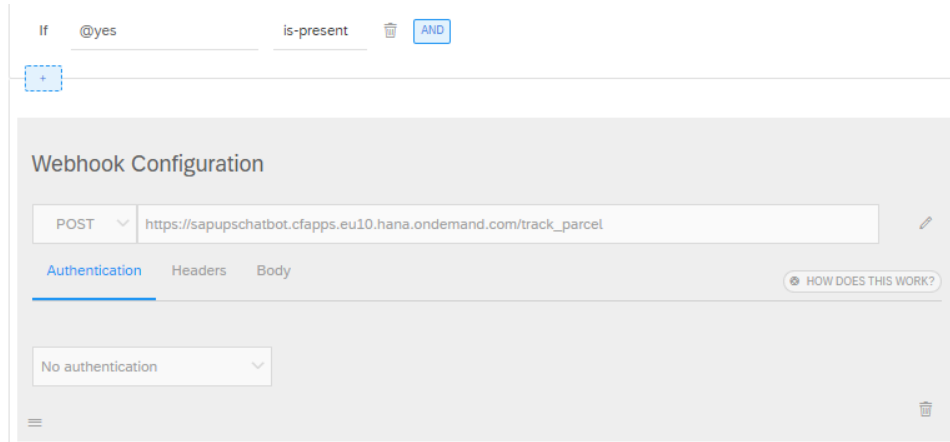


Exercise 3 - Consume APIs and call webhooks from your chatbot

SAP Conversational AI allows for two types of external connections – webhook and API service. The distinction is that webhooks expect the response to be specifically formatted for SAP Conversational AI while API service connections can map arbitrary API responses to the conversation components without a middleware service.

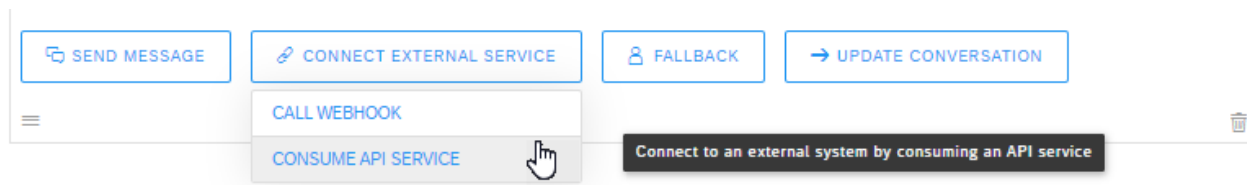
This means that connecting via the API Service Configuration will potentially save multiple network hops per message and provide a faster user experience!

Prerequisite: track-parcel skill configured with a webhook configuration:



Step 1: Convert to API Service Configuration

Delete the webhook configuration and click **Connect External Service -> Consume API Service**



Select **POST** and enter the url <https://wwwcie.ups.com/rest/Track>

Instead of connecting to the Cloud Foundry App that handles the mapping for us, we are now connecting directly to the UPS API.

For the request body, define the following:

```
{
  "UPSSecurity": {
    "UsernameToken": {
      "Username": "ysu_deliverybot",
      "Password": "190819&Jiqiren4TEST" },
    "ServiceAccessToken": {
      "AccessLicenseNumber": "0D6985C32BF33012" }
  },
  "TrackRequest": {
    "Request": {
      "RequestOption": "1",
```

```

    "TransactionReference": {
    "CustomerContext": "Your Test Case Summary Description" } },
    "InquiryNumber": {{ memory.parcel-number.raw }}
  }}

```

In the response namespace enter '**data**':

The screenshot shows the 'Response' tab of an API client. On the left, under 'Namespace', there is a text input field containing 'data'. Below it is a checkbox labeled 'Include headers' which is unchecked. On the right, under 'Response Data Schema', there is a text area containing a JSON snippet:

```
"api_service_response": {
  "data": {
    "body": { ... },
    "status_code": 200
  }
}
```

Click **Save**.

Step 2: Map the API response to conversation UI components

Because the API can return a response with either 1 or multiple records, we need to account for that in our response mapping.

First, we will need to define the memory variables we are going to use. Click **Update Conversation -> Edit Memory**:

The screenshot shows the 'Set Memory Field' dialog. On the left, there are two input fields: 'activity-status-description' and 'activity-date'. Below them is a '+ Add fields' button. In the center, there is a checkbox labeled 'Use scripting to build my JSON' which is checked. To the right of this checkbox is a text area containing a JSON script snippet:

```
1 [{"#if api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0]]{{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0].Status.Description}}{else}}{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.Status.Description}}{if}}"]
```

 On the far right, there is a 'JSON value format example' section with a dropdown menu showing 'A String:' and 'Paris'.

activity-status-description

```

"{{#if
api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0]]{{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0].Status.Description}}{else}}{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.Status.Description}}{if}}
"
```

activity-date

```

"{{#if
api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0]]{{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.[0].Status.Description}}{else}}{api_service_response.data.body.TrackResponse.Shipment.Package.Activity.Status.Description}}{if}}
"
```

Then we can send the message back to the user using the standardized values. Click **Send Message -> Text** and enter:

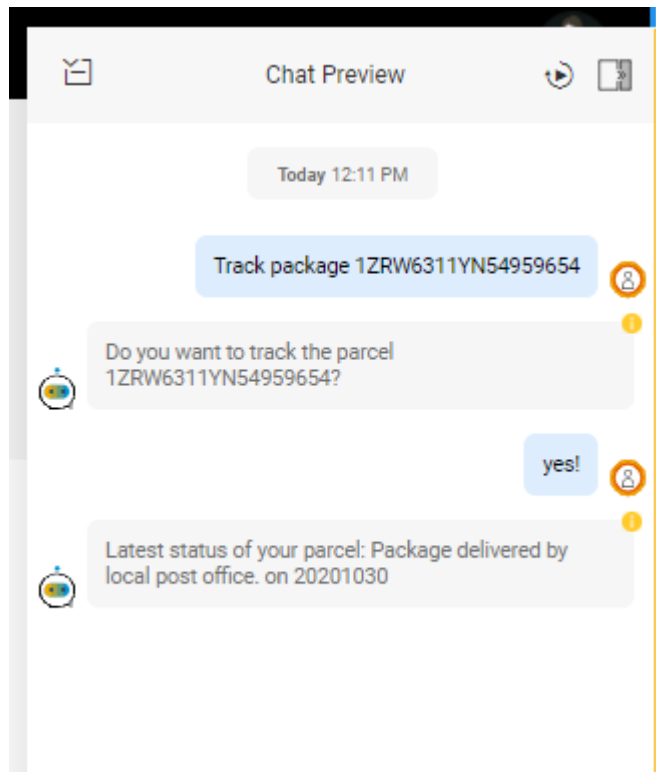
```
{{#if api_service_response.data.body.TrackResponse }}
```

Latest status of your parcel: {{ memory.activity-status-description }} on {{ memory.activity-date }}

```
{{else}}Sorry I could not find any information with the offered number. I met the error: {{  
api_service_response.data.body.Fault.detail.Errors.ErrorDetail.PrimaryErrorCode.Description }}
```

```
{{/if}}
```

Now we are ready to test our bot:



Now your bot is not relying on middleware code to connect to an external system!