Date :  12-Oct-23

Weather bot Data Adapter

VERSION: 2.0.0

# **Overview:**

This connector will allow you to leverage a simple weather chatbot for retrieving weather information. You can ask the bot for weather updates for your location, and even get forecasts. Watch out for a rainy day!

Weather Bot is a data adapter that helps you quickly integrate intelligent conversational user interface into your mobile apps to provide accurate weather information.

## **Use case:**

## You can use the Dialogflow Data Adapter to integrate with various types of applications like Customer support chatbot, Smalltalk bot, Weather bot etc.

Consider a case that you want to develop a mobile app that provides weather information to bikers. Using the mobile app, the bikers should be able to get the accurate weather information and look for the weather information hourly and location basis.

You can develop a mobile app and make use of the Weather Bot data adapter to get the weather details. Weather Bot is a ready to use data adapter and can be easily integrated with the mobile apps. You can avoid writing lengthy code to obtain weather information from various web services, which consumes lot of time and effort. With the usage of Weather Bot, you just need to do the following:

1. Design and develop the user interface and functionality of the mobile app using Iris
2. Import the Weather Bot data adapter into foundry and create integration service with it.
3. Create operations with required APIs to fetch the weather details.
4. Integrate the mobile app with the integration service and invoke the operations to execute when required.

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

75% (Data can be customizable and customer need to implement UI by themselves).

## **Features:**

* Easy to create applications to provide weather information
* Builds a connection with [Dialogflow](https://dialogflow.com/) (formerly api.ai) Weather Agent
* Predefined APIs to fetch weather details
* Easy to make API calls and view responses

# **Getting Started**

## **Prerequisites**

Before you start using the Weather bot data adapter, ensure you have the following:

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

## **Platforms Supported**

### Mobile

#### iOS

#### Android

## **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**.



1. Click **IMPORT** to import a custom data adapter.

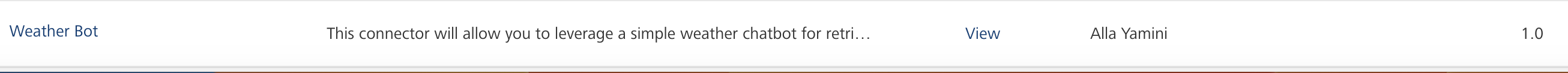


1. On the Import Data Adapter dialog box, click browser to import.



1. Selected Weather bot 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.

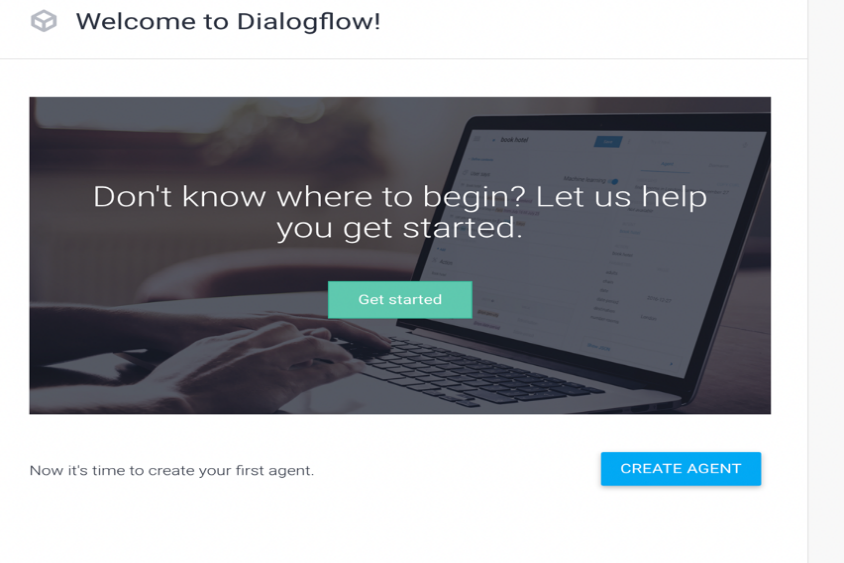
**Follow the given steps to train your chatbot in DialogFlow:**

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

A logo on a white background

Description automatically generated

* 1. Click on **create agent.**



1. After creating agent, Train your bot with intents, entities according to your requirement for your chatbot.
2. In the Integrations, click on web demo.

