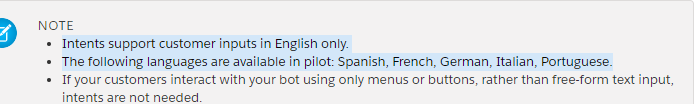
**Introduction**

As of writing this article, intents in Salesforce Einstein Bot support the following languages(for more details, click [here](https://help.salesforce.com/articleView?id=bots_service_train_bot.htm&type=5)):



Google dialogflow supports over 25 language-locales for text. You can find the list [here](https://cloud.google.com/dialogflow/docs/reference/language).

In this document we have integrated Einstein bot with Google DialogFlow such that Salesforce passes text that is entered by the user to Dialogflow. Dialogflow identifies the intent from the text and returns an action and the parameters back to Salesforce.

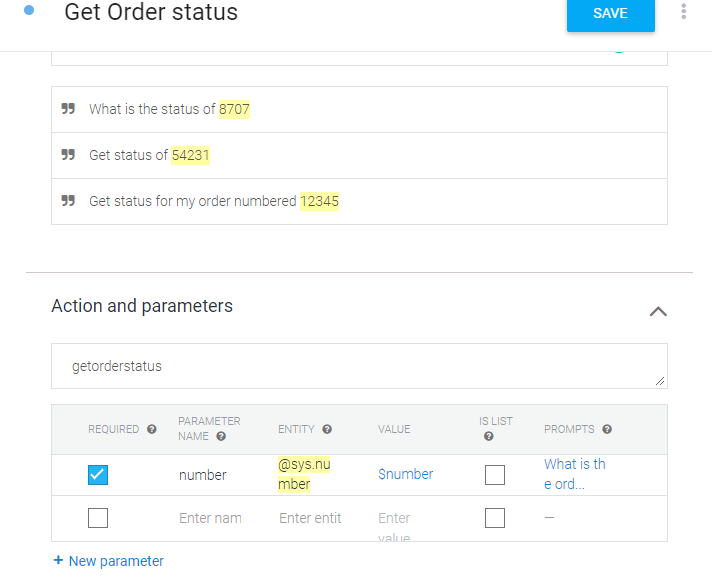
**Pre-requisites**

Create your Einstein bot on salesforce:  
<https://trailhead.salesforce.com/en/content/learn/projects/build-an-einstein-bot>

Create a DialogFlow account and agent on google:  
<https://cloud.google.com/dialogflow/docs/quick/build-agent>

**Creating a dialogflow agent**

Create an intent called ‘Get Order Status’ with your training phrases. Examples of a few phrases are below.



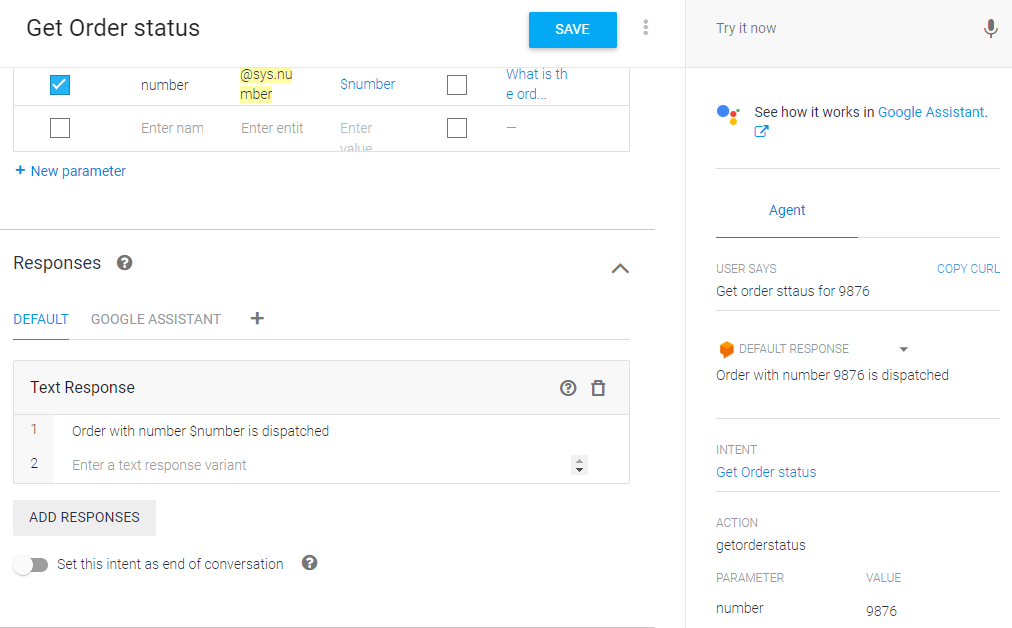
We select the “order number” in the phrase and mark it as a sys.number entity to mark it as a parameter.

