Intelligent Customer Help Desk with Smart Document Understanding

A project report

by

MANSI SOMAIYA

Category: Machine Learning

Application ID: SPS_APL_20200000680

Project ID: SPS_PRO_99

Internship at TheSmartBridge
May 2020

Mansi Somaiya
somaiyamansi@gmail.com
https://github.com/SmartPracticeschool/llSPS-INT-375-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding

Table of Contents

1. Introduction

1.1 Overview

A chatbot is a software application that is used to conduct an on-line chat conversation which simulates a human-like interaction. A chatbot is often described as one of the most advanced and promising expressions of interaction between humans and machines.

In this project, I have attempted to build a customer helpdesk or chatbot using IBM Cloud services. If the customer question to the helpdesk is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owner's manual. It returns relevant sections of the owner's manual to help solve the customers' problems. The project uses Smart Document Understanding feature of Watson Discovery to train it on what text in the owner's manual is important and what is not. This will improve the answers returned from the queries.

1.2 Purpose

The purpose of the project is to enhance the customer helpdesk so that it can answer operational questions which might otherwise require a representative or a consultant to answer them.

2. Literature Survey

2.1 Existing System Study

The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn’t valid or offer to speak to a real person.

Thus, the existing chatbots can offer little or no assistance with respect to the operational aspect of any product. Advancements in Artificial Intelligence and mainly Natural Language Processing have made way for systems that not only understand human queries but also effectively answer them.

2.2 Proposed System

The proposed solution is a chatbot that is built using IBM Cloud Services. It is a web application that utilizes multiple IBM Watson services to create a better customer care experience.

Using the Watson Discovery Smart Document Understanding (SDU) feature, we will enhance the Discovery model so that queries will be better focused to only search the most relevant information found in a typical owner's manual. Using Watson Assistant, we will use a standard customer care dialog to handle a typical conversation between a customer and a company representative. When a customer question involves operation of a product, the Assistant dialog will communicate with the Discovery service using a webhook. The webhook will be created by defining a web action using IBM Cloud Functions.

Smart Document Understanding (SDU)

SDU trains Watson Discovery to extract custom fields in your documents. Customizing how your documents are indexed into Discovery will improve the answers returned from queries. With SDU, you annotate fields within your documents to train custom conversion models. As you annotate, Watson is learning and will start predicting annotations. SDU models can also be exported and used on other collections. Current document type support for SDU is based on your plan.

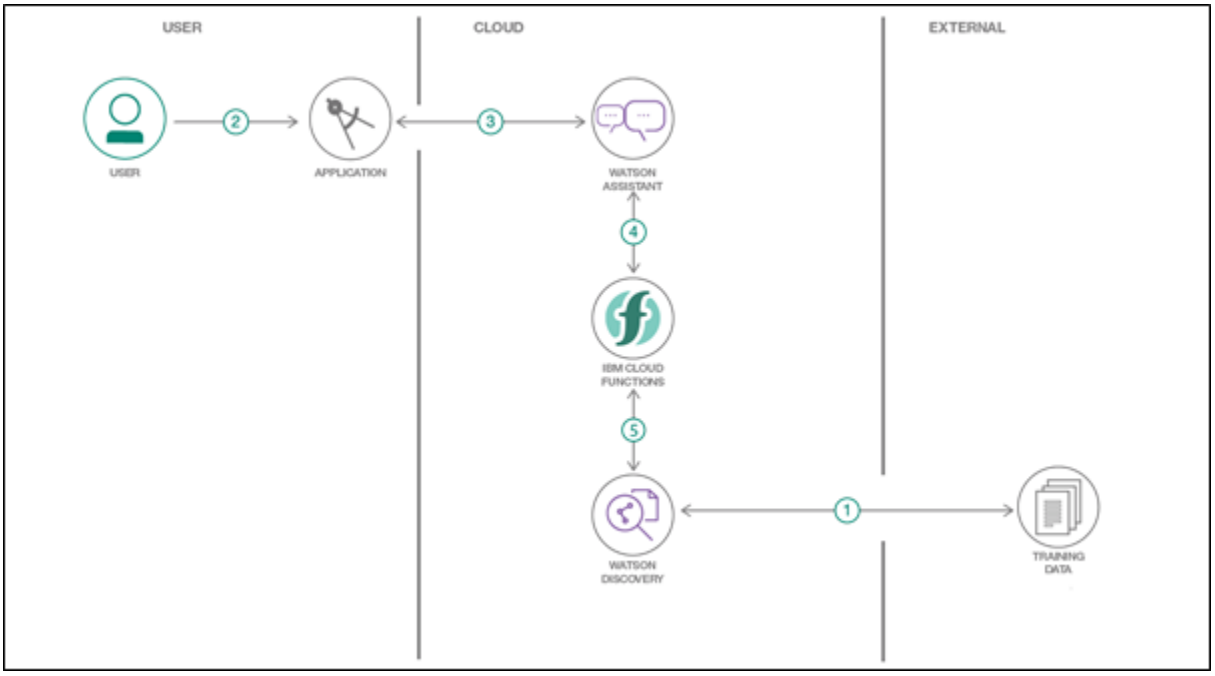
Lite plans: PDF, Word, PowerPoint, Excel, JSON, HTML
Advanced plans: PDF, Word, PowerPoint, Excel, PNG, TIFF, JPG, JSON, HTML

Webhook

A webhook is a mechanism that allows you to call out to an external program based on something happening in your program. When used in a Watson Assistant dialog skill, a webhook is triggered when the Assistant processes a node that has a webhook enabled. The webhook collects data that you specify or that you collect from the user during the conversation and save in context variables, and sends the data to the Webhook request URL as an HTTP POST request. The URL that receives the webhook is the listener. It performs a predefined action using the information that is provided by the webhook as specified in the webhook definition, and can optionally return a response.

In the proposed system, the webhook will communicate with an IBM Cloud Functions web action, which is connected to the Watson Discovery service.

