Project Report
on
Intelligent Customer Helpdesk with Smart Document Understanding
in
Artificial Intelligence
by
Aneri Shah.

1. Introduction

1.1Overview

I have built a chatbot that uses various Watson AI services such as Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-RED to deliver a Web based UI to contact the bot.

I have integrated Watson Discovery service with Watson Asssistant using webhooks.

- **Technical Requirements:** AI, ML, Watson AI, Node JS.
- **Project Requirements**: Node-RED, IBM Cloud, IBM Watson, Node JS
- **Software Requirements**:Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED
- **Project Deliverables**: Intelligent Customer Help Desk with Smart Document Understanding.
- **Project Team**: Aneri Shah
- **Project Duration**: 20 days
- **Scope of work**:
- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery.
- Build a web application with integration to all these services & deploy the same on IBM Cloud Platform.

.

1.2. Purpose

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.In this project, there will be another option. If the customer question 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 owners manual. So now, instead of “Would you like to speak to a customer representative?” we can return relevant sections of the owners manual to help solve our customers’ problems. To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owners manual is important and what is not. This will improve the answers returned from the queries.

2. Literature Survey

2.1. Existing Problem

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.

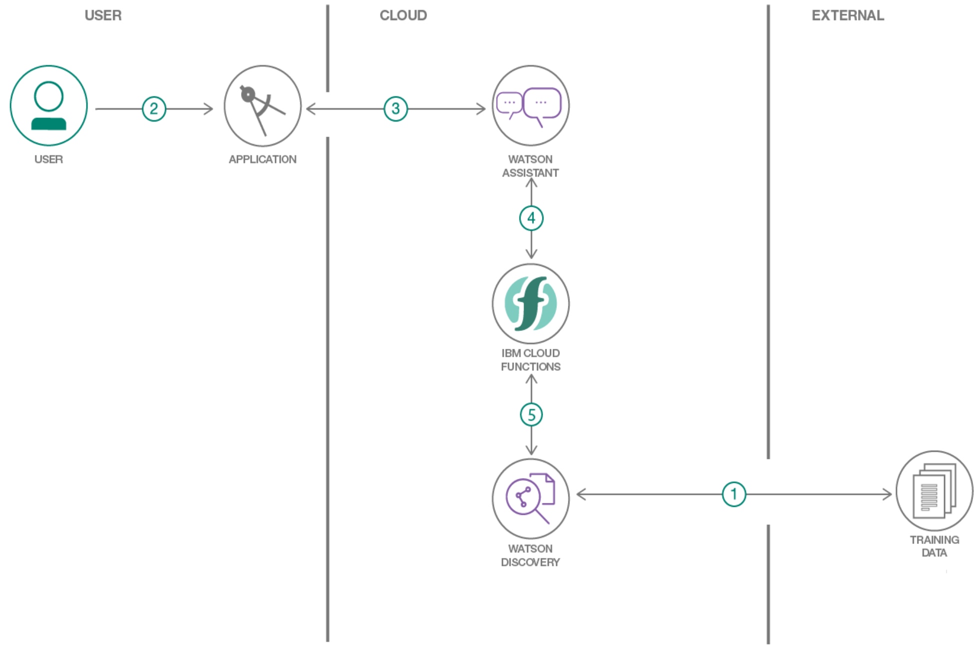
2.2. Proposed Solution

In this project, there will be another option. If the customer's question 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 owners manual. So now, instead of “Would you like to speak to a customer representative?” we can return relevant sections of the owners manual to help solve our customers’ problems. So unless and untill customer specifically asks for a customer representative the bot will try to solve all your queries.

To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owners manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as webhook, Watson Discovery can be integrated with Watson assistant. Finally using Node-Red, Watson assistant can be integrated with a web UI. This UI can then be used to connect with Watson assistant and chat with it.

