Date :  26-Mar-24

Small Talk Data Adapter

version: 2.0.0

# **Overview**

Small Talk is a data adapter that helps you rapidly build mobile apps with conversational interface. A mobile app integrated with the Small Talk data adapter is intelligent enough to understand user queries and answer as a real person. The Small Talk data adapter makes use of the capabilities of Dialogflow Small Talk Agent.

Dialogflow Small Talk is a pre-built agent developed with a set of basic questions and answers. You can customize the agent's responses to fit into your use case.

## **Use case:**

## Consider a case that you want to develop a chatbot for your e-commerce mobile app. The chatbot should be able to greet users, ask users to provide their details, such as name, phone number, interested products, and so on, and try to answer basic questions asked by the users. You can use the Small Talk data adapter to fulfill your requirement. The Small Talk data adapter is easy to integrate into your mobile app and is intelligent to converse small talks with your users.

## **Percentage of re-use:**

70% (Data can be customizable and customer need to implement UI by themselves).

# **Getting Started**

## **Prerequisites**

Before you start using the Small Talk Bot data adapter, ensure you have the following:

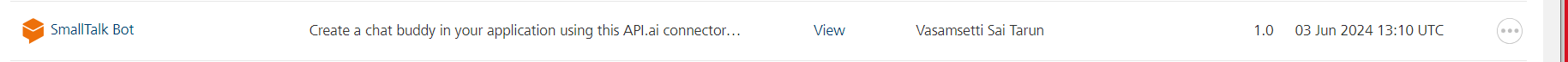
* [HCL Foundry](https://manage.hclvoltmx.com/)

## **Importing the Adapter**

**To import the Data Adapter to Volt Foundry, do the following:**

1. Sign in to the  [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Custom Data Adapters**.  
   
4. Click **IMPORT** to import a custom data adapter.  
   
5. On the Import Data Adapter dialog box, click browser to import.  
   
6. Selected DAILOGFLOW data adapter zip file and click **IMPORT**.

After importing the data adapter, Volt Foundry opens a window that shows the metadata of the data adapter.



After you import the data adapter, you can view it on the Custom Data Adapters page and use it to create services on Volt Foundry.

After you add the data adapter to your Volt Foundry app. You can refer to below section for more information.

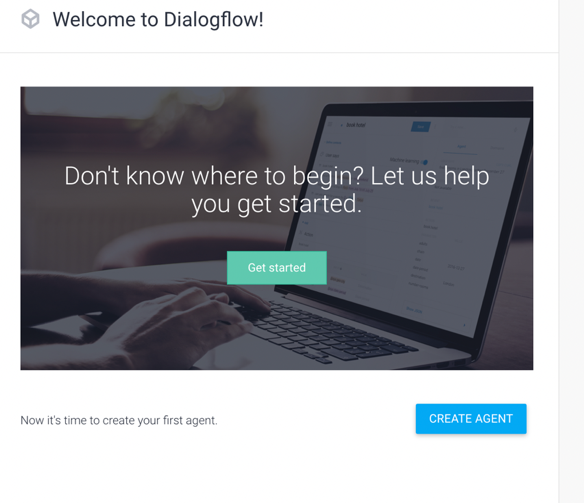
**Follow the given steps to train your chatbot in Dialog Flow**