2.3 System Architecture

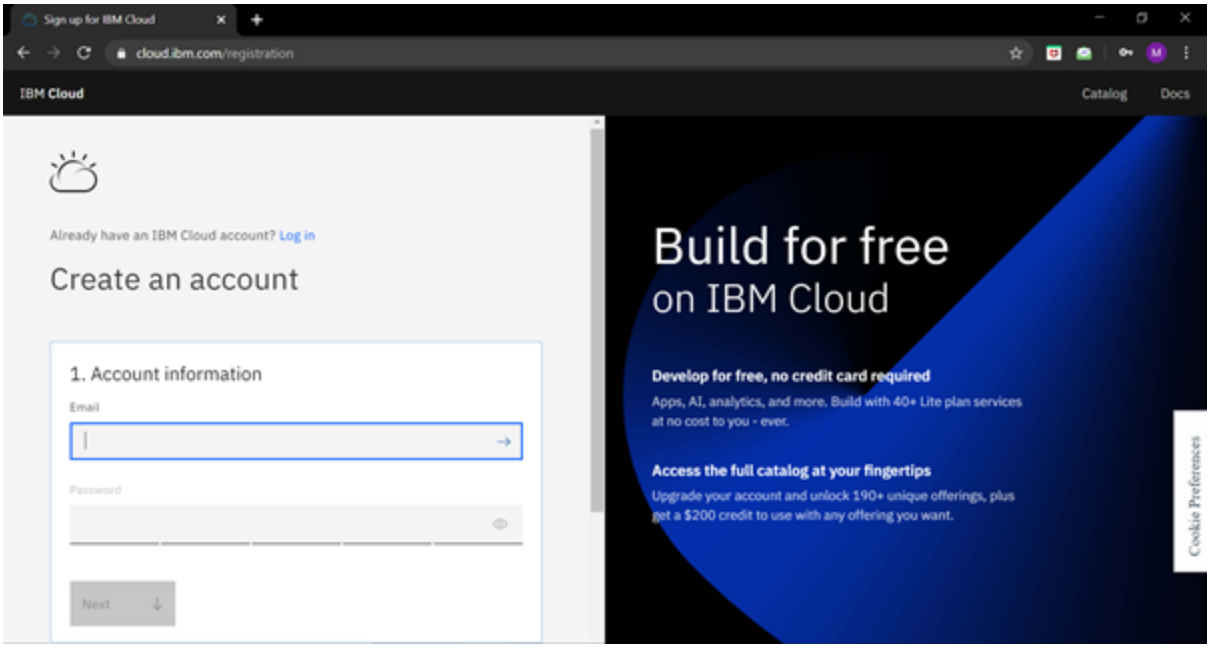


- The document is annotated using Watson Discovery SDU
- The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that engages the user in a conversation.
- Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
- If the user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

3. Methodology

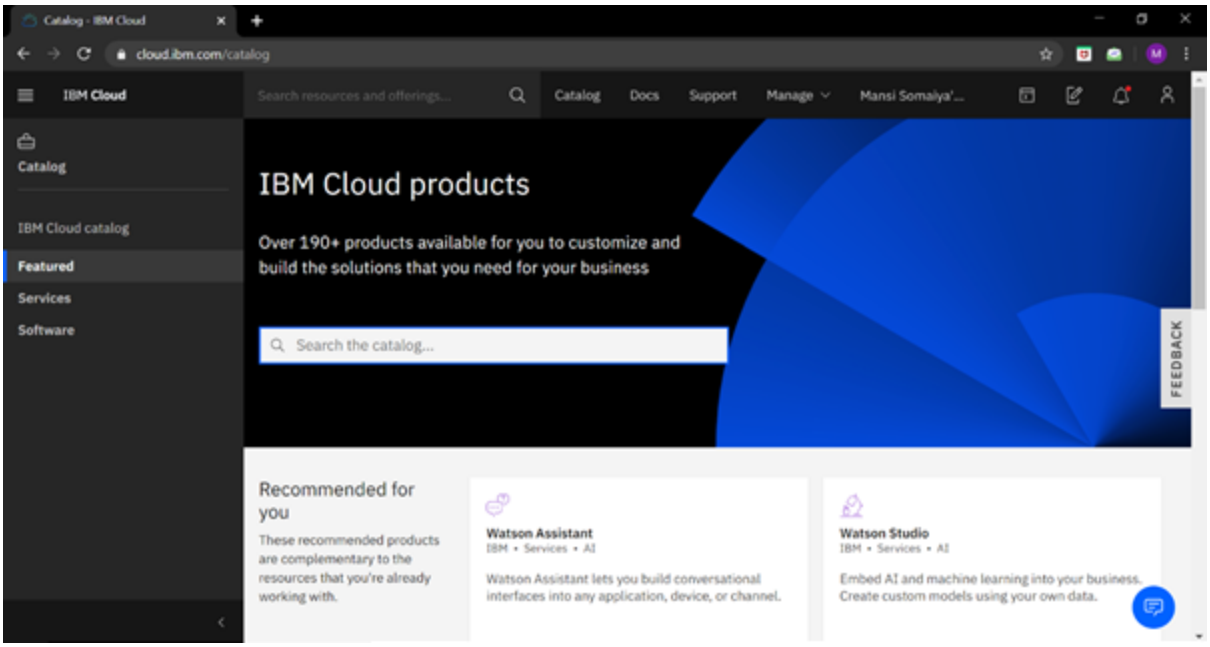
A. Create an IBM Cloud account

- Create an account by signing up on cloud.ibm.com or use an existing account.
- For this project, I created an IBM Cloud Lite account.



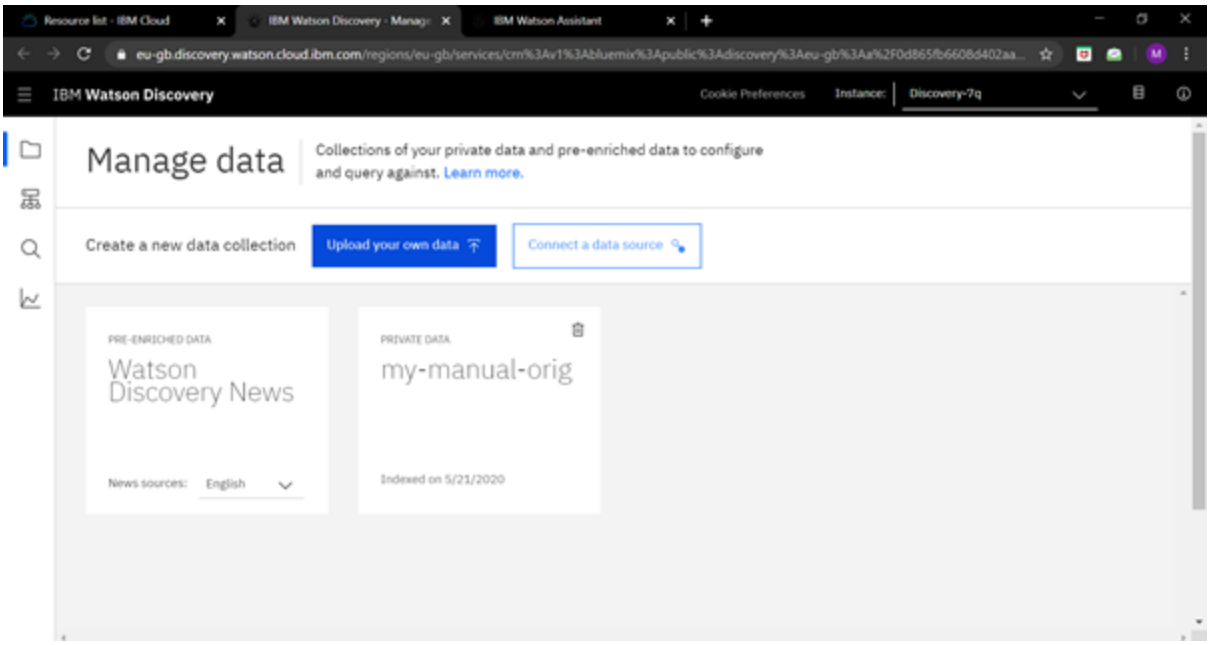
B. Create the necessary services

- Create a Watson Assistant instance
- Create a Watson Discovery instance
- Create a Node-RED starter application



