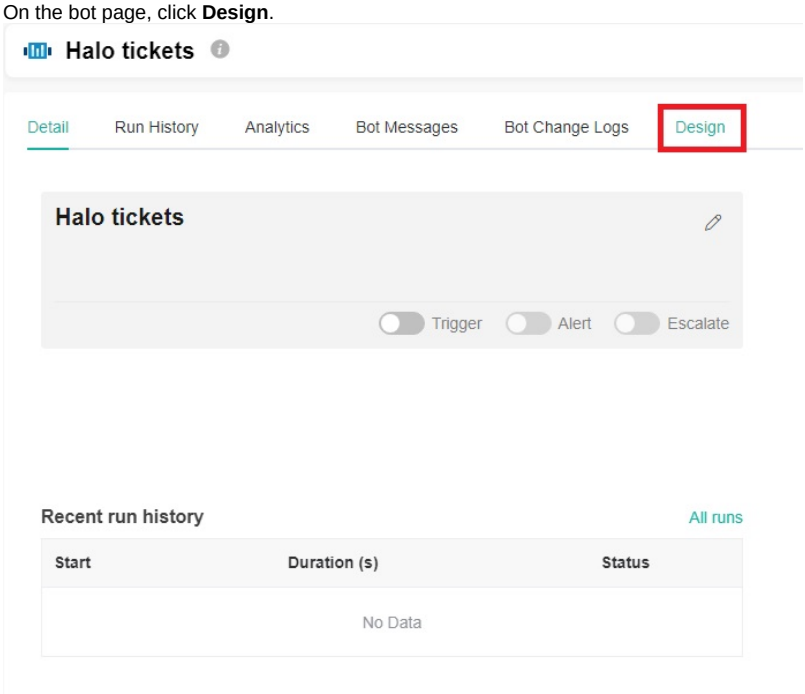


4. In the **Create a new bot** popup, fill in the following fields:
   - **Bot Name**
   - **Role**
   - **Tag**
   - **Description**



5. Click **Continue**.

6. On the bot page, click **Design**.

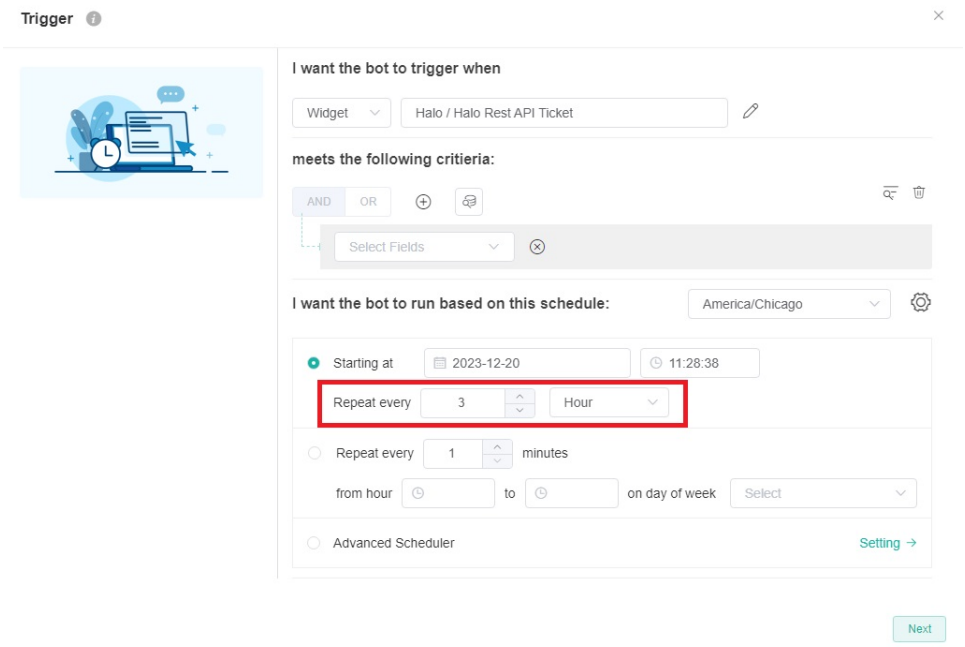


7. Configure the Trigger block, please refer to 2. Set up the bot trigger.
   a. For **I want the bot to trigger when**, add the widget you have created as the data source.
   b. For **meets the following critieria:** , if you want to further filter the data, you can configure the filtering criteria here.
   c. For **I want the bot to run based on this schedule:** , set to repeat every 3 hours.