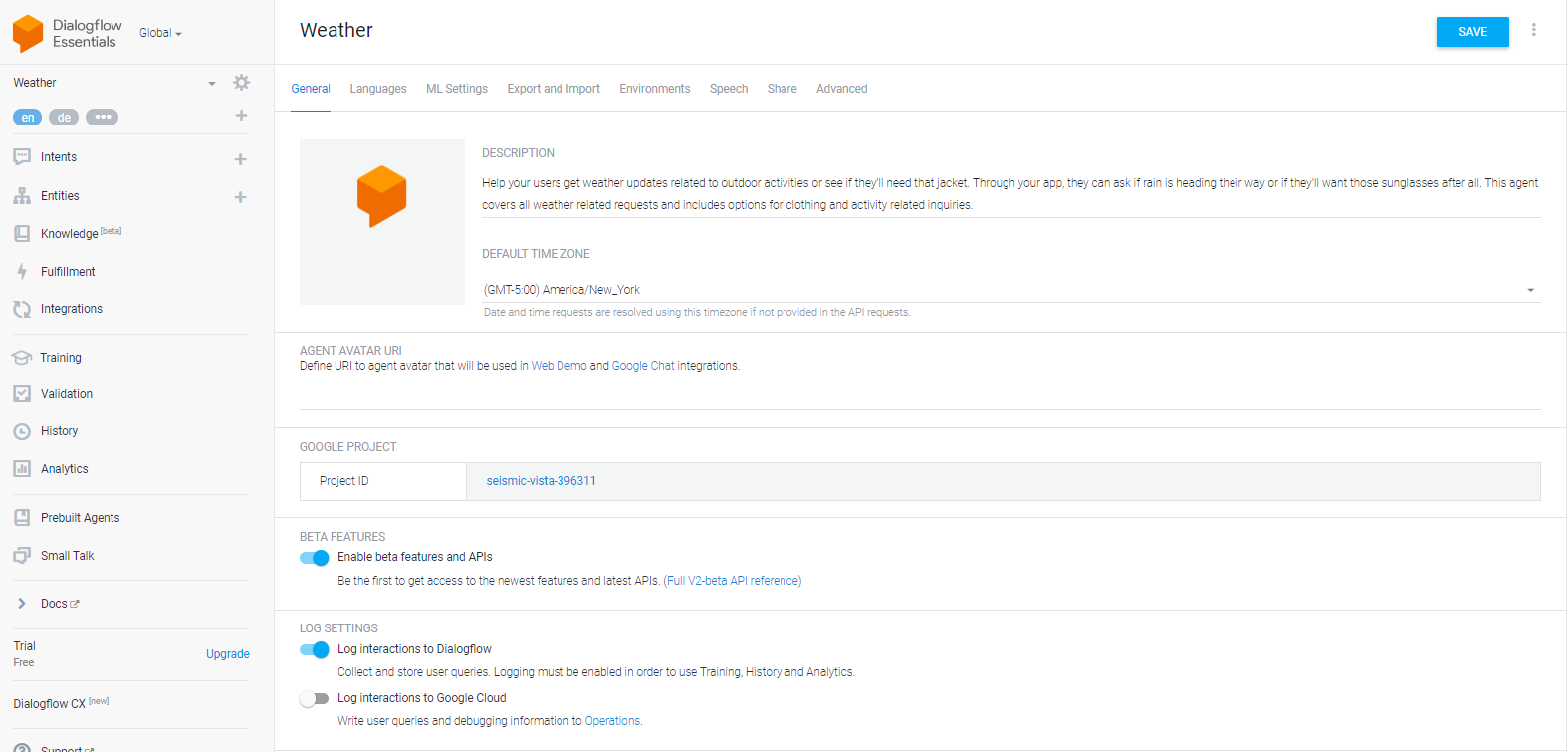
A screenshot of a computer

Description automatically generated

5. Click on the link appear on the above screenshot and check whether your bot is trained correctly as per your requirement