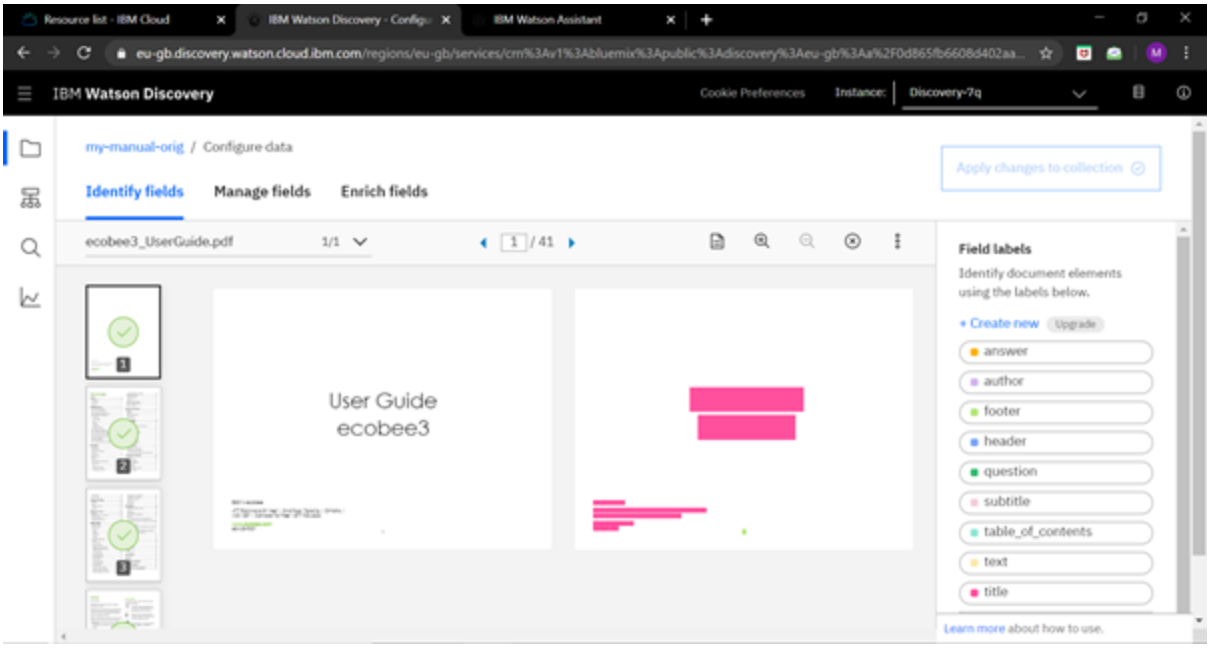
C. Configure Watson Discovery

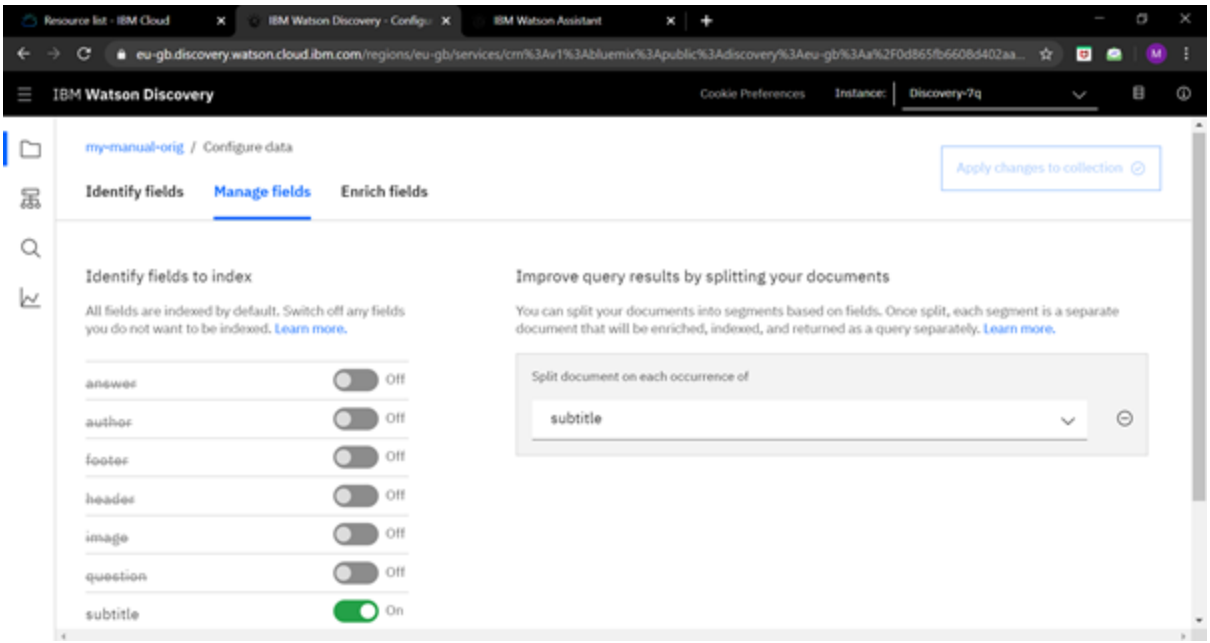
- Launch the Watson Discovery tool and create a new data collection by selecting the upload your own data option.

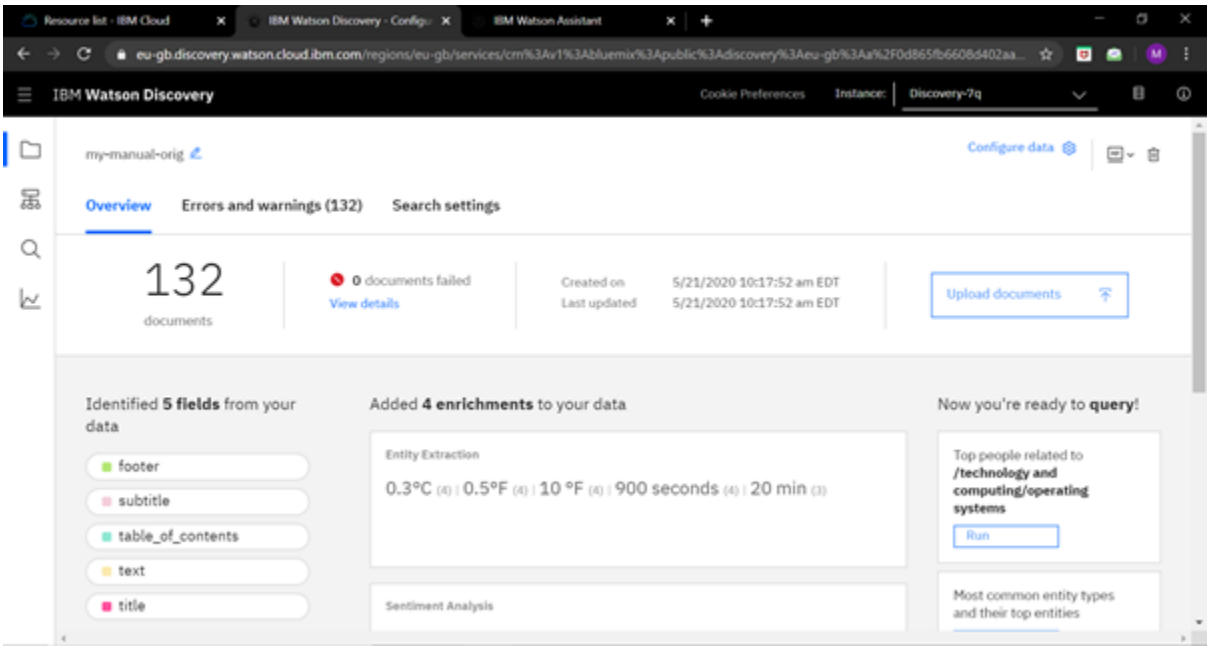


- Upload the Ecobee thermostat manual.
- Apply Smart Document Understanding to the document.
    - The goal is to annotate all of the pages in the document so that Discovery can learn what text is important, and what text can be ignored.