3. Theoretical Analysis

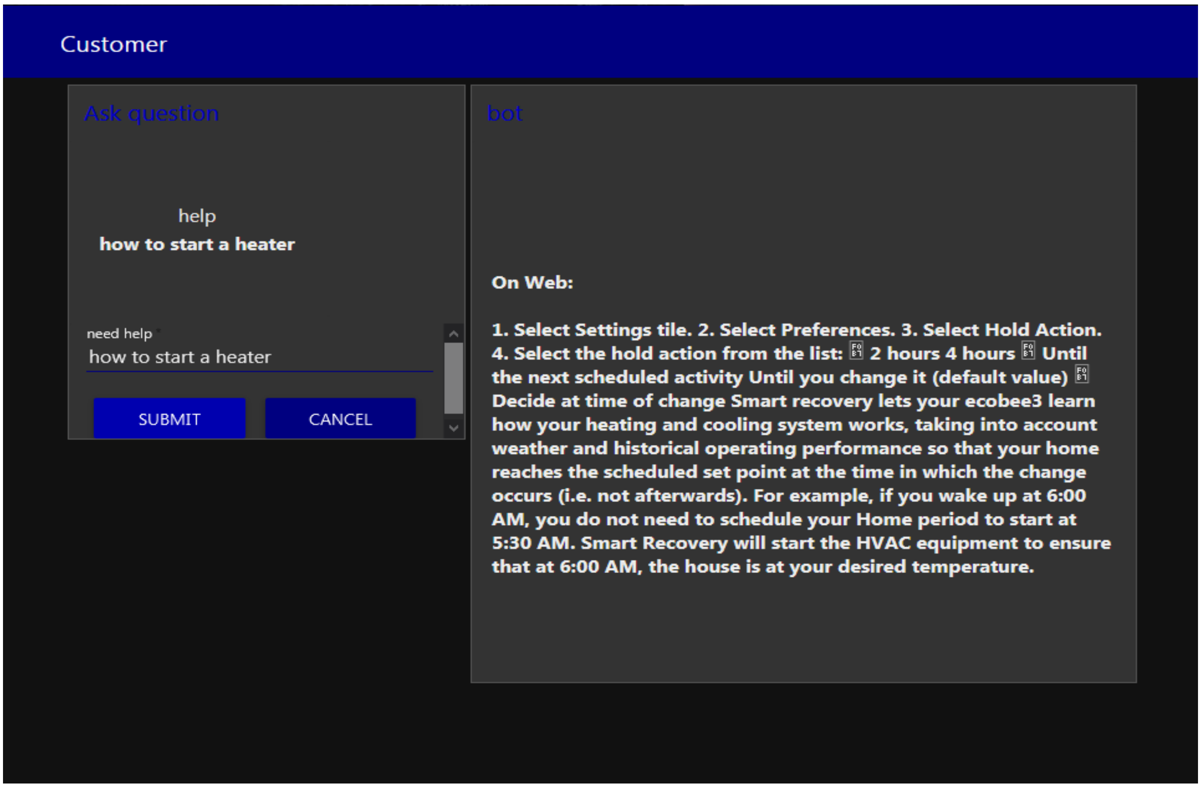
Block / Flow Diagram



Hardware / Software Designing

1. Create necessary Watson Services.
2. Configure Watson Discovery.
3. Create Watson Cloud Functions Action.
4. Configure Watson Assistant.
5. Integrate Watson Discovery with Watson Assistant using webhook.
6. Build Node-RED flow to integrate Watson Assistant and Web Dashboard.

4. Experimental Investigation



help

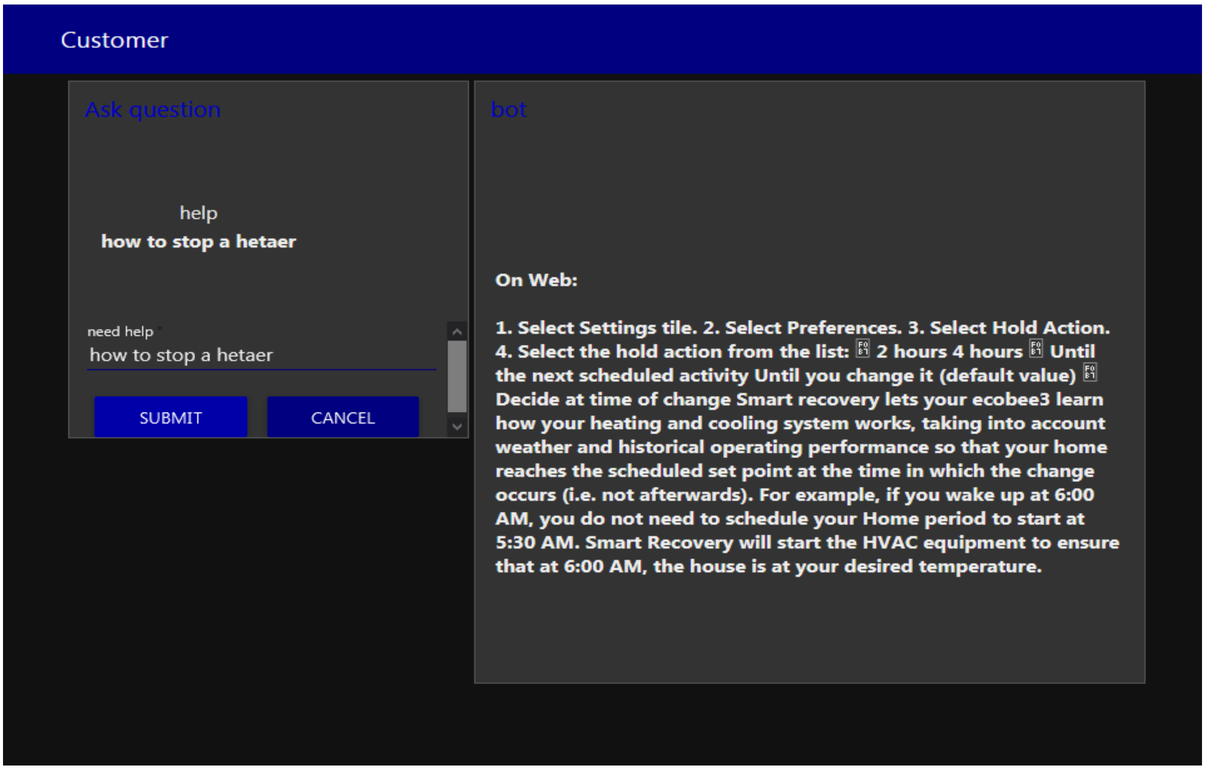
how to start a heater

need help *

how to start a heater

On Web:

1. Select Settings tile. 2. Select Preferences. 3. Select Hold Action. 4. Select the hold action from the list: 2 hours 4 hours Until the next scheduled activity Until you change it (default value) Decide at time of change Smart recovery lets your ecobee3 learn how your heating and cooling system works, taking into account weather and historical operating performance so that your home reaches the scheduled set point at the time in which the change occurs (i.e. not afterwards). For example, if you wake up at 6:00 AM, you do not need to schedule your Home period to start at 5:30 AM. Smart Recovery will start the HVAC equipment to ensure that at 6:00 AM, the house is at your desired temperature.



help