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

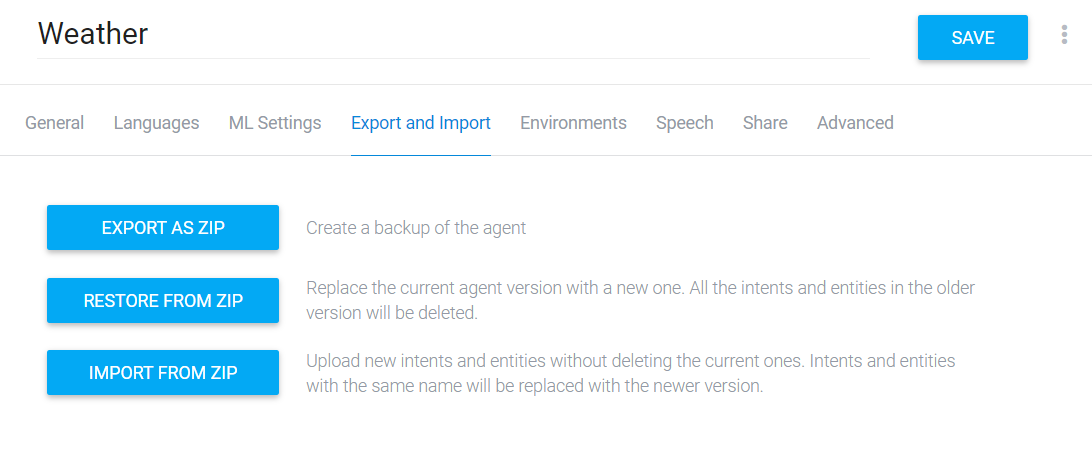
i. Go to General, click on projectID NAME, you will be redirecting to google cloud.



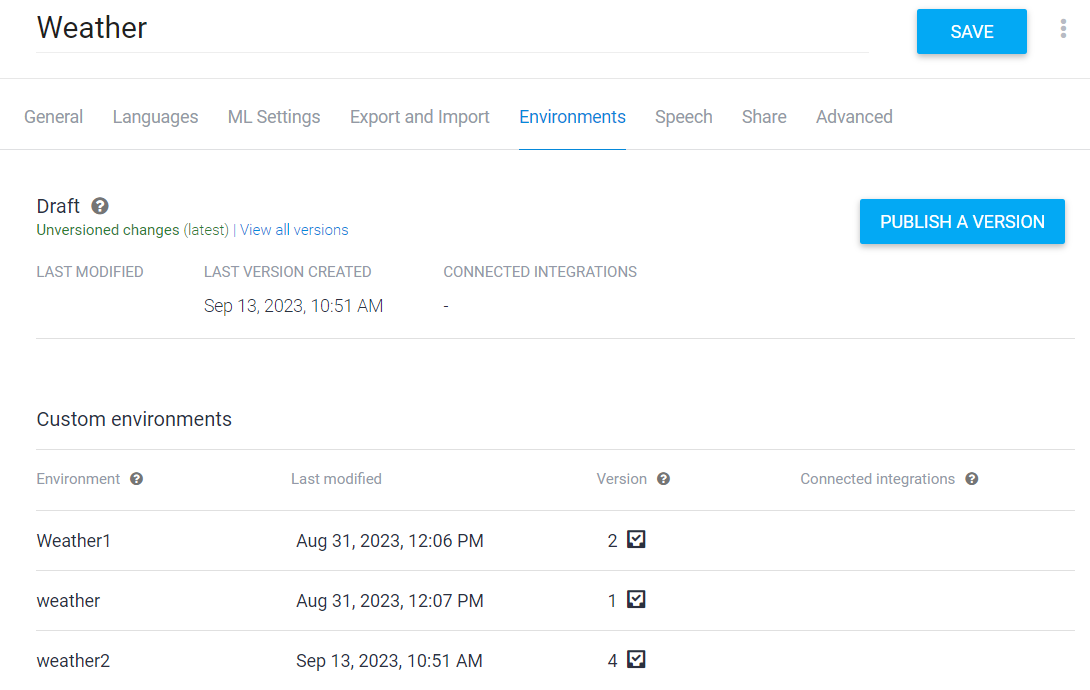
ii. There we need to go to API & Services and create an API key there after one .json file will be downloaded after creation of API key.

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.



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



**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.

A screenshot of a computer

Description automatically generated

**[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 DIALOGFLOW 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**.

A screenshot of a computer

Description automatically generated

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 **Weatherbot**, and click **SAVE**.