- As you go through the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages.

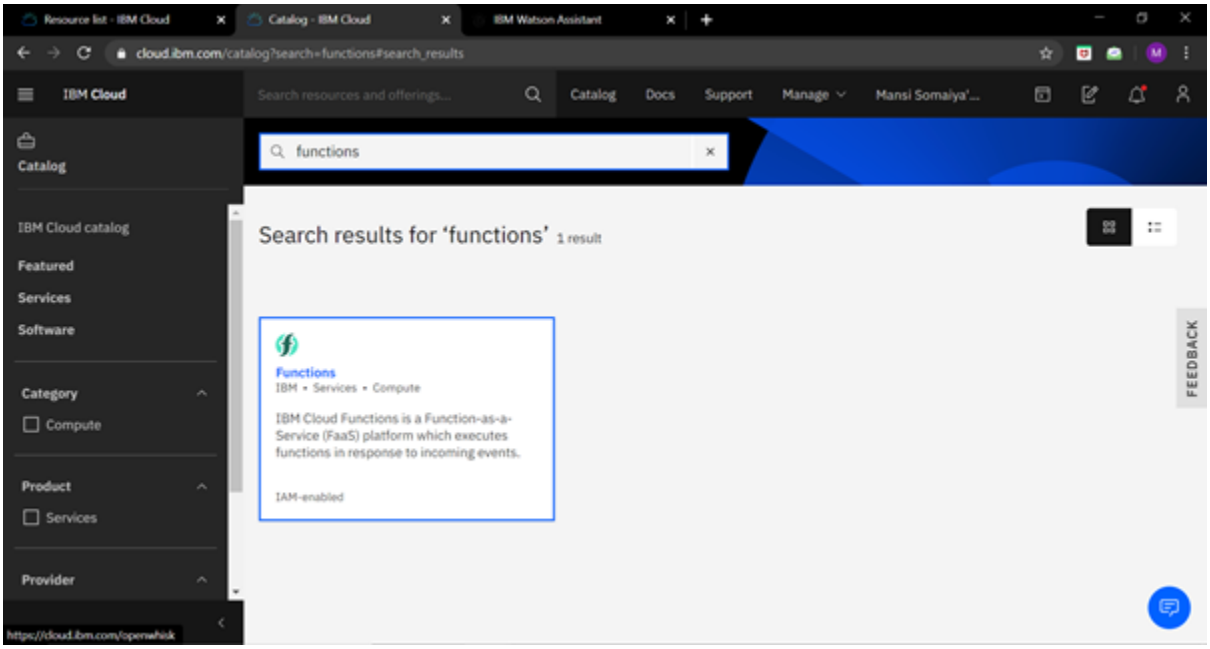




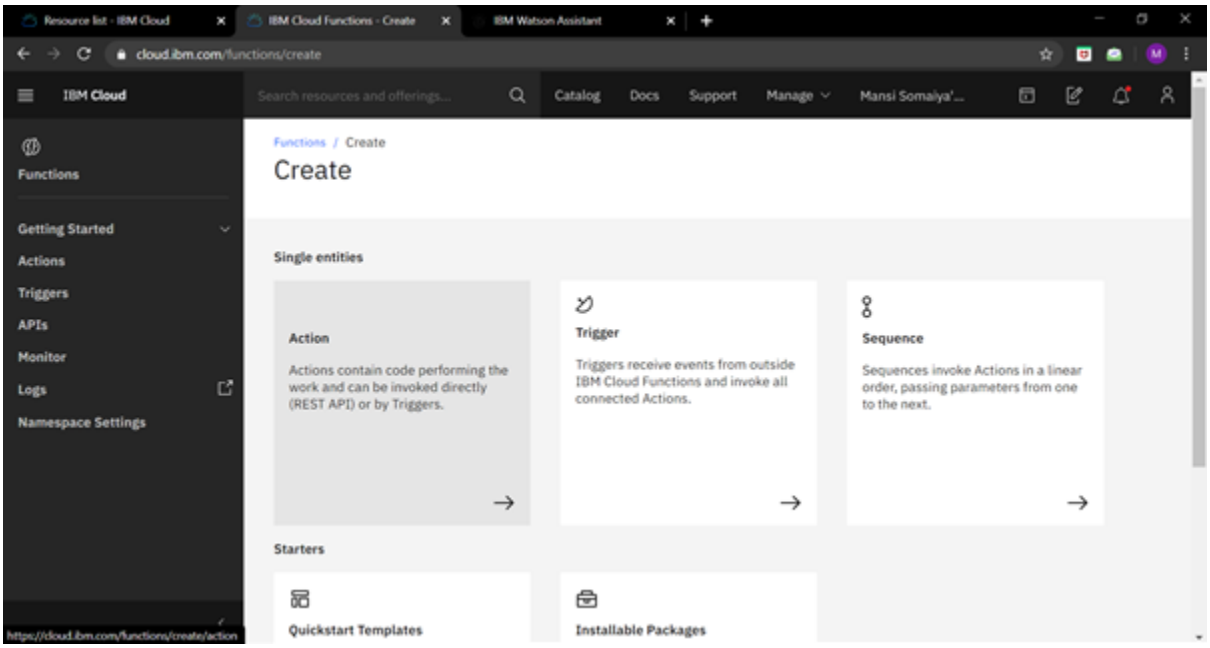


D. Create IBM Cloud Functions action

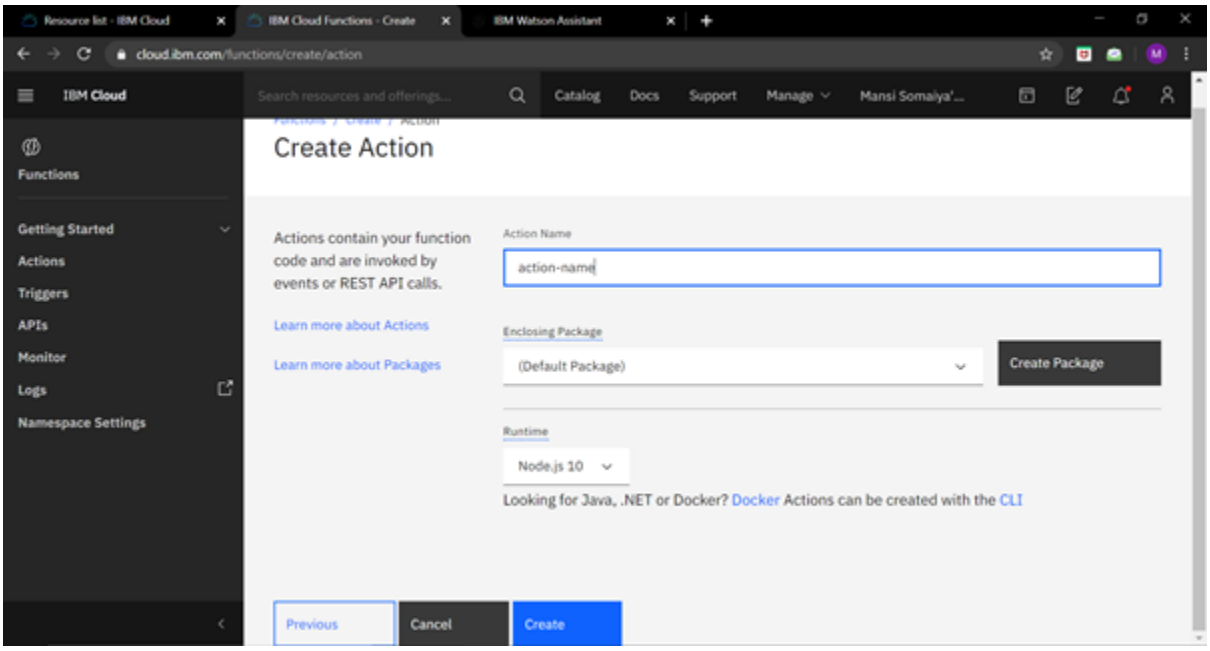
- Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard.



