Knowledge Transfer Virtual Workshops

Improve your business operations by enabling a SAP Conversational AI chatbot

**Build a chatbot from scratch consuming an external service with SAP Conversational AI**

Mariajose Martínez
Platform & Technologies Solution Engineer

October 7, 2020
Build a chatbot from scratch consuming an external service with SAP Conversational AI

Prerequisites:
- Have an SAP Conversational AI account or trial account.

Let’s start now 😊
- Create a new bot.

- Select **Perform Actions** and **Greetings** as the predefined skill for your bot.
- Give the bot a name, in my case I’ll put **bestrunbot**.
• Add a **description** and **topics** if you want to.
• Select **English** as the language.
• Select **Non-personal** in the **Data Policy**.
• Store the conversation data and,
• Select **Non-vulnerable**.
• Make your bot **Public** or **Private** as you wish.

Then click on **Create a bot**.

Now make sure you’re in the **Intents** tab inside the **Train** tab. Let’s **create an Intent** for Categories available with Products for sale.

• **Give the intent a name**, I’m naming it: **categoriesforsale** and create it.

• **Add the following expression**: See Categories for Sale.
Now let’s go to the Entities tab. You can go back to the previous tab or click again or click in the Train tab.

- Create a Restricted entity and give the name of category.
• Enter your entity just created and add the following categories as values:
  o Food
  o Beverages
  o Electronics

Let’s create another entity for products, but in this case, we are going to fetch the data from an external API service.

• Go back to Entities and click on create a new one.
• Put product as the name and select the restricted option.
• Click on Fetch via service API.

• Select GET and paste the following endpoint:
  https://services.odata.org/V3/(S(vnym1b3ehndm0p4fr0bdtbon))/OData/OData.svc/Products/?$format=json (this is a public OData service, if you want to explore it, feel free 😊)
• Click on Fetch
- And in the second step change the script with the following:
  `{{{json (pluck api_service_response.body.value 'Name')}}}

- Click now in **transform** to see the values we are going to import.
- In the import option, select **replace**.
- Then, click on **import**.
You should be able to see the values imported like this:

Now let’s build the skills of the bot.

- Click on the **Build** tab and **Add a skill**
• Give the skill a name, I’m naming it **forsale**.
• Select a **Business** skill and create it.

You’ll see your skill created like this:
Now click on it and go to the **Triggers** tab.

- Add the **categoriesforsale** intent as a trigger for this skill.

![Image of Triggers tab]

Now let’s go to the **Requirements** tab and,

- Add **category** as entity and give it the same name to save it in the memory.
- Click on **if #category is missing** to add a custom message fetching data from an external service.

![Image of Requirements tab]
Select **Connect External Service**.

And select **Consume API Service**.

Select a **GET** operation.

Copy and paste this API endpoint:
https://services.odata.org/V3/(S(vnym1b3ehndm0p4fr0bdtn0))/OData/OData.svc/Categories/?$format=json


- Let's add a Script for a **Custom Message**.

```json
{
    "type": "buttons",
    "delay": "",
    "content": {
        "title": "Please select the category you are interested in!",
        "buttons": [
            {{#eachJoin api_s.service_response.default.body.value}}
                {
                    "title": "{{Name}}",
                    "value": "{{Name}}",
                    "type": "postback"
                }{{/eachJoin}}
        ]
    }
}
```
• And save it.

Now let’s add the product entity as a **new requirement** as we did with the category one.

Select **product** as the entity and put the same name on the right.
• Open it and click on if #product is missing

We are going to repeat some steps:

• Select Connect External Service.
• And select Consume API Service.
• Select a GET operation.
• Copy and paste this API endpoint:
  https://services.odata.org/V3/(S(vnym1b3ehndm0p4fr0bdtn0))/OData/OData.svc/Products/?$filter=Categories/any(Category:Category/Name eq {{memory.category.raw}})&&$format=json

• Add a text message and paste the following script:

  {{#if (length api_service_response.default.body.value)}} Here are {{pluralize 'product' quantity=(length api_service_response.default.body.value) }}. {{/if}}

  Note: this is to customize the message if we have more than one result, so it can be shown as a plural response.

• Add a custom message, select List as the message type and paste the following json:

  {"type": "list",
   "content": {
     "elements": [
       {{#eachJoin api_service_response.default.body.value}}
         {"title": "{{Name}}",
          "imageUrl": "",
          "subtitle": "$ {{Price}}",
          "buttons": [
            {"title": "{{Name}}",
             "value": "{{Name}}",
             "type": "postback"
          ]
        }
     {{/eachJoin}}
   ]
  }
}
Now let’s add a new skill for the checkout process.

- Click on the Build tab
• Add a Business skill and name it checkout.