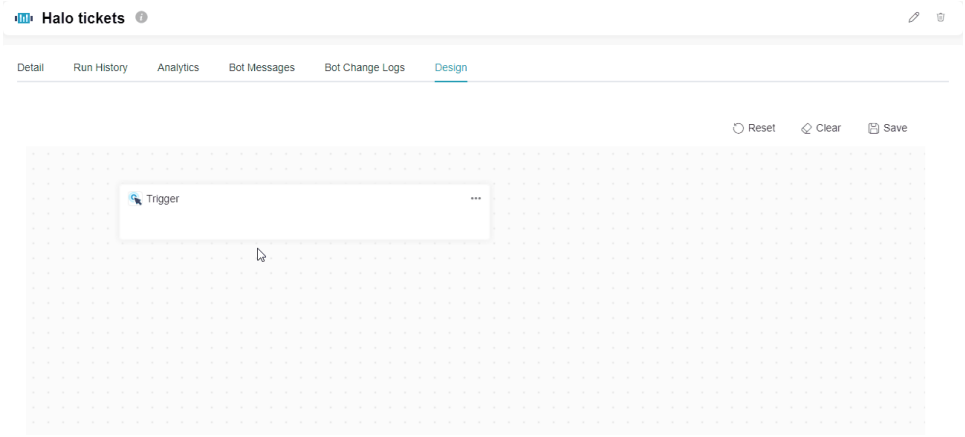
   Due to the rate limit imposed by the Halo app, the ticket synchronization frequency is as follows:

   - If you have configured the WebHook ,
     - During working hours, all tickets are synchronized through WebHook , and MSPbots do not actively synchronize them.
     - During non-working hours, the synchronization occurs once every 2 hours.
   - If you have not configured the WebHook ,
     - During working hours, the synchronization occurs every 5 minutes.
     - During non-working hours, the synchronization occurs once every 2 hours.

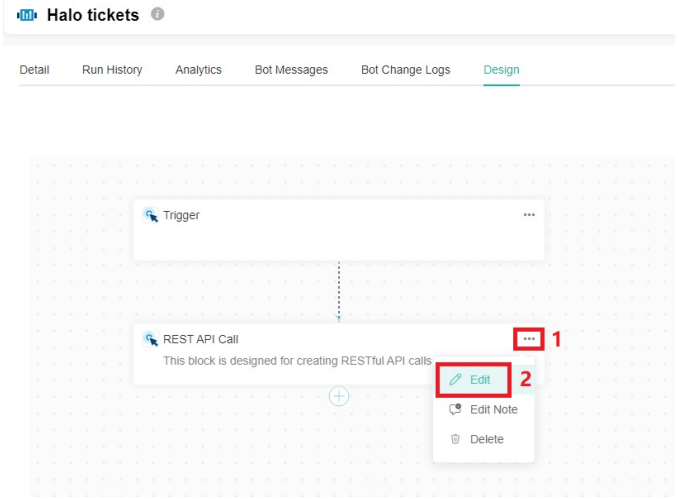
   Therefore, in order to avoid duplicate API calls, please set the triggering frequency to every 3 hours.



8. Add **REST API Call** block - Click Trigger block, click the plus button, click Add a block, click **Add a block**, search for **REST API Call** in the search bar, and click that block to add it.



9. Click the ellipsis button, select **Edit** to enter the REST API Call block page.

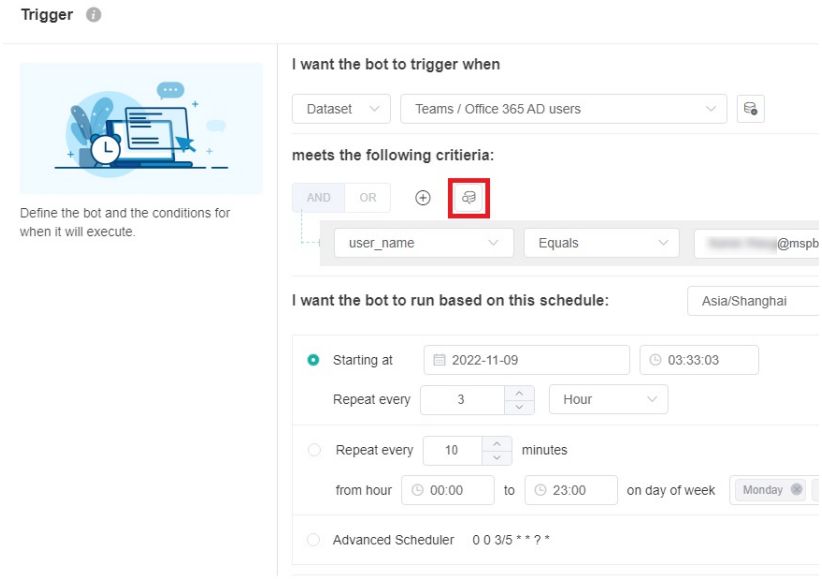


10. On the **REST API Call** block page, fill in the following fields:
    a. url - Fill in the interface address for creating or modifying tickets in Halo: **https://{host}/api/Tickets**, for example: *https://mspbotstest.halopsa.com/api/Tickets*.
    b. integration - Select **Halo** from the dropdown menu. The program has already been authorized for Halo automatically, so no additional action is required.
    c. method - Select **POST** from the dropdown menu. For specific details, please refer to the API documentation. The POST method is used for creating and updating tickets.
    d. headers - Can be left empty.
    e. params
        i. Click **body**.
        ii. Select **json**.
        iii. Input parameter fields in the JSON body, using the following columns as an example.

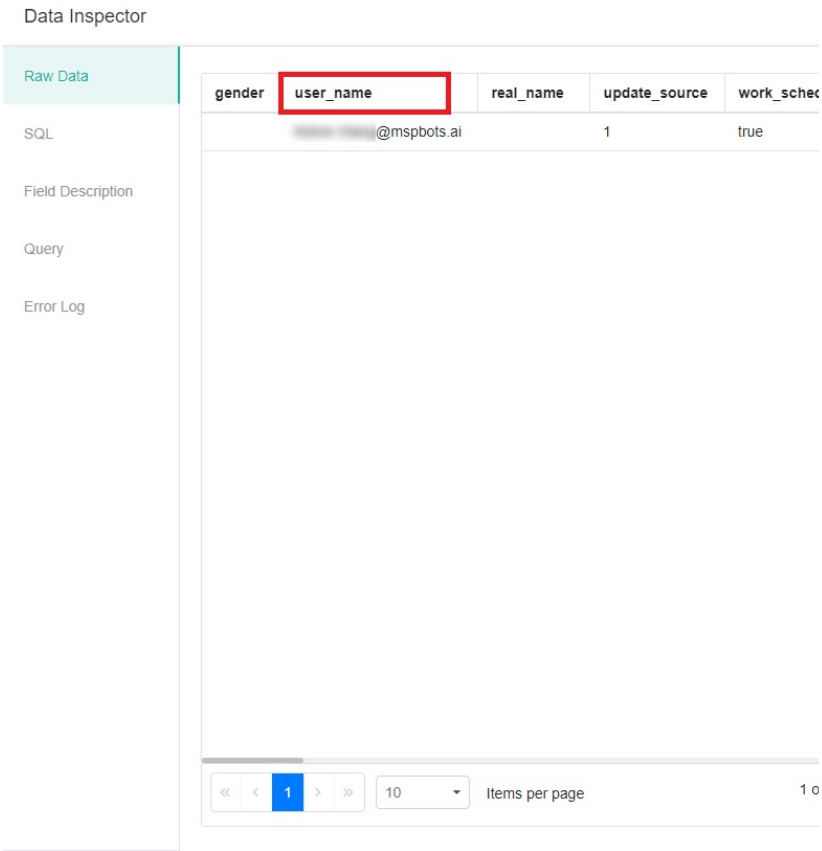
```
[
    {
        "actioncode":0,
        "dateoccurred":"2023-12-15T14:35:55.618Z",
        "summary":"{job_title}",
        "details":"{user_name}",
        "status_id":"2",
        "tickettype_id":"{update_source}"
    }
]
```

1. About the example:
    a. The "actioncode", "dateoccurred", "summary", "details", "status_id", "tickettype_id" are from https://halo.haloservicedesk.com/apidoc/resources/tickets.
    b. The {job_title}, {user_name}, {update_source} are from the data you added in the widget or dataset in the Trigger block. Please note that the format of the field must be **{Field}**.
        i. To view the field parameters:

1. Locate the **meets the following criteria** in the **Trigger** block and click  .



2. View the fields in the Row Data.

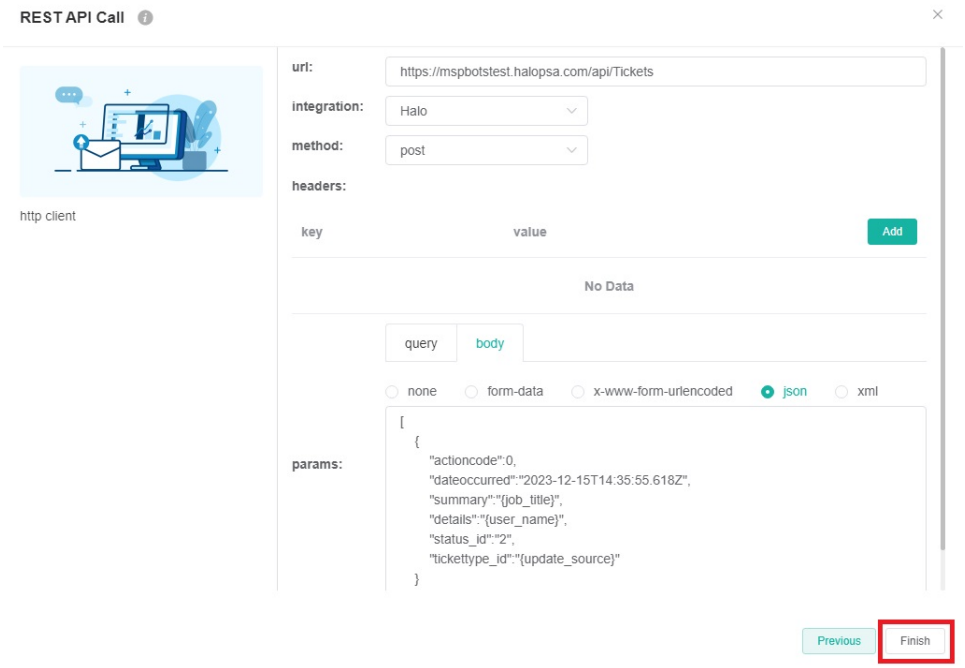


2. Every data filtered by widgets will trigger one bot execution.
3. The field values enclosed in {} in the JSON body will be dynamically replaced with the values of the queried data.
4. Please refer to the input parameter fields here: https://halo.haloservicedesk.com/apidoc/resources/tickets.
   Here are some commonly used attribute field explanations.

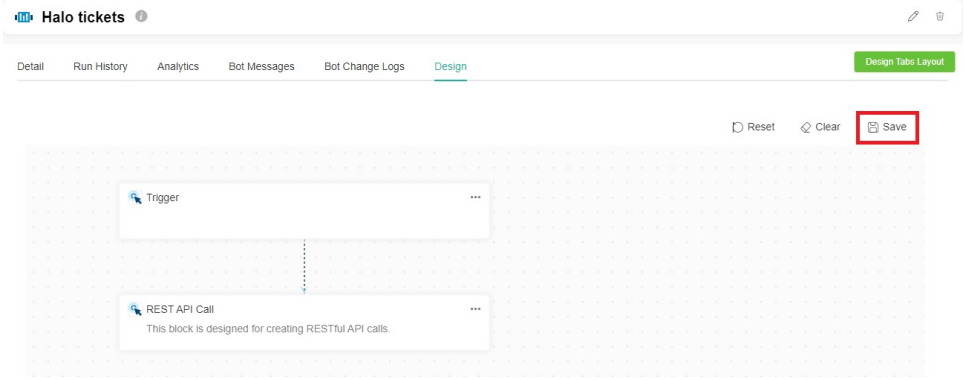
Field	Description
id	The ID of the ticket.

dateoccurred	The creation time of the ticket.
summary	A brief overview of the ticket.
details	Further information regarding the ticket.
status_id	The ID number corresponding to the status of the ticket. For example, 1 for New, 2 for In Progress. If you want to know the ID for each status, please refer to How to view the ID for ticket status or ticket type.
tickettype_id	The ID number corresponding to the ticket type. For example, 1 for Incident, 21 for Alert. If you want to know the ID for each ticket type, please refer to How to view the ID for ticket status or ticket type.

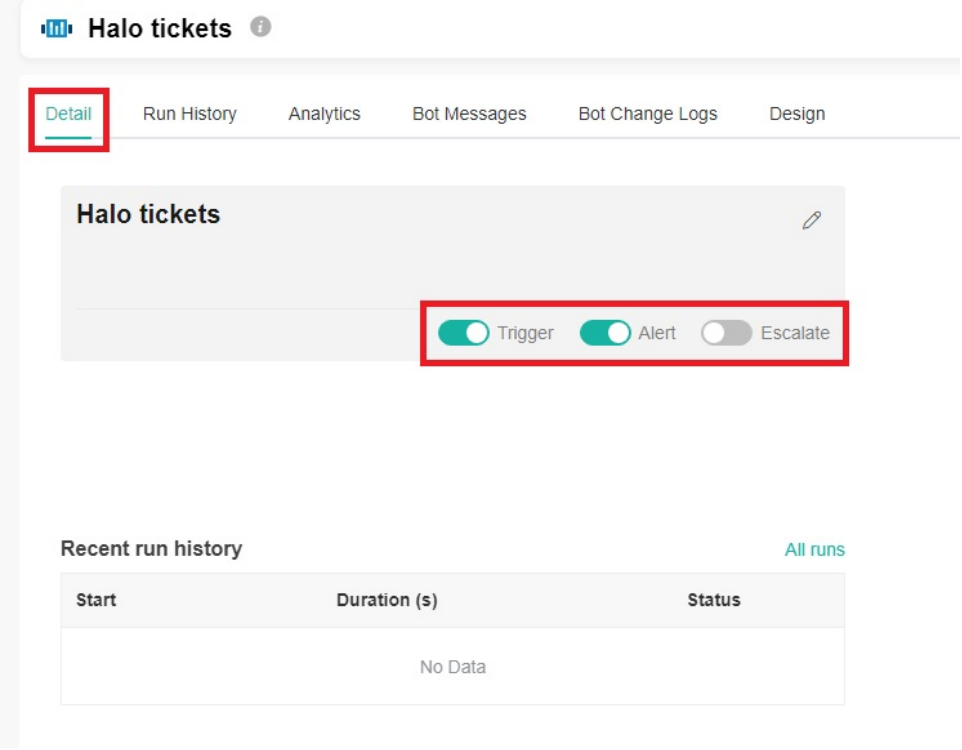
11. Click **Finish** to complete the configuration of the block.



12. After configuring, click the **Save** button to save the Bot.



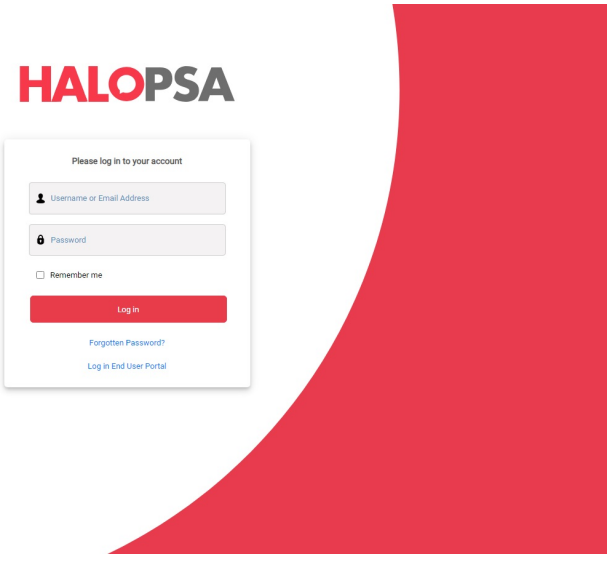
13. Click **Detail**, open the Bot's runtime switch and wait for the scheduled task to trigger. After the Bot is running, it will create new tickets or modify existing tickets based on your configuration.



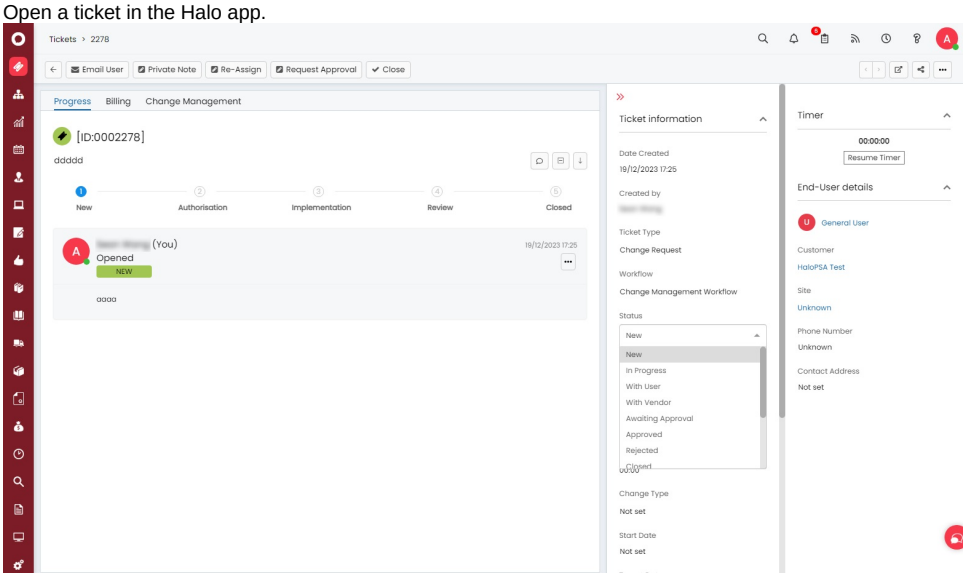
How to view the ID for ticket status or ticket type

Viewing the ID for ticket status and ticket type follows a similar process. The following guide provides an example of how to view the ID for the "In Progress" ticket status.

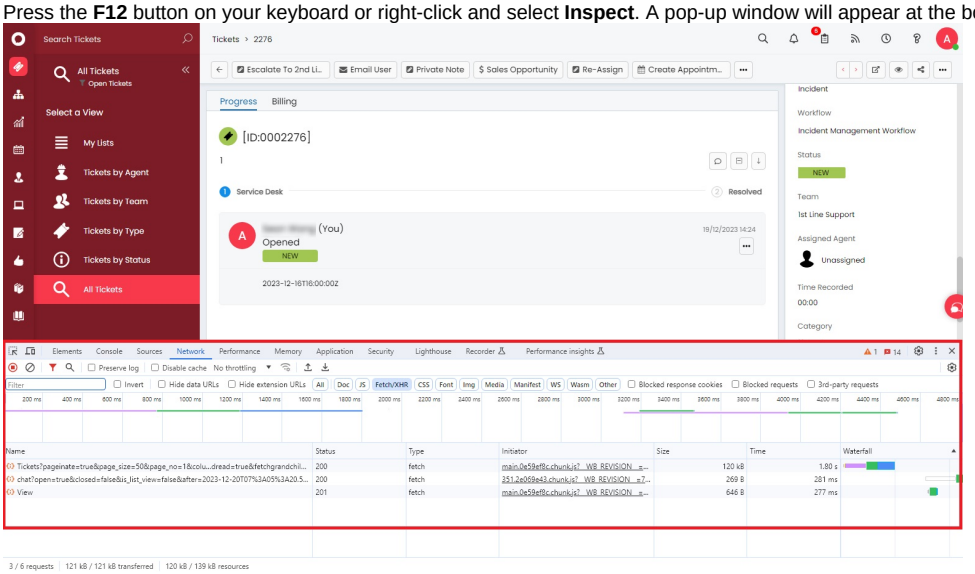
1. Log in to Halo app.



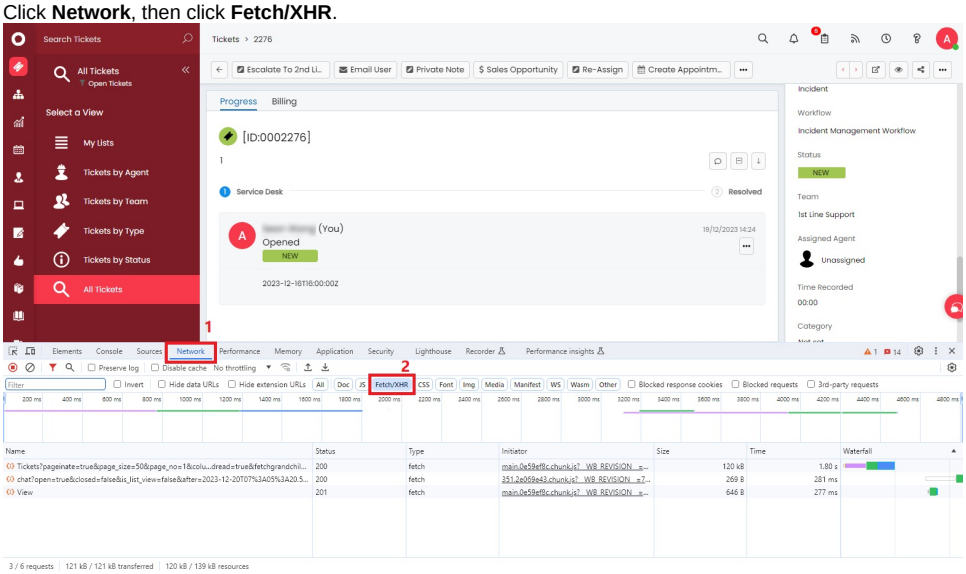
2. Open a ticket in the Halo app.



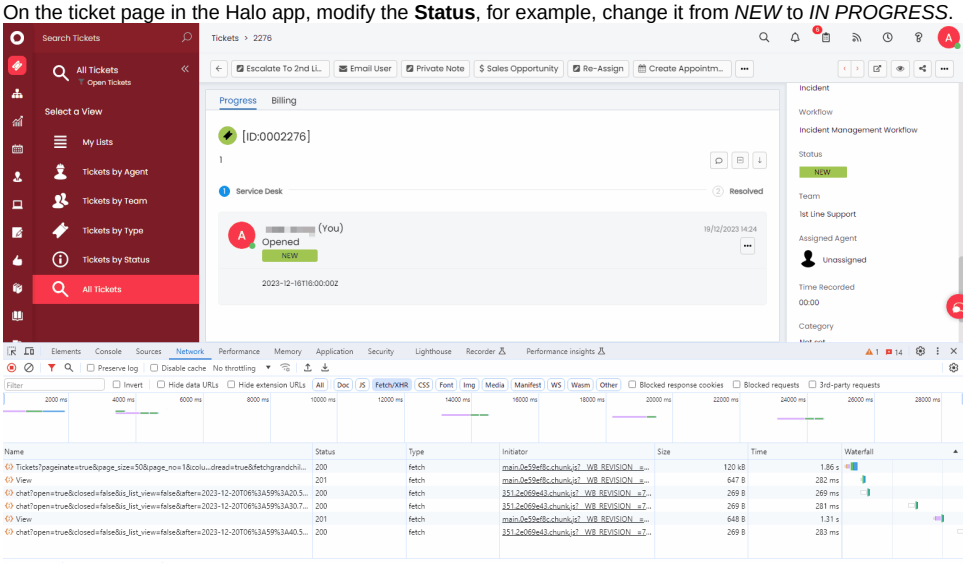
3. Press the **F12** button on your keyboard or right-click and select **Inspect**. A pop-up window will appear at the bottom of the page.



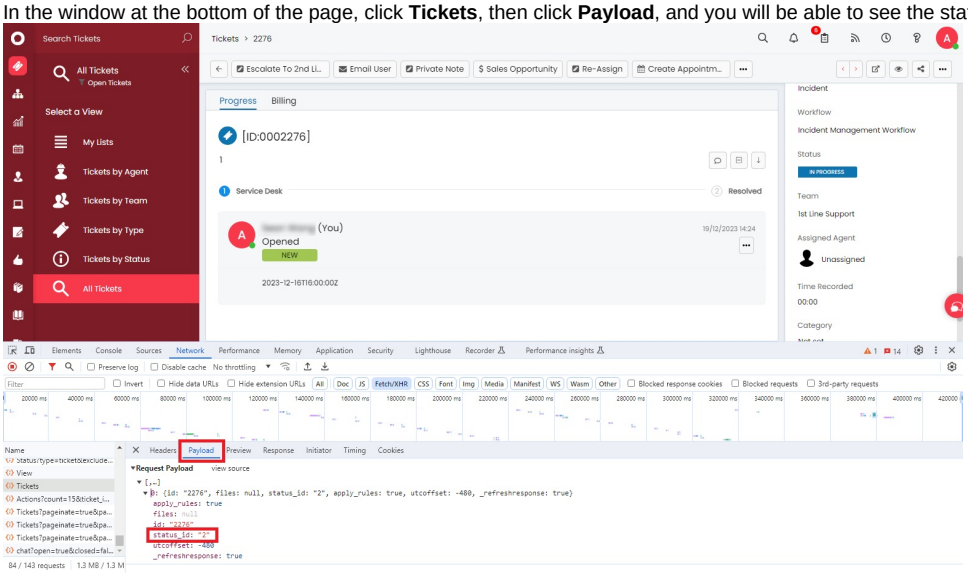
4. Click **Network**, then click **Fetch/XHR**.



5. On the ticket page in the Halo app, modify the **Status**, for example, change it from *NEW* to *IN PROGRESS*.



6. In the window at the bottom of the page, click **Tickets**, then click **Payload**, and you will be able to see the status_id as 2 with IN PROGRESS.



Related Topics

- What Filter Conditions and Formats are Available for Creating Widgets
- How to change the URL used by bots to send alerts
- Halo Integration Setup
- NextTicket Manager for Halo
- How to Create a Target Card Widget
- API Data Synchronization Failure or Pending in Halo Integration
- Creating a Scorecard Dashboard
- How to Create Slicers in Widgets
- Creating a Column Chart Widget
- Creating a Grid Widget
- Creating the Line Chart and Stacked Line Widgets
- Halo Public Datasets
- Bots: Types, Functions, and FAQs
- Bots
- MSPbots 3.0 Bots for ConnectWise Manage