1. Go to [DialogFlow portal](https://dialogflow.cloud.google.com/)

A logo on a white background

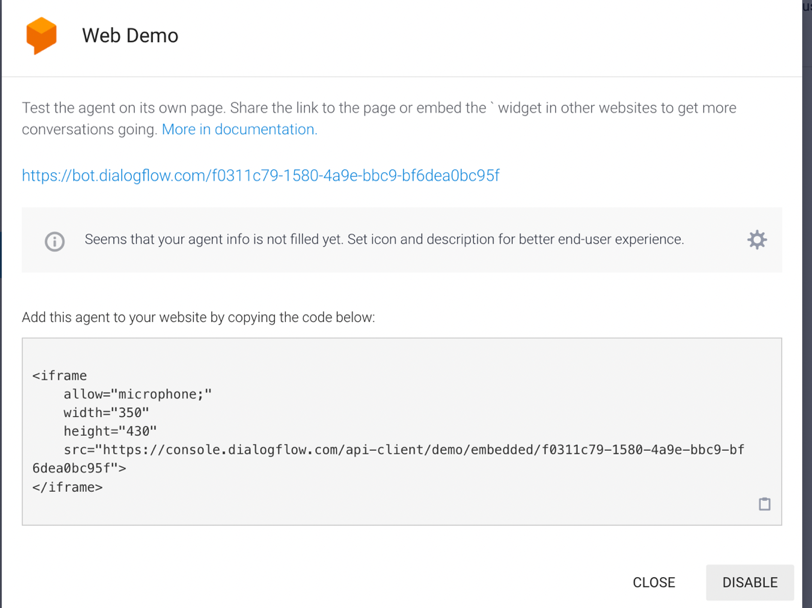
Description automatically generated

2. Click on **create agent**



3. After creating agent, Train your bot with intents, entities according to your requirement for your chatbot

4. In the Integrations, click on web demo



5. click on the link appear on the above screenshot and check whether your bot trained

6. After that click on gear icon (beside your agent name)

i. Go to General, click on projectID NAME, you will be redirecting to google cloud.

A screenshot of a chat

Description automatically generated

ii. There we need to go to API &Services and create an API key there after one .json file will be downloaded after creation of API key.for reference checkout -> [CREATIONOFAPIKEY](https://www.youtube.com/watch?v=rWcLDax-VmM&pp=ygUiY3JlYXRlIGFuIGFwaSBrZXkgc2VydmljZSBhY2NvdW50cw%3D%3D)

iii. Go to **Export and Import** first export the .zip file and again import the same .zip file

A screenshot of a computer

Description automatically generated

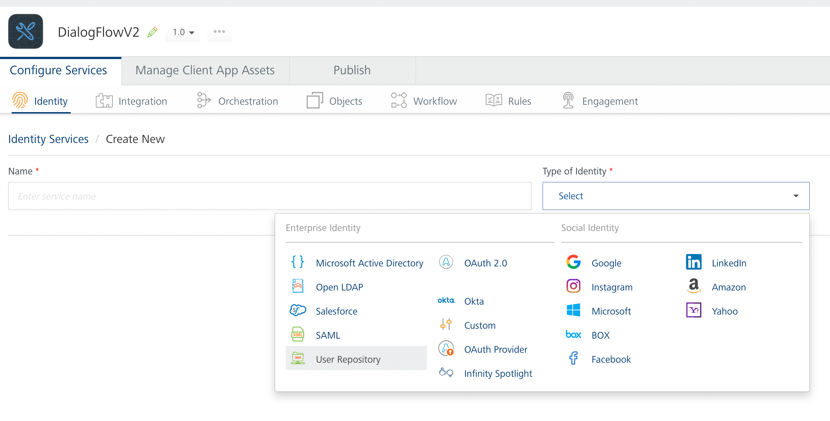
iv. Go to Environments and click on **PUBLISH A VERSION**

A screenshot of a computer

Description automatically generated

# **Creating an Identity service**

After you import the data adapter into Volt Foundry, you can use it to create an Identity Service. Select **User Repository** as type of Identityand click on add user.



# **[Creating an Integration service](javascript:void(0);)**

After Creating Identity service Volt Foundry, you can use it to create an Integration Service.

Follow the given steps to create an Integration service using the DAILOGFLOW Data Adapter.

1. Sign in to the [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Integration**.



1. To create a new service, click the **+** button or the **CONFIGURE NEW** button.



1. On the Service Definition tab, select the service type as DAILOGFLOW, and click **SAVE**.

A screenshot of a computer

Description automatically generated

**[Open](javascript:void(0);)****5.** [**Creating and Executing operations**](javascript:void(0);)

After you create an integration service, you can create and execute operations using the service.