A screenshot of a computer

Description automatically generated

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

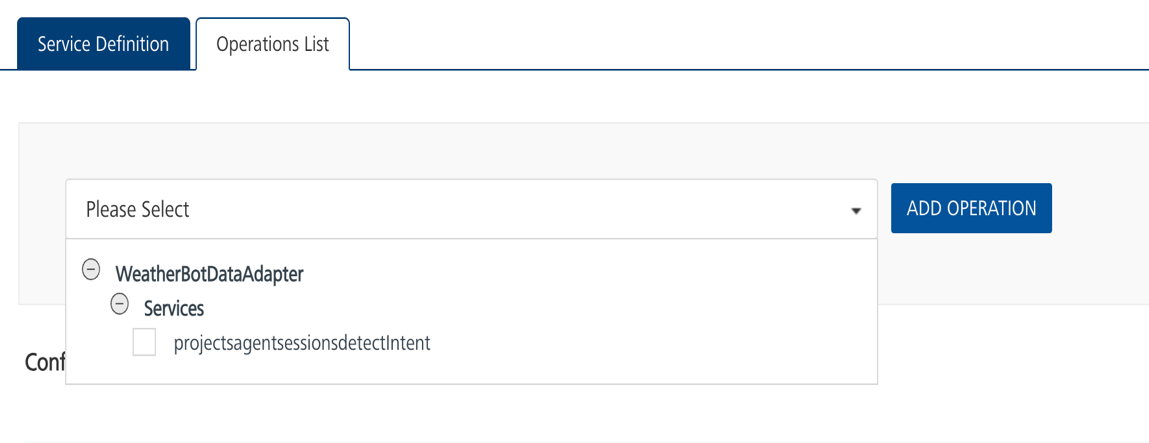
After you create an integration service, you can create and execute operations using the service.

#### **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.



* 1. From the dropdown list, select an operation that you want to execute, and click **ADD OPERATION**.



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

I)First one, we need to generate an access token.

A screenshot of a computer

Description automatically generated

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

ii. after that click on **Save And Fetch Response.**

iii. below screenshot providing sample response output for CopyDialogflowAuthentication (i.e., to generate access token).