We mark the parameter ‘number’ as required and define a prompt if the order number is not provided by the user

We’ve named the action ‘getorderstatus’. We will use this name to identify what action must be taken in apex

We create a response using ‘$number’ to show the order number that was entered by the user

We can test the entire flow by using the ‘Try it now’ section on the right



**Connect salesforce to dialogflow**

You can create an auth provider for google with Open ID by following the steps in the following link:  
<https://medium.com/@ennoucas/working-with-salesforce-named-credentials-example-with-google-calendar-api-61fa6588167a>  
  
To test the connection you can use the following code to check if you get the appropriate response:

HttpRequest req = new HttpRequest();

req.setEndpoint('callout:DialogFlow/v2/projects/project-name /agent/sessions/1236:detectIntent?key=your\_api\_key');

req.setMethod('POST');

req.setBody('{"queryInput":{"text":{"languageCode":"en","text":"Get order status for 7"}}}');

req.setHeader('Content-Type','application/json');

Http http = new Http();

HTTPResponse res = http.send(req);

Here, ‘callout:DialogFlow’ is your named credential that you create using the Auth Provider, also use your project name and your API key.

More details about API key:  
<https://support.google.com/googleapi/answer/6158862?hl=en&ref_topic=7013279>

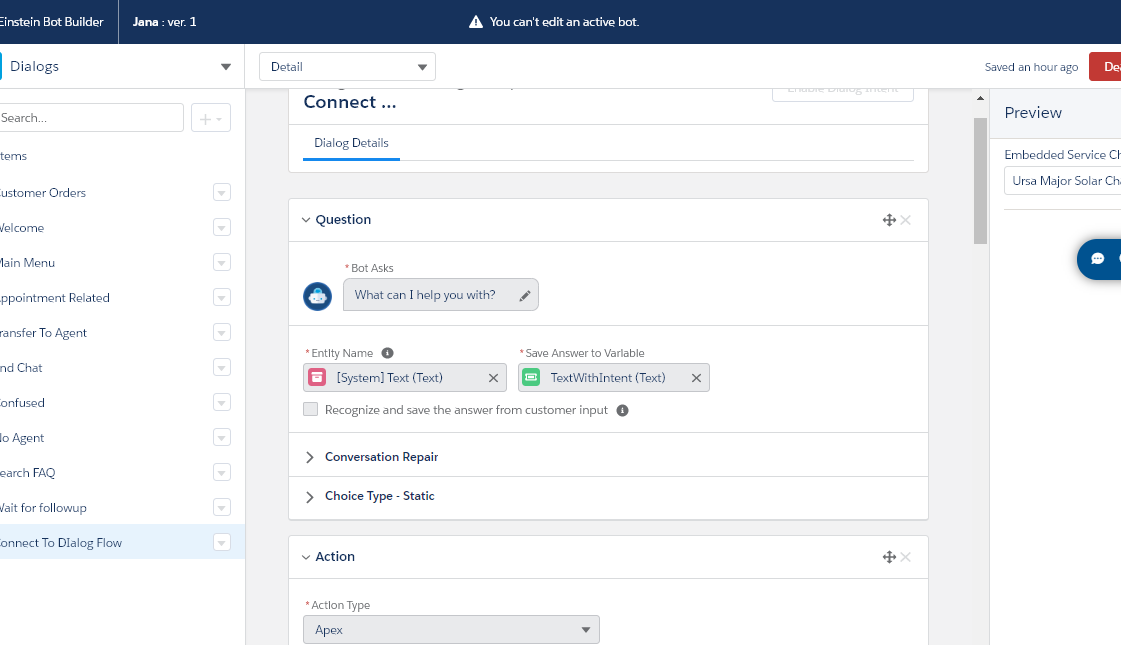
More details about this REST API:  
<https://cloud.google.com/dialogflow/docs/reference/rest/v2/projects.agent.sessions/detectIntent>

