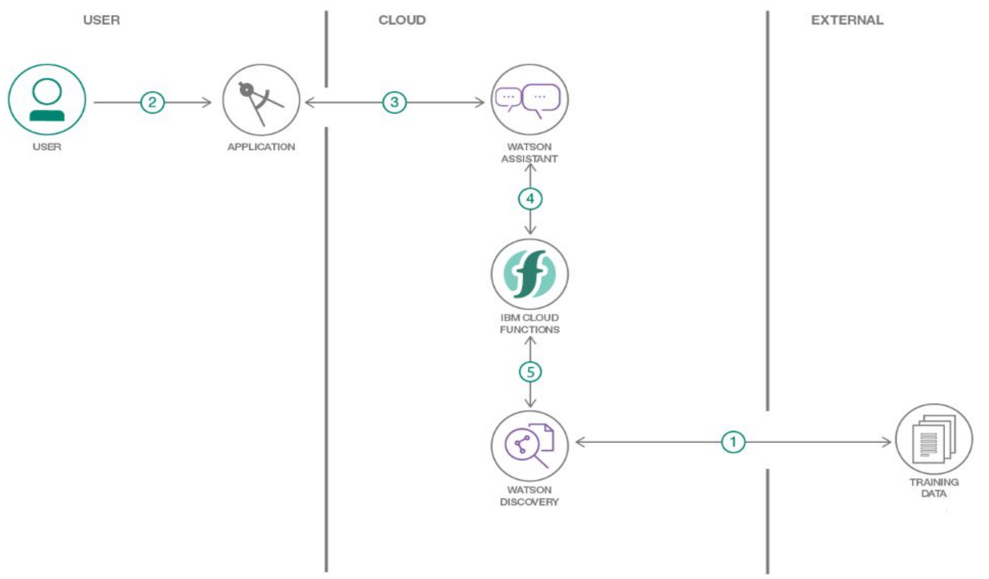
Project Documentation

Flow



1. The document is annotated using Watson Discovery SDU
2. The user interacts with the backend server via the app UI. The frontend app, made using RedNode UI is a chatbot that engages the user in a conversation.
3. The Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
4. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
5. The Cloud Functions action will query the Watson Discovery service and return the results.

Steps to Build the Project:

1.Create an account on IBM Cloud service.

https://www.ibm.com/watson

2.Create a Watson Discovery Service:

Import the document of your choice which contains all the answers to the common questions expected from the user.

Apply SDU to the document so that it can produce a more accurate answer. (From the Discovery collection panel, click the Configure data button (located in the top right corner) to start the SDU process.)

As you go through the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages. For this specific owner's manual, we mark the following:

- The main title page as the title
- The table of contents (shown in the first few pages) as table_of_contents
- All headers and sub-headers (typed in light green text) as a subtitle
- All page numbers as footers
- All warranty and licensing information (located in the last few pages) as a footer
- All other text should be marked as text.

3. Create IBM Cloud Functions Action:

Now create the web action that will make queries against our Discovery collection.

Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter and then select the Functions card.

From the Function’s main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option.

On the Create Action panel, provide a unique Action Name, keep the default package, and select the Node.js 10 runtime. Click the Create button to create the action. Once your action is created, click on the Code tab:

In the code editor window, type the code that simply connects to the Discovery service, makes a query against the collection, then returns the response.

Now add the credentials.

Add the following keys:

- url
- environment_id
- collection_id
- iam_apikey

For values, please use the values associated with the Discovery service you created in the previous step.

Now that the credentials are set, return to the Code panel and press the Invoke button again. Now you should see actual results returned from the Discovery service:

Next, go to the Endpoints panel

Click the checkbox for Enable as Web Action. This will generate a public endpoint URL.

Take note of the URL value, as this will be needed by Watson Assistant in a future step.

4.Launch the Watson Assistant:

Add new intent

The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this. Create a new intent that can detect when the user is asking about operating the Ecobee thermostat. From the Customer Care Sample Skill panel, select the Intents tab.

Click the Create intent button.

Name the intent #Product_Information, and at a minimum, enter the following example questions to be associated with it.

Create a new dialog node

Now we need to add a node to handle our intent. Click on the Dialog tab, then click on the drop-down menu for the Small Talk node, and select the Add node below option.

Name the node "Ask about the product" and assign it to our new intent.

This means that if Watson Assistant recognizes a user input such as "how do I set the time?", it will direct the conversation to this node.

Enable webhook from Assistant

Set up access to our WebHook for the IBM Cloud Functions action you created in Step #4.

Select the Options tab :

Enter the public URL endpoint for your action.

Important: Add .json to the end of the URL to specify the result should be in JSON format.

Return to the Dialog tab, and click on the Ask about product node. From the details panel for the node, click on Customize, and enable Webhooks for this node:

Click Apply.

The dialog node should have a Return variable set automatically to $webhook_result_1. This is the variable name you can use to access the result from the Discovery service query. You will also need to pass in the user's question via the parameter input [2]. The key needs to be set to the value:

```
"<?input.text?>"
```

If you fail to do this, Discovery will return results based on a blank query.

Optionally, you can add these responses to aid in debugging:

Test in Assistant Tooling

From the Dialog panel, click the Try it button located at the top right side of the panel.