- From the Functions 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.

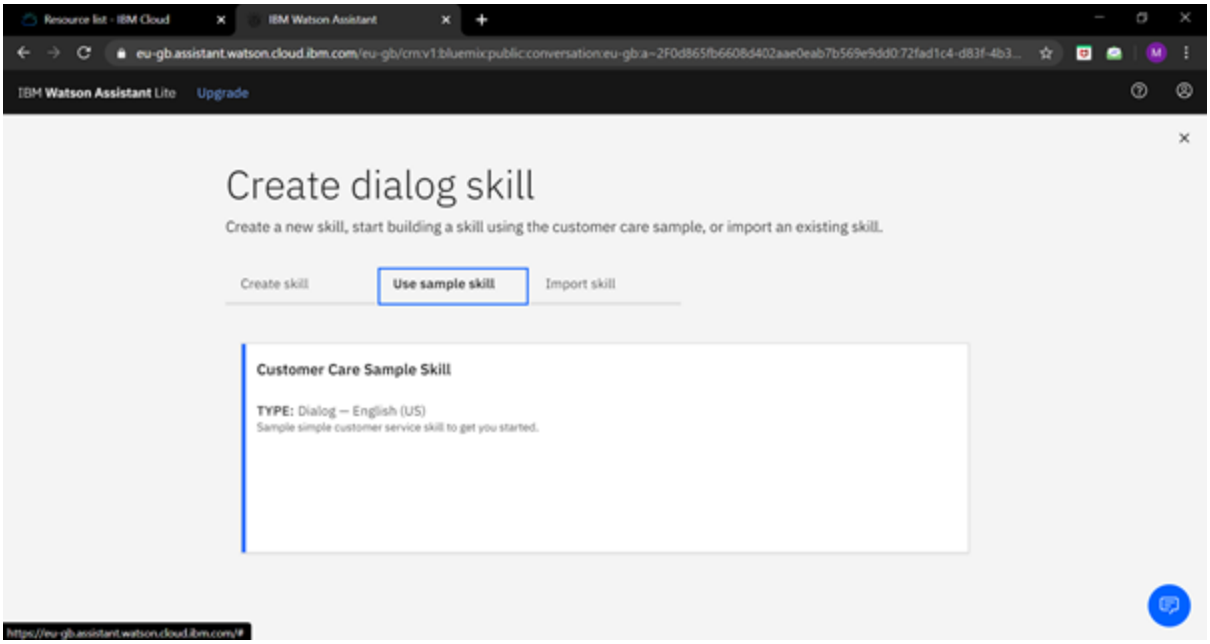


- Type the appropriate code.

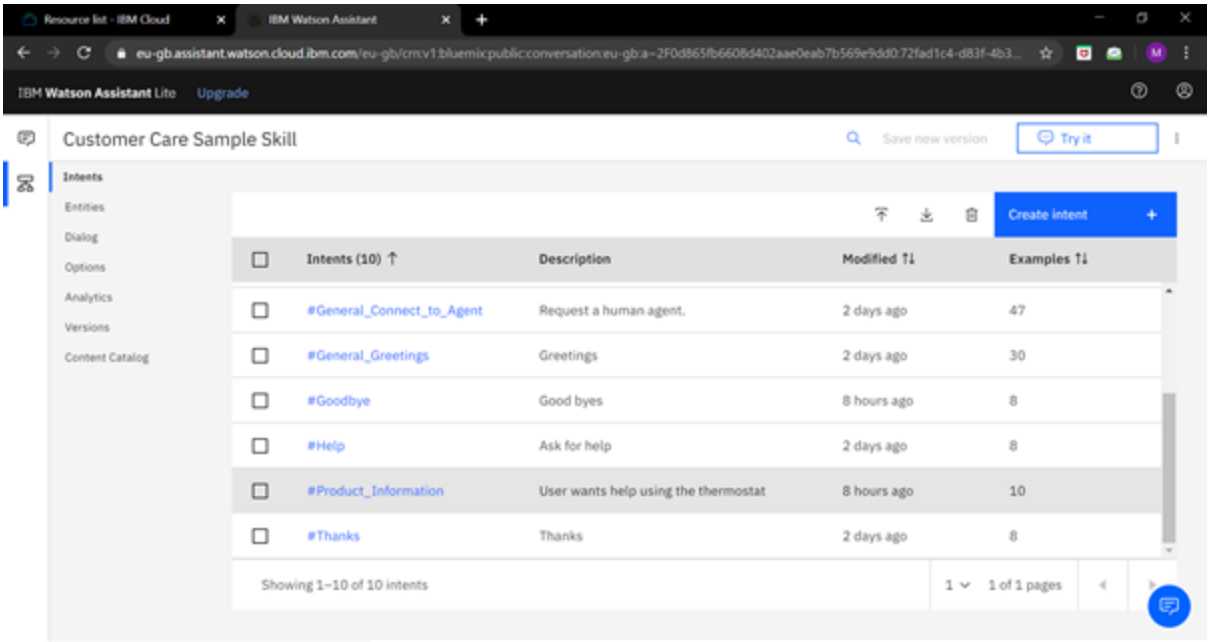
E. Configure Watson Assistant

- Launch the Watson Assistant Tool and create a new dialog skill. Select the use sample