**Configure the Salesforce bot**

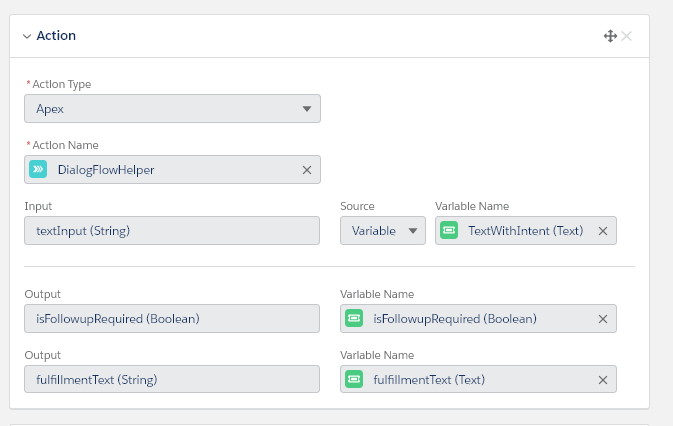
We create a dialog, ‘Connect To Dialogflow’.

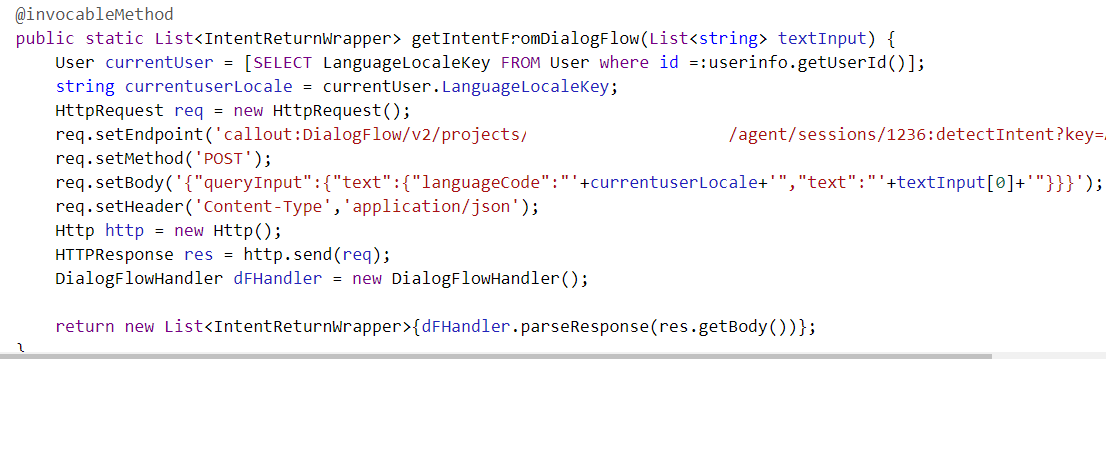
Add a question that the bot asks, for example, “What can I help you with?”