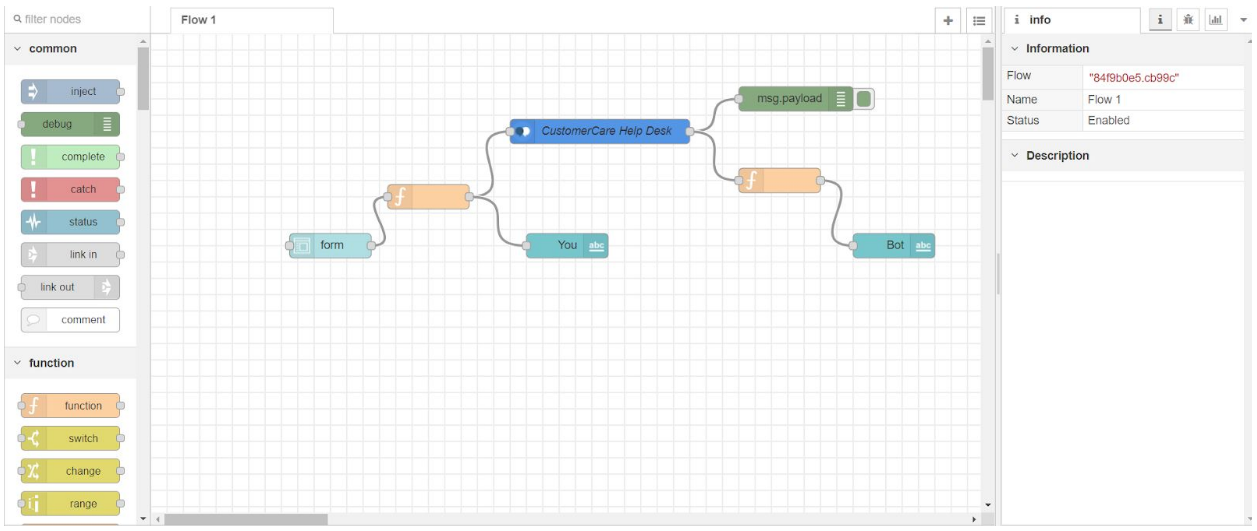
Enter some user input:

Note that the input "how do I turn on the heater?" has triggered our Ask about product dialog node, which is indicated by the #Product_Information response.

And because we specified that $webhook_result_1.passages be the response, that value is displayed also.

You can also verify that the call was successfully completed by clicking on the Manage Context button at the top right. The response from the Discovery query will be stored in the $webhook_result_1 variable:

5.Design the Node-Red Fow



Flow	"84f9b0e5.cb99c"
Name	Flow 1
Status	Enabled

Create a form to display the chatbot on the screen

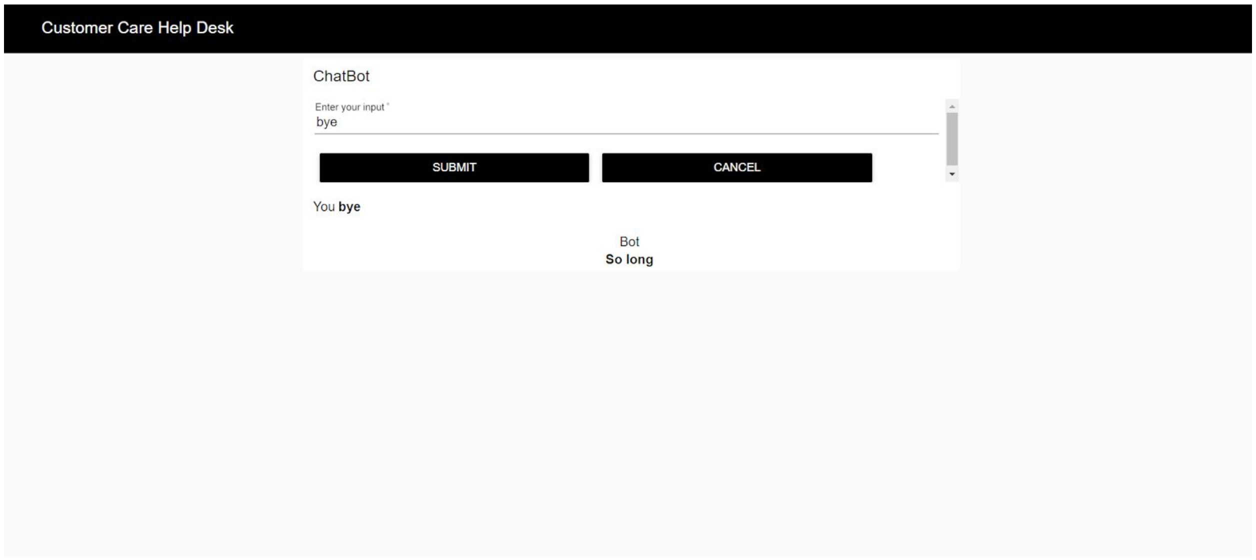
Create a function to input text from the user

Add the Watson Assistant by putting in the API key and other credentials

Create an output function to display the output the generated by Watson Assistant.

The final output will be displayed in another form.

Final UI



bye

You **bye**

Bot
So long