Move the skill by clicking in the cross and click it to train the skill:

• Go to the Requirements tab.
• Select the entity Person and put the same name to save it in the memory, we’re going to ask for the user’s full name.
• Click on if #person is missing.
Click on send message.
Select Text as message type and,
Copy and paste the following:
Could you please share with me your full name?
Save it.

Now let’s do the same to ask the user his/her email and location (address). Important fact: as you could see we didn’t create an entity for person, email and location. This is because they are golden entities already predefined in SAP Conversational AI, and there are tons more.

Click on the plus “+” to add new requirements
After adding email and location as requirements for the checkout you should see them like this:

Now let’s put the questions for each of them.

- Click on if #email is missing and paste the following as a text message: And your email as well?
- Save it
• Click on if #location is missing and paste the following also as text message:
  Now I need your shipping address, could you please write it down?
• Save it.

Now let’s go back to the **forsale** skill

• Click on the **Build** tab
• Click on the **forsale** skill
• Go to the **Action** tab inside the skill and,
• Click on **Add new message group**
• Now click on **send message**
• Select **Buttons** as your message type and,
• Copy and paste the following:
  You’ve selected this product {{memory.product.raw}} in your cart. To proceed with your purchase, click here to checkout.
• Click to **add a button** and select **Trigger Skill**

In the **button title** put: Go to checkout
• And select **checkout** as the skill to trigger
Now let’s complete for now the checkout skill. Let’s go to the Build tab and click on the checkout skill.

- Go to the Actions tab inside the checkout skill
- Click on add a new message group
- Select text as message type
- And paste the following:

  Alright {{memory.person.raw}}, you've selected the product: {{memory.product.raw}}. We'll send you an email to {{memory.email.raw}} with the order confirmation which will be delivered to your shipping address: {{memory.location.formatted}}. Thank you for using BestRunMarket Bot! :)


Note: We are going to leave the exercise ‘til this point. However, you can use this as an example on how to build a commercial bot and continue integrating it with a backend server app or external service such as a payment motor to charge the order to the customer.

Now let’s add the intent for asking categories for sale in the greetings’ skill. Let’s go to the Build tab and click on the greetings’ skill.

Go to the Actions tab. Notice that this skill and these responses are already configured because at the beginning we selected greetings as a predefined skill for the bot.

- **Delete** all the responses, we’re going to put a new one.
• Click on send message and,
• Select Card as message type.

• As image url, put the following url with the BestRunMarket logo: https://i.imgur.com/reQirgK.png
• In the title put: Hello! Welcome to the BestRunMarket Bot!
• Subtitle: How may I help you?
• Click to add a button and select Trigger Skill
In the **button title** put: See Categories for Sale
And select the **forsale** skill.

Now let’s add another button:

- Click to add another button and select **Postback**

In the **button title** and **Postback value** put the same: Ask for an Order Tracking (we’re going to leave it ‘til here, using the intent see categories for sale as the main intent for this exercise... however this is an example of how we can put more options for the user to select and interact with the bot).

- **Save** it.
Now let’s test the bot.

- Click on Chat Preview.

And test the bot as following:
To see the bot’s memory, click on any yellow i in the conversation flow. For example, by clicking in the last one, you’ll be able to see the whole entities saved in the bot’s memory, like this:

```
{
  "conversation": {
    "id": "test-1601849579140",
    "language": "en",
    "memory": {
      "category": {
        "confidence": 0.99,
        "raw": "beverages",
        "value": "beverages"
      },
      "product": {
        "confidence": 0.99,
        "raw": "cranberry juice",
        "value": "cranberry juice"
      },
      "person": {
        "confidence": 0.99,
        "raw": "maria martinez",
        "fullname": "maria martinez"
      },
      "email": {
        "confidence": 0.99,
        "raw": "mariajose.martinez@sap.com",
        "domain": "@sap.com",
        "tag": null,
        "local": "mariajose.martinez"
      },
      "location": {
        "confidence": 0.55,
        "raw": "paseo de la reforma 410, juarez, 06680, ciudad de mexico",
        "country": "mx",
        "lng": -99.17873919999999,
        "city": "juarez",
        "formatted": "av. paseo de la reforma 410, juarez, cundinamarca, 06680 juarez, cdmx, mexico",
        "street_number": "410",
        "place": "Chihuajalpolo_XURt259wYX9gts",
        "state": "CDMX",
        "type": "street_address",
        "postal_code": "06680",
        "lat": 19.42498813,
        "street_name": "av. paseo de la reforma"
      },
      "skill": "checkout",
      "skill_occurrences": 4,
      "skill_id": "5d656738-cb92-4810-bb8b -e06ed12351f7"
    },
    "logs": {
      "input": "paseo de la reforma 410, juarez, 06680, ciudad de mexico",
      "intent": ""
    }
  }
}
```

Hope you’d enjoyed this hands-on exercise! 😊