The answer to that question is saved in a text variable that we name ‘TextWithIntent’. This is the text that the user enters and will be passed to dialogflow

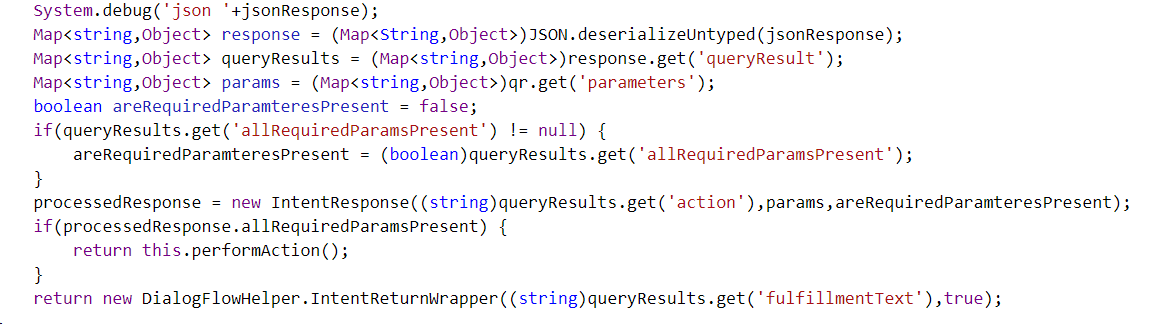


In the action, we call an apex function that passes the text to dialog flow and gives us an action and fulfillment text



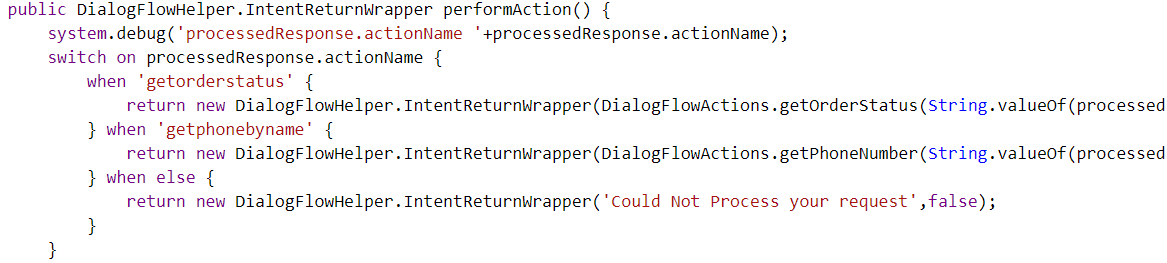


We check the response to get the action, fulfillment text and if all required parameters are present.

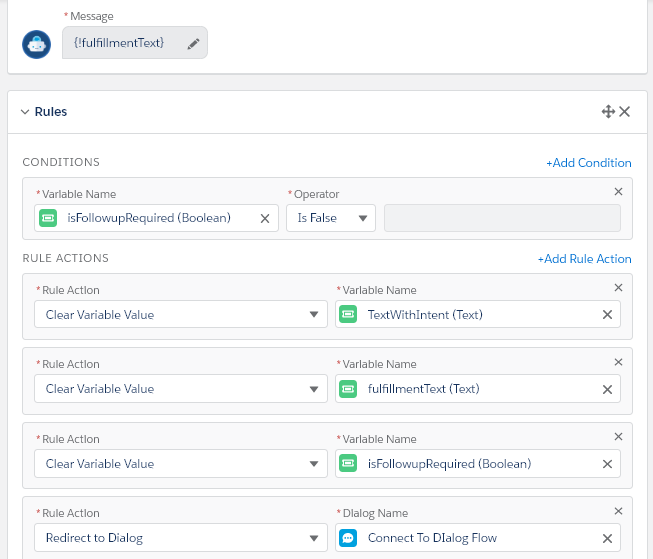


If all parameters are present, we can perform the action, else we display the fulfillment text that prompts the user to enter the required data. This is the text that we entered as ‘Prompt’ when we marked our parameter as ‘Required’ in Dialogflow intent