#### **Creating an operation**

1. In **API Management**, in the **Integration** section, select the service that you created.
2. After you select the service, navigate to the **Operation List** tab.  
   
3. From the drop-down list, select an operation that you want to execute, and click **ADD OPERATION**.

A screenshot of a computer

Description automatically generated

4.In Dialog flow we need two services to fetch our response.

i. First one, to generate an access token.

A screenshot of a computer

Description automatically generated

ii. In the preprocessor ->JavaScript add the .json file text which got above while creating API key.

iii. After that click on **Save And Fetch Response,**

iv. Below screenshot providing sample response output for Dialogflow authentication (i.e to generate access token).

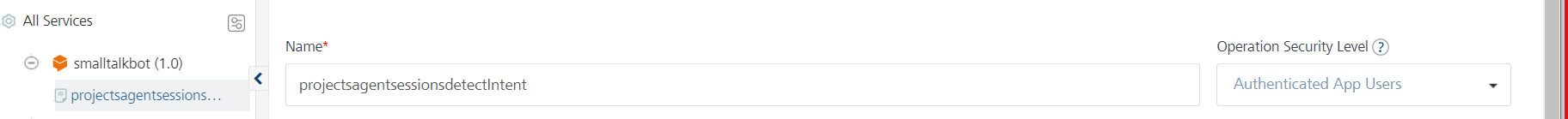
A screenshot of a computer

Description automatically generated

v. To fetch response from operations.

**2. Executing an operation**

* 1. From the **Operations List** tab, in the **Configured Operations** section, select the operation you want to execute.



* 1. On the Operation Page, in the Request Input tab, enter a TEST VALUE for all the fields.

A screenshot of a computer

Description automatically generated

Reference:

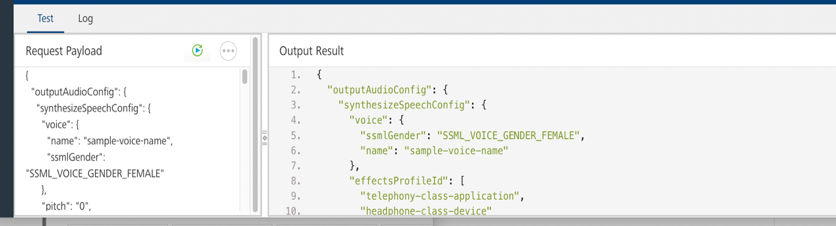
Go to [Method: projects.agent.environments.users.sessions.detectIntent  |  Dialogflow ES  |  Google Cloud](https://cloud.google.com/dialogflow/es/docs/reference/rest/v2beta1/projects.agent.environments.users.sessions/detectIntent) and click on Try it present on righthand side of the screen

* 1. In the Header section, give Authorization as access token with token\_type as **“Bearer**”

A screenshot of a computer

Description automatically generated

1. Select a run-time environment and click **Save and Fetch Response** to get a response from DAILOGFLOW based on your inputs.



**[Open](javascript:void(0);)**

# **References**

## **Dynamic Usage**

You need to configure the Request Input variables on Volt Foundry based on the operation you want to perform. You can refer to the following drop-down sections for sample requests and responses for the operations.

### **dialogflowsessionsdetectintent**

#### Description

You can use this operation to get a desired response.