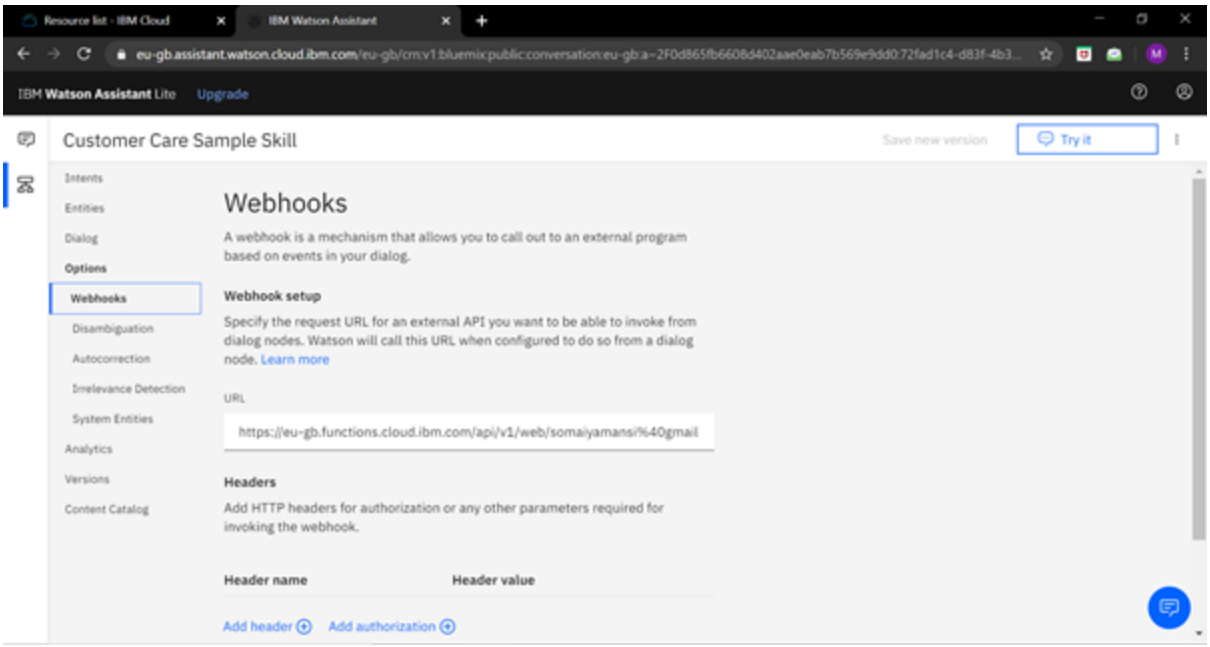
skill option as the starting point.

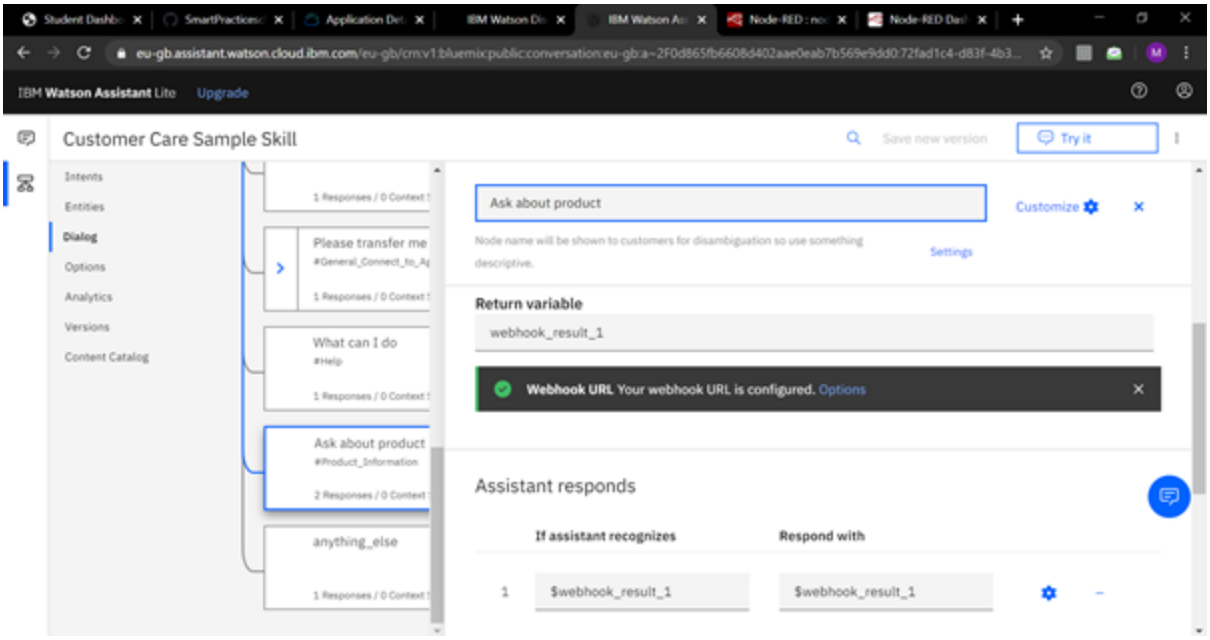


- Create a new intent that can detect when the user is asking about operating the Ecobee thermostat.
- From the Customer Care Sample Skill panel, select the Intents tab.
- Click the Create intent button. Name the intent #Product_Information, and enter the example questions to be associated with it.



- Create a new Dialog node, name it “ask about product” and assign it our new intent.
- Now select the Options tab and enable webhook for the Cloud Function created above.