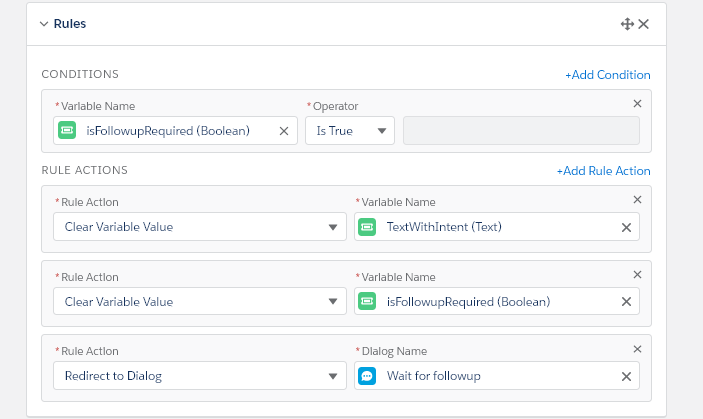
Based on the action name, we can call our apex functions



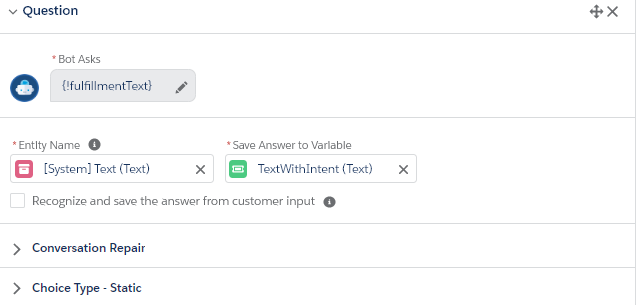
If no follow up question is required to complete the action, we display the result, clear old values from the variables and call the same dialog again



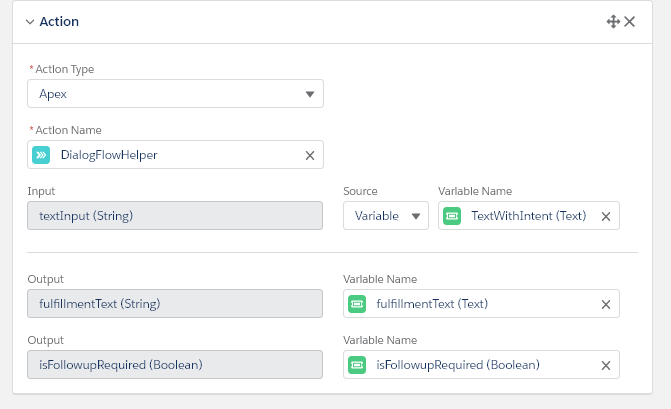
If a follow up question is required, we call a ‘Wait for follow up’ dialog



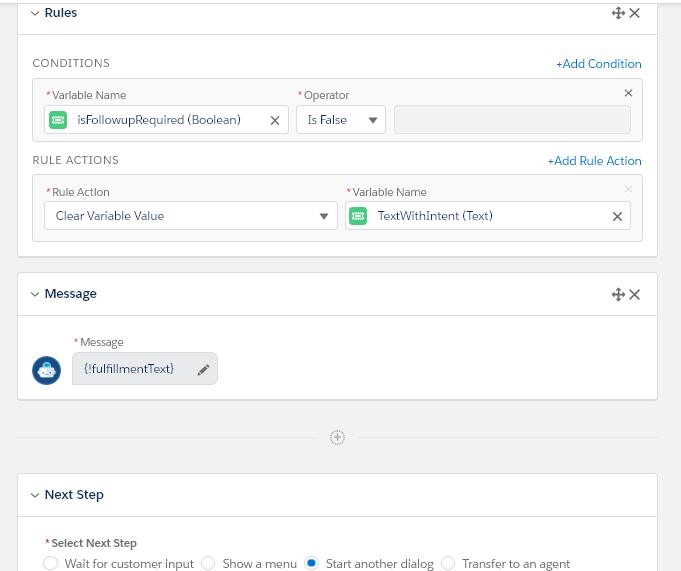
As a part of Wait for follow-up, we display the fulfillment text so that the user can enter all the required parameters.