#### Request Input Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Required/Optional** | **Description & Allowed Values** |
| name | Required | Provide any voice name |
| ssmlGender | Required | speech synthesis gender for natural and lifelike conversational interactions |
| audioEncoding | Required | Converting spoken input into machine-readable format for accurate voice recognition. |
| text | Required | The text you want to give to chatbot to answer your query |
| languagecode | Required | chatbot's understanding and response language |
| entityOverrideMode | Required | chooses between user input and predefined values. |
| projectsId | Required | unique identification code to distinguish individual projects or applications within the platform |
| sessionsid | Required | allowing the system to distinguish and maintain context during conversational interactions |
| analyzeQueryTextSentiment | Required | evaluates the sentiment of input text. |

#### **Sample Request**

{

"outputAudioConfig": {

"synthesizeSpeechConfig": {

"voice": {

"name": "sample",

"ssmlGender": "SSML\_VOICE\_GENDER\_UNSPECIFIED"

},

"pitch": "0.0",

"speakingRate": "0",

"volumeGainDb": "0.0",

"effectsProfileId": [

"test"

]

},

"audioEncoding": "OUTPUT\_AUDIO\_ENCODING\_UNSPECIFIED",

"sampleRateHertz": "0"

},

"queryInput": {

"text": {

"text": "Hi",

"languageCode": "en"

}

},

"queryParams": {

"contexts": [

{

"name": "projects/dialogflowrooms-wgcr/agent/sessions/c46ae231-6585-445b-9a41-77f9f9d4a54a/contexts/context33",

"lifespanCount": "10"

}

],

"sessionEntityTypes": [

{

"entities": [

{

"value": "val",

"synonyms": [

"val1"

]

}

],

"name": "projects/dialogflowrooms-wgcr/agent/sessions/35968daf-7797-4961-82c4-9f51df2c4a66/entityTypes/Locations",

"entityOverrideMode": "ENTITY\_OVERRIDE\_MODE\_OVERRIDE"

}

],

"geoLocation": {

"latitude": "0",

"longitude": "0"

},

"sentimentAnalysisRequestConfig": {

"analyzeQueryTextSentiment": "false"

},

"resetContexts": "false",

"timeZone": "America/Los\_Angeles"

},

"projectsId": "small-talk-nvdp",

"sessionsId": "35968daf-7797-4961-82c4-9f51df2c4a66",

"inputAudio": ""

}

#### **Sample Response**

#### **A screenshot of a computer Description automatically generated**

1. {
2. "outputAudioConfig": {
3. "synthesizeSpeechConfig": {
4. "voice": {
5. "name": "sample"
6. },
7. "effectsProfileId": [
8. "test"
9. ]
10. }
11. },
12. "opstatus": 0,
13. "queryResult": {
14. "allRequiredParamsPresent": "true",
15. "fulfillmentMessages": [
16. {
17. "text": {
18. "text": [
19. "Hi!"
20. ]
21. }
22. }
23. ],
24. "outputContexts": [
25. {
26. "name": "projects/small-talk-nvdp/agent/sessions/35968daf-7797-4961-82c4-9f51df2c4a66/contexts/context33",
27. "lifespanCount": "9",
28. "parameters": {
29. "Color.original": "",
30. "Color": ""
31. }
32. }
33. ],
34. "queryText": "Hi",
35. "action": "smalltalk.greetings.hello",
36. "languageCode": "en",
37. "parameters": {},
38. "intent": {
39. "displayName": "smalltalk.greetings.hello",
40. "name": "projects/small-talk-nvdp/agent/intents/75e6d4e4-d79e-4d3a-b7dc-b1d4a1cd2ead"
41. },
42. "fulfillmentText": "Hi!",
43. "intentDetectionConfidence": "1"
44. },
45. "responseId": "2dd87978-cbfc-4794-ab77-eb3a5922e54b-896d2846",
46. "httpStatusCode": 200
47. }

# [**Publishing your application**](javascript:void(0);)

After adding the **SmallTalkBot** Data Adapter to your app and configuring the necessary configurations, you must publish the app to VoltMX Foundry. For more information, refer to [Publish a Project to VoltMX Foundry.](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/PublishVoltMXFoundryServicesApp.html)

# **Revision History**

App version 2.0.0:

## **Known Issues**

None

## **Limitations**

Limited support for languages, complex context handling, manual training, restricted customization, dependency on internet connectivity, potential privacy concerns.