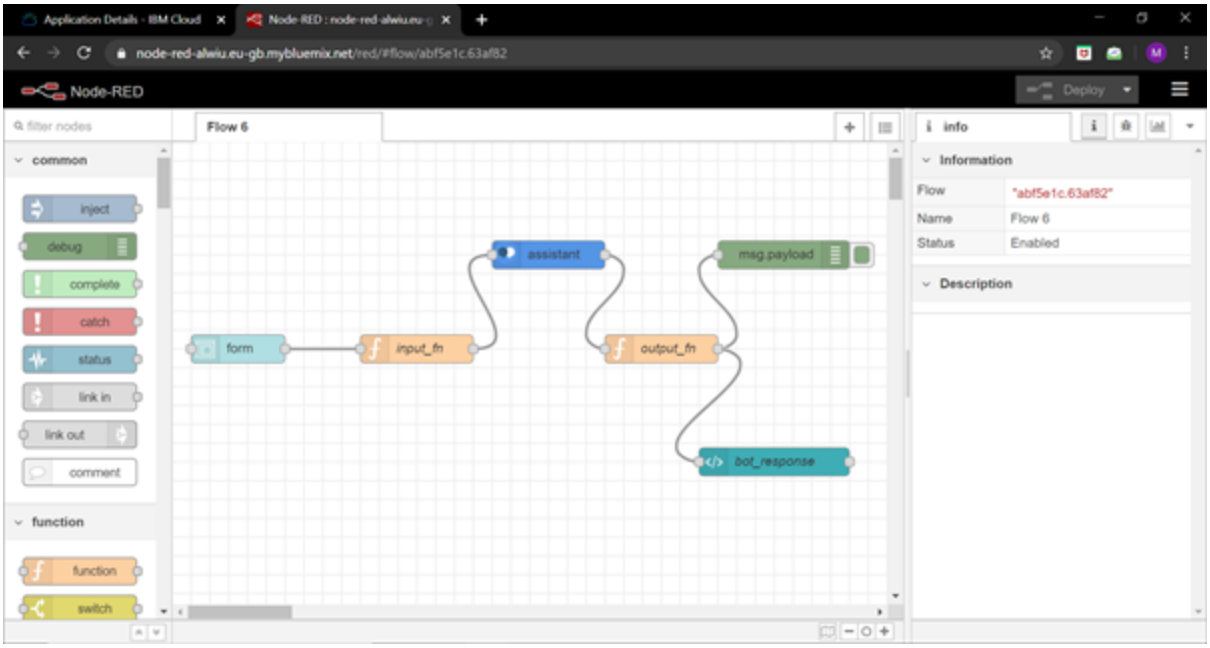




F. Build a Node-RED Flow to integrate all services

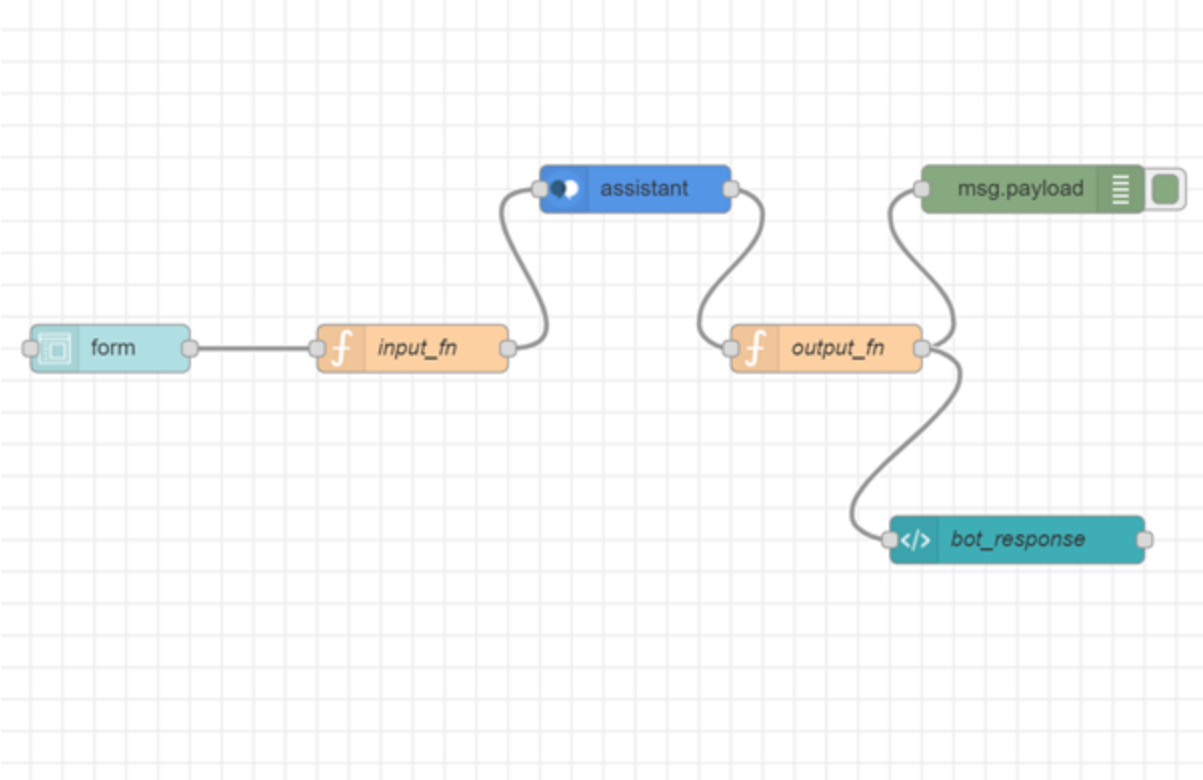
- The form node takes an input query from the user and parses it to the assistant node using the function node.

- The assistant (Watson Assistant instance created above) processes the query along with Watson Discovery and returns the result.
- The result is parsed using the function node and displayed using a template node. A debug node is also used.



G. Configure the nodes and build a web dashboard in Node-RED

- Configure the nodes by writing the appropriate code.



- Add a text field to get user input using the form node.
- Code for input_fn

```
msg.payload=msg.payload.query;
return msg;
```

- Code for output_fn

```
msg.payload.text="";
if(msg.payload.context.webhook_result_1)
{
for(var i in msg.payload.context.webhook_result_1.results)
{
msg.payload.text=msg.payload.text+"\n"+msg.payload.context.webhook_result_1
.results[i].text;
}
msg.payload=msg.payload.text;
}
else
msg.payload = msg.payload.output.text[0];
return msg;
```

- Code for bot_response - to display bot's response on the UI

```
<div style="height: 100px; color:mediumseagreen">
```

{{msg.payload}}

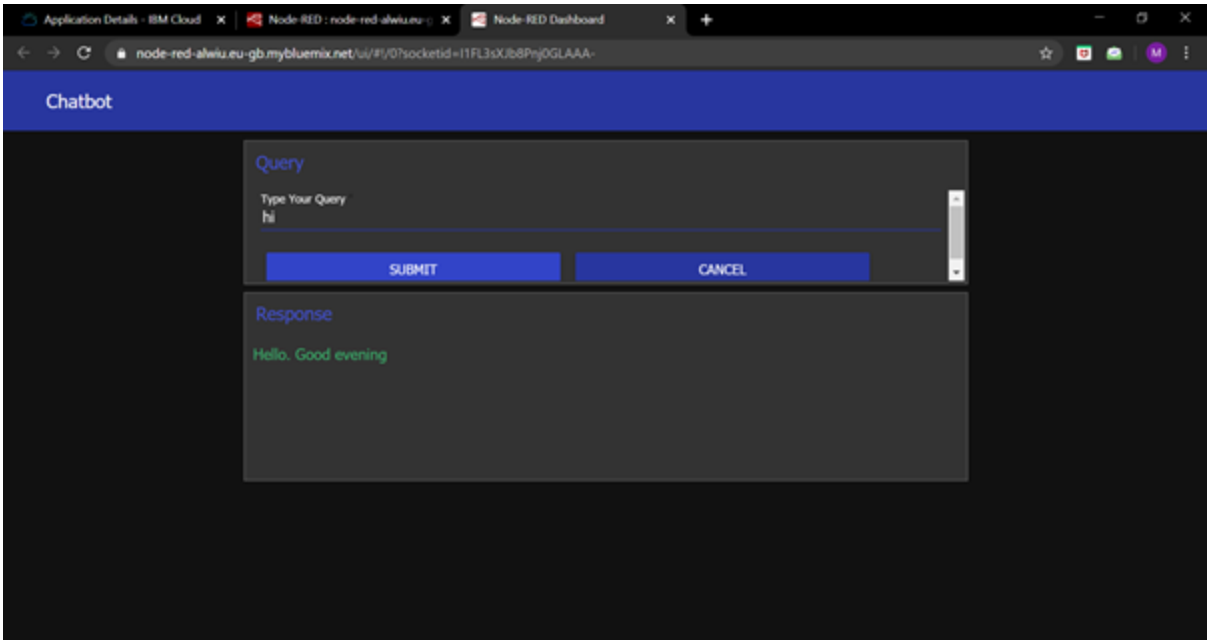
* *

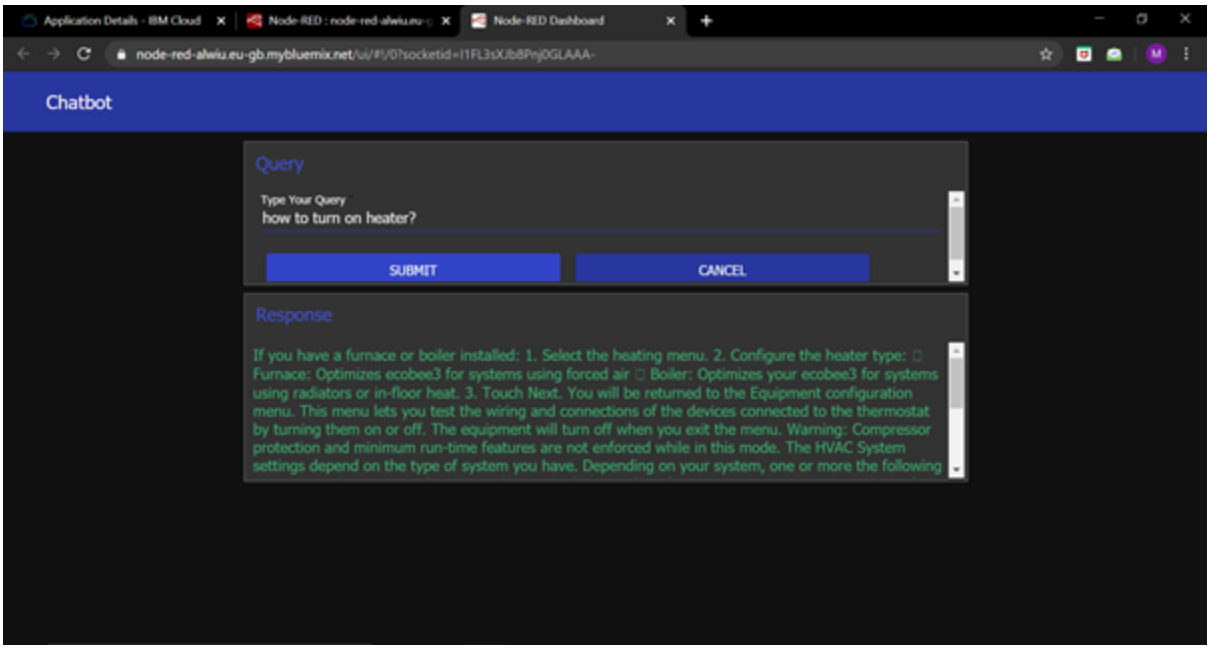
- Configure the assistant by adding the following details: API Key, Service Endpoint, Workspace ID.

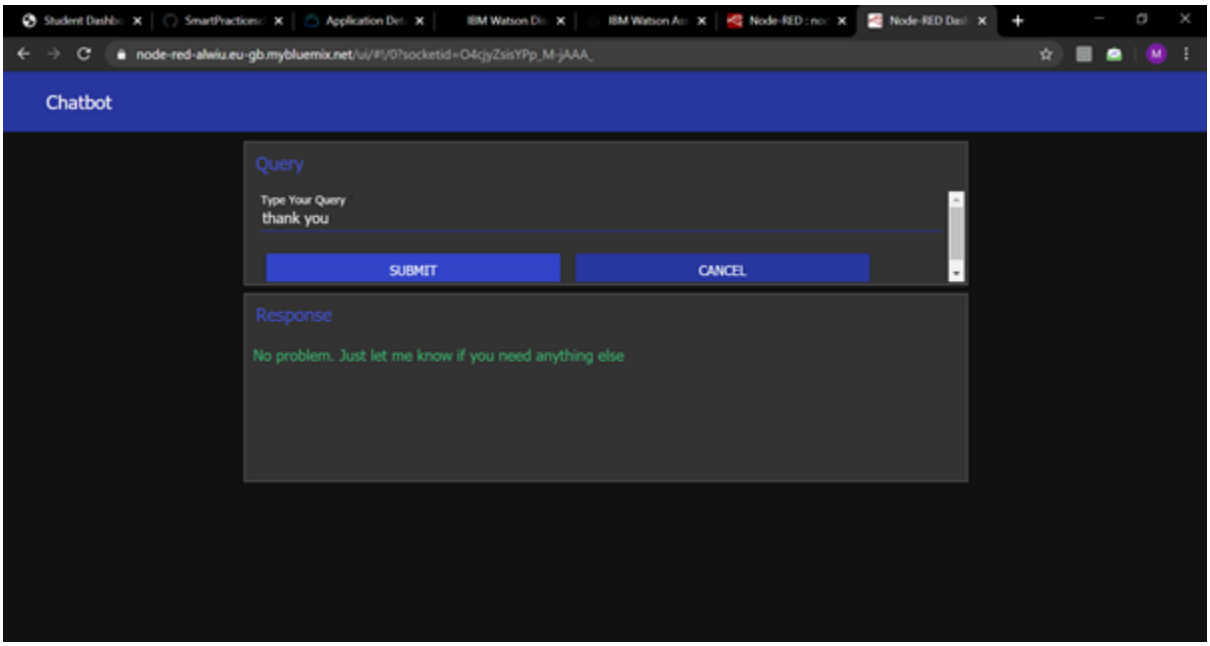
H. Deploy and run the application

- Deploy the application by using the deploy button on the top right of Node-RED editor.
- Go to the UI to test the application.

4. Result


Application Details - IBM Cloud
Node-RED : node-red-alwiu.eu-g
Node-RED Dashboard
node-red-alwiu.eu-gb.mybluemix.net/ui/#!/0?socketid=I1FL3sXJb8PnjOGLAAA-
Chatbot
Query
Type Your Query
hi
SUBMIT
CANCEL
Response
Hello. Good evening


Application Details - IBM Cloud
Node-RED : node-red-alwiu.eu-g
Node-RED Dashboard
node-red-alwiu.eu-gb.mybluemix.net/ui/#!/0?socketid=I1FL3sXJb8PnjOGLAAA-
Chatbot
Query
Type Your Query
how to turn on heater?
SUBMIT
CANCEL
Response
If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: Furnace: Optimizes ecobee3 for systems using forced air Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3. Touch Next. You will be returned to the Equipment configuration menu. This menu lets you test the wiring and connections of the devices connected to the thermostat by turning them on or off. The equipment will turn off when you exit the menu. Warning: Compressor protection and minimum run-time features are not enforced while in this mode. The HVAC System settings depend on the type of system you have. Depending on your system, one or more the following



The proposed system is able to satisfactorily respond to user queries about the operation of the thermostat among other queries.

5. Advantages & Disadvantages

Advantages:

- Faster customer service
- Increased customer satisfaction
- Lower labour costs
- Data collection
- 24x7 availability
- Multiple customer handling
- Can be integrated with various external services

Disadvantages:

- Limited responses for customers
- Customers could become frustrated
- Maintenance required
- They aren’t human ultimately so cannot be reliable in case of emergencies

6. Future Scope

- The helpdesk can be extended to fulfil a lot of other functionalities as required by the company.
- It can be trained on more scenarios so that it can respond efficiently.
- It can be integrated with external applications for wide use.

7. Conclusion

We have successfully created and tested an intelligent customer helpdesk using various IBM Cloud services like IBM Watson Discovery, IBM Watson Assistant and IBM Cloud Function. It is able to answer various user queries and also queries pertaining to the operation of the Ecobee thermostat.

References

1. https://www.ibm.com/cloud/get-started
2. https://developer.ibm.com/patterns/enhance-customer-help-desk-with-smart-document-understanding/
3. https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
4. https://github.com/IBM/watson-discovery-sdu-with-assistant
5. https://cloud.ibm.com/docs/openwhisk?topic=cloud-functions-getting-started
6. Cloud function code for integrating Watson Discovery and webhook: http://p.ip.fi/cCCT