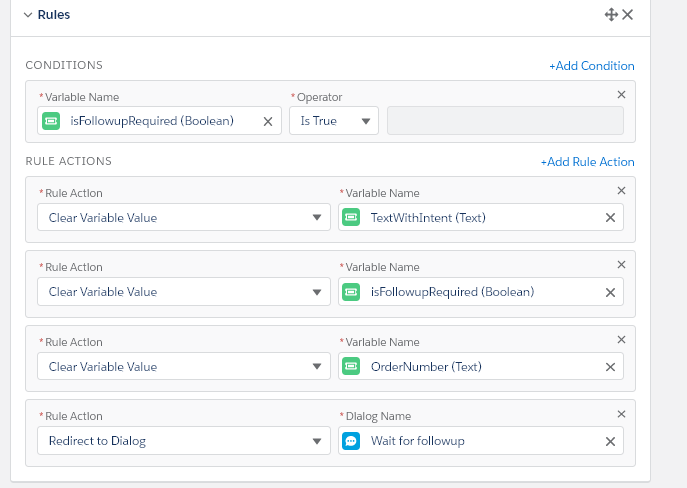
And then call the same apex action with the user input



Then, if follow up questions are no required, we display the appropriate message and start the ‘Connect to dialogflow’ dialog again



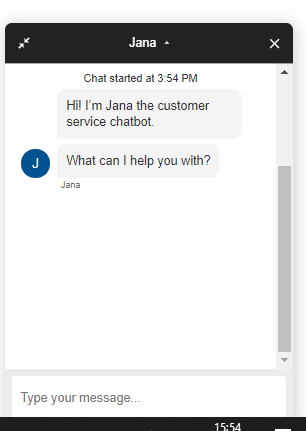
If a follow up is required, we call the same ‘Wait for follow up’ dialog again

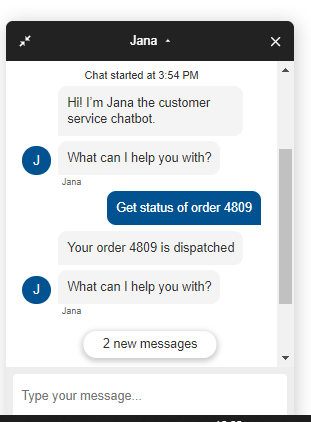


When the action has all the required parameters, it will perform the apex action and display the appropriate result.

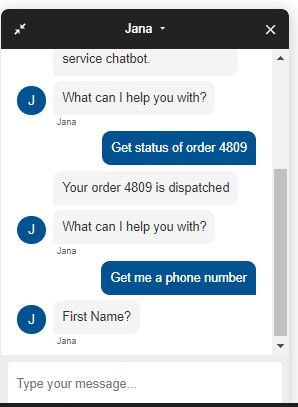
**In action**

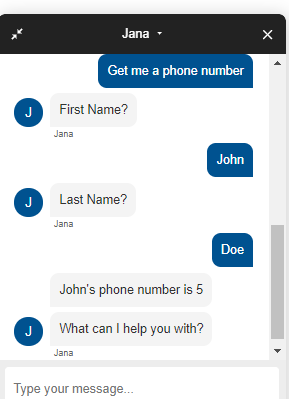
When all parameters are supplied:

****

****

**Bot asking for both parameters:**

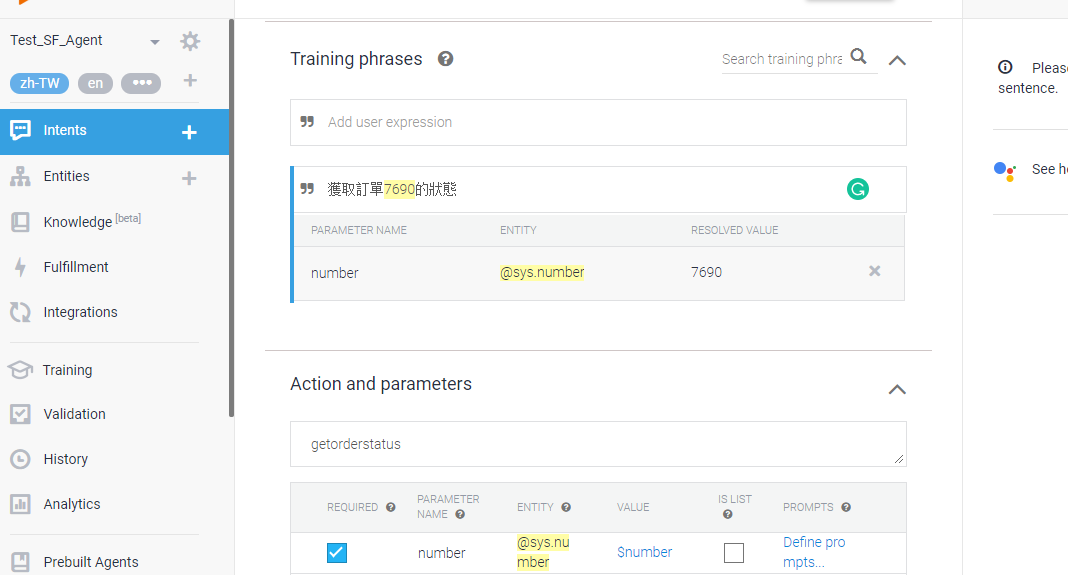
****

****

**Creating a multilingual agent**

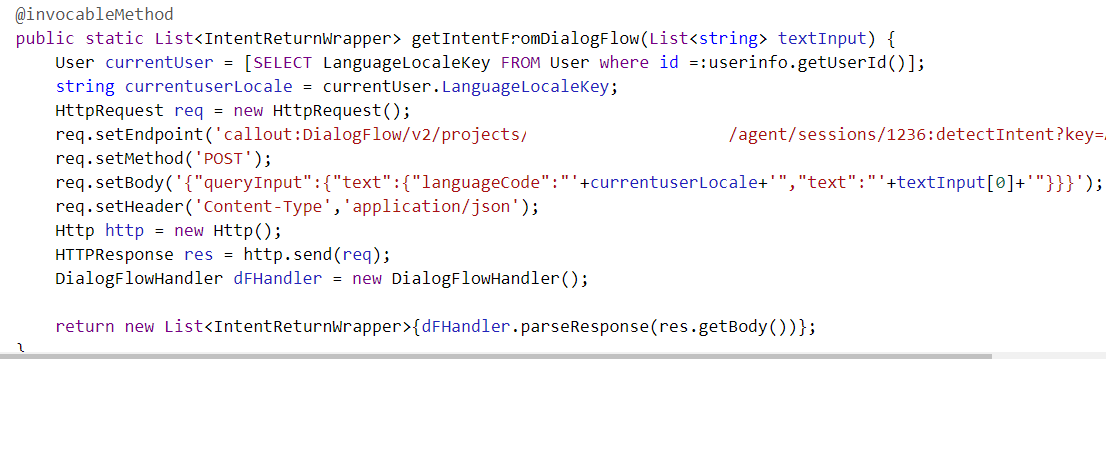
You can add multiple languages to your agent in dialogflow. For the purpose of this demo we are adding the Chinese-Traditional language.

More details about adding languages to the agent can be found [here](https://cloud.google.com/dialogflow/docs/agents-multilingual).



In the above screenshot, we add training phrases and responses for the language that is chosen on the top right under the agent name (zh-TW, in this case)

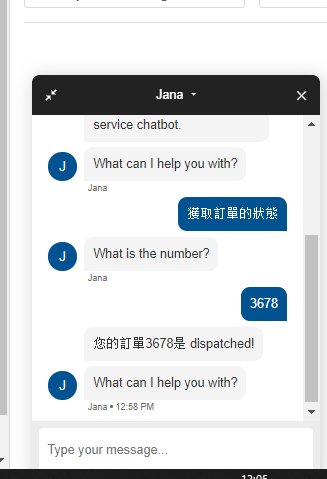
If you check the request that we send to dialogflow, we are specifying the language code that the input text is in.



Here, we use the logged in user’s locale to specify the language. We could also use Google’s Translation API to detect the language.

Also, we directly use the fulfillment message that we get in the response so that the output can be shown in the same language as the input.

For this example, we have left the prompt of order number in English.



To change the order number prompt we can change the text in dialogflow here:  