A screenshot of a computer

Description automatically generated

II) To fetch response from operations

* **Executing an operation**

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

A screenshot of a computer

Description automatically generated

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

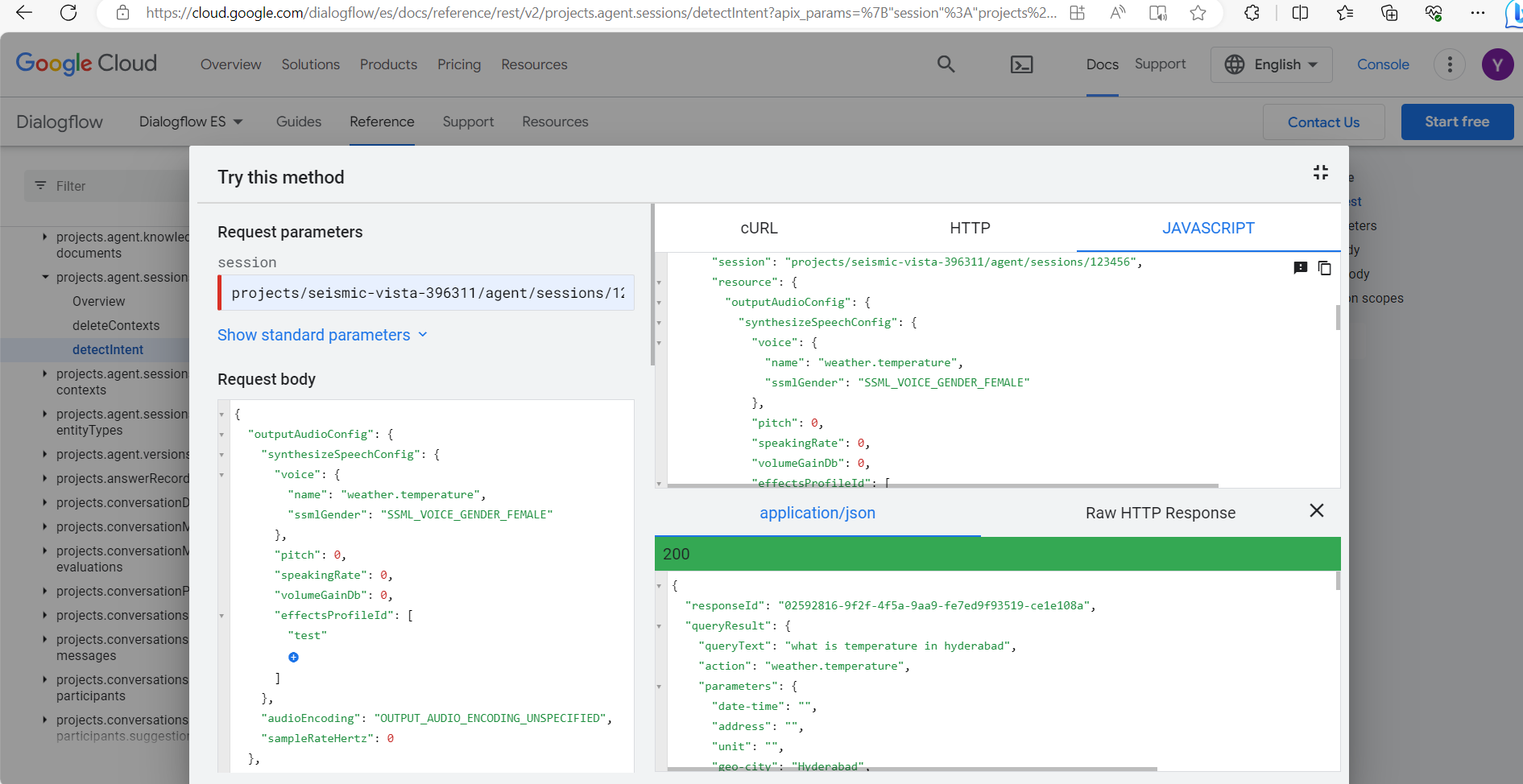
A screenshot of a computer

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

Ex:



3.In the Header section, give Authorization as access token with token type as **“Bearer**”.

A screenshot of a computer

Description automatically generated

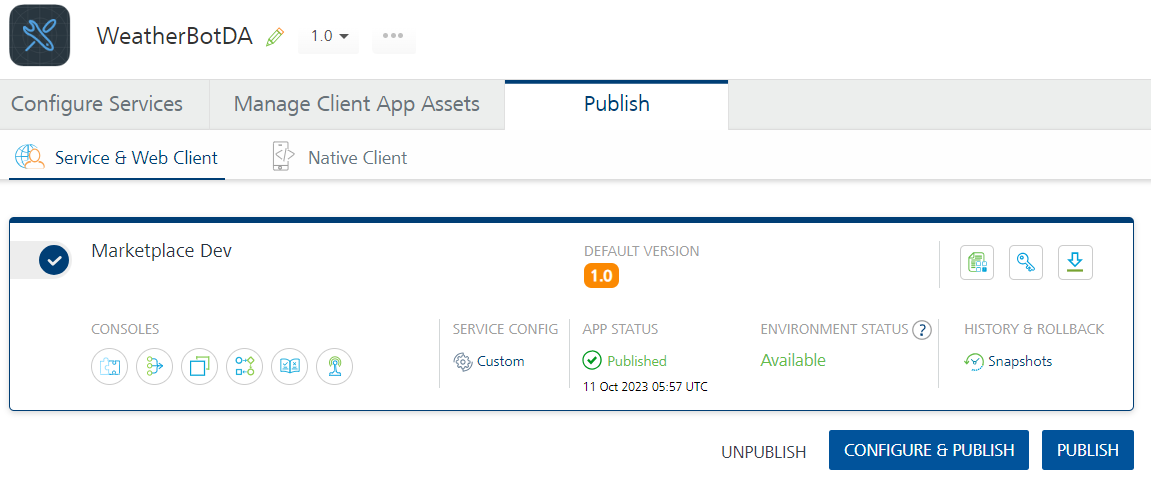
4.Select a run-time environment and click **Save and Fetch Response** to get a response from DIALOGFLOW based on your inputs.

A screenshot of a computer

Description automatically generated

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

If you want to use the services in client applications, you need to publish the app to a run-time environment.



# **References:**

## **Dynamic Usage:**

## **Volt Foundry Operations**

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.

