Outbound Integrations

Puzzel Ticketing's Outbound Integrations or Webhooks are an excellent and powerful way of communicating with a third party web application. When an event occurs in Puzzel Ticketing, it triggers an action for a predefined message to be sent out to the Webhook URL.

For example: When the priority of a ticket changes, we can send out a generic or customised message to the registered Webhook URL.
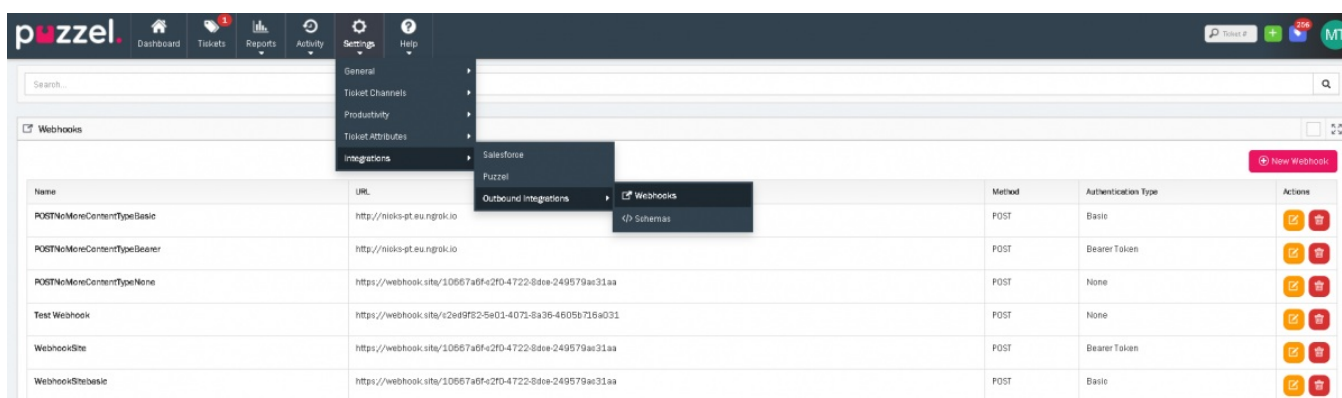
This is essentially done in three steps:

1. Register a Webhook URL

2. Define the Schemas or the message to be sent when the event occurs

3. Create a Event rule and define the event and tag the Webhook URL and message to be sent when the event occurs.

Register a Webhook URL

To register a Webhook,

1. Go to Settings -> Integrations -> Outbound Integrations -> Webhooks

2. Click on button on the right to open the New Webhook screen

3. Enter the Webhook URL, name of the Webhook, and Authentication Type. Authentication can be Basic or Bearer token type depending on the destination URL. If no authentication is required, you can choose None from the drop-down menu.

Test Webhook

URL*

https://webhook.site/c2ed9f82-5e01-4071-8a36-4605b716a031

Name*

Test Webhook

Method*

POST

Authentication Type*

None

| None |
| Basic |
| Bearer Token |

Cancel   💾 Save

4. Click Save to register the new Webhook.

> **Note**
>
> Currently, the method is restricted to **POST** only.

You can edit/delete the existing Webhook by clicking on the edit/delete icon against the Webhook.

Define the Schemas

Schema defines the body of the Webhook. This is defined separately to allow you to send different schemas[messages] to the same Webhook[destination URL]. To create a Schema,

1. Go to Settings -> Integrations -> Outbound Integrations -> Schemas
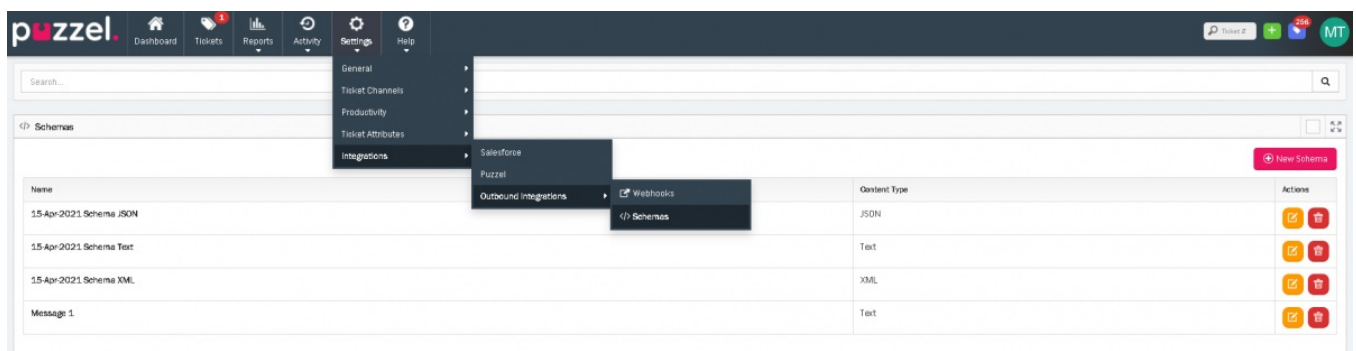
2. Click on the ⊕ New Schema icon to open the Schema Details screen

3. Enter the Schema name, Content Type and the body of the content. Content Type can be of Text, JSON, XML format.

Schema Details

Name

Message 1

Content Type

Text

Content

{
Hello
}

4. Click Save.

> **Note**
>
> You can also use placeholders in the Schema content to dynamically replace the body content with specific attributes. For example, {{ticket:ref}}. This will change the placeholder content with ticket reference number when the message is being sent. The format of the Placeholder is {{$MODEL:$ATTRIBUTE}}. To know the entire list of placeholders that can be used here refer to Placeholder article

Create Event rule

Event rule can be defined to create a trigger for the message to be sent to the destination URL if an event has occurred. To know more about the Event rules and how to create them, refer to the Event Rules article. To demonstrate the Webhook scenario, an Event Rule can be defined as shown in the picture below.

Rule Conditions

If [all ▼] of the following conditions are met:

| Ticket Event ▼ | is equal to ▼ | Priority Changed ▼ | ✕ Remove |

+ Add condition

☐ Stop checking further rules if the conditions of this rule are met.

☐ Stop checking further groups if the conditions of this rule are met.

Rule Actions

Set Response Target   OFF ⦿   **Webhooks**

Set Resolve Target   OFF ⦿ | Test Webhook | Message 1 | 🗑 |

+ Add Webhook

Cancel   💾 Save

Add a condition which says, if priority of a Ticket has changed, then trigger**Test Webhook** and send **Message 1** to the URL mentioned in the Test Webhook. For demonstration purposes, we have included test Webhook site to show that Message 1

was successfully delivered when a ticket has changed its priority.

> **Note**
>
> 1) Webhooks will only be triggered if the same combination of Webhook, Schema and ticket combination hasn't been invoked in the last minute.
> 2) If no confirmation of delivery is received from the destination URL, it will retry 13 times with an extended delay between each try.
> 3) There is no guarantee that the Webhooks will be delivered in the same order that it was triggered as it may not be successful first time round.