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

#### **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": "weather.temperature",

"ssmlGender": "SSML\_VOICE\_GENDER\_FEMALE"

},

"pitch": "0",

"speakingRate": "0",

"volumeGainDb": "0",

"effectsProfileId": [

"test"

]

},

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

"sampleRateHertz": "0"

},

"queryInput": {

"text": {

"text": "what is temperature in hyderabad",

"languageCode": "en"

}

},

"queryParams": {

"contexts": [

{

"name": "projects/seismic-vista-396311/agent/sessions/123456/contexts/my\_context2",

"lifespanCount": "17"

}

],

"sessionEntityTypes": [

{

"entities": [

{

"value": "val1",

"synonyms": [

"value1"

]

}

],

"name": "projects/seismic-vista-396311/agent/sessions/123456/entityTypes/weather.temperature",

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

}

],

"geoLocation": {

"latitude": "45.67",

"longitude": "45.89"

},

"sentimentAnalysisRequestConfig": {

"analyzeQueryTextSentiment": "false"

},

"resetContexts": "false",

"timeZone": "America/Los\_Angeles"

},

"projectsId": "seismic-vista-396311",

"sessionsId": "123456",

"inputAudio": ""

}

#### **Sample Response:**

1. {
2. "outputAudioConfig": {
3. "synthesizeSpeechConfig": {
4. "voice": {
5. "ssmlGender": "SSML\_VOICE\_GENDER\_FEMALE",
6. "name": "weather.temperature"
7. },
8. "effectsProfileId": [
9. "test"
10. ]
11. }
12. },
13. "webhookStatus": {
14. "code": "16",
15. "message": "Webhook call failed. Error: UNAUTHENTICATED, State: URL\_ERROR, Reason: ERROR\_AUTHENTICATION, HTTP status code: 401."
16. },
17. "opstatus": 0,
18. "queryResult": {
19. "allRequiredParamsPresent": "true",
20. "fulfillmentMessages": [
21. {
22. "text": {
23. "text": [
24. "Temperature in Hyderabad is 34C"
25. ]
26. }
27. }
28. ],
29. "outputContexts": [
30. {
31. "name": "projects/seismic-vista-396311/agent/sessions/123456/contexts/weather",
32. "lifespanCount": "2",
33. "parameters": {
34. "date-time.original": "",
35. "number": "",
36. "unit.original": "",
37. "unit": "",
38. "address": "",
39. "geo-city": "Hyderabad",
40. "date-time": "",
41. "temperature": "temperature",
42. "geo-city.original": "hyderabad",
43. "number.original": "",
44. "temperature.original": "temperature",
45. "address.original": ""
46. }
47. },
48. {
49. "name": "projects/seismic-vista-396311/agent/sessions/123456/contexts/my\_context2",
50. "lifespanCount": "16",
51. "parameters": {
52. "date-time.original": "",
53. "unit.original": "",
54. "number": "",
55. "unit": "",
56. "address": "",
57. "date-time": "",
58. "geo-city": "Hyderabad",
59. "temperature": "temperature",
60. "geo-city.original": "hyderabad",
61. "number.original": "",
62. "temperature.original": "temperature",
63. "address.original": ""
64. }
65. }
66. ],
67. "queryText": "what is temperature in hyderabad",
68. "action": "weather.temperature",
69. "languageCode": "en",
70. "parameters": {
71. "number": "",
72. "unit": "",
73. "address": "",
74. "geo-city": "Hyderabad",
75. "date-time": "",
76. "temperature": "temperature"
77. },
78. "intent": {
79. "displayName": "weather.temperature",
80. "name": "projects/seismic-vista-396311/agent/intents/cfeb6976-460a-4d30-b43a-f6938e449f4f"
81. },
82. "diagnosticInfo": {
83. "webhook\_latency\_ms": "71"
84. },
85. "fulfillmentText": "Temperature in Hyderabad is 34C",
86. "intentDetectionConfidence": "1"
87. },
88. "responseId": "9b571421-26f3-4794-a8b1-ea7b9dbef83f-d5f6109d",
89. "httpStatusCode": 200
90. }

## **Properties**

-- None of the properties are exposed.

## **Events**

-- None of the events are exposed.

## **APIs**

-- None of the APIs are exposed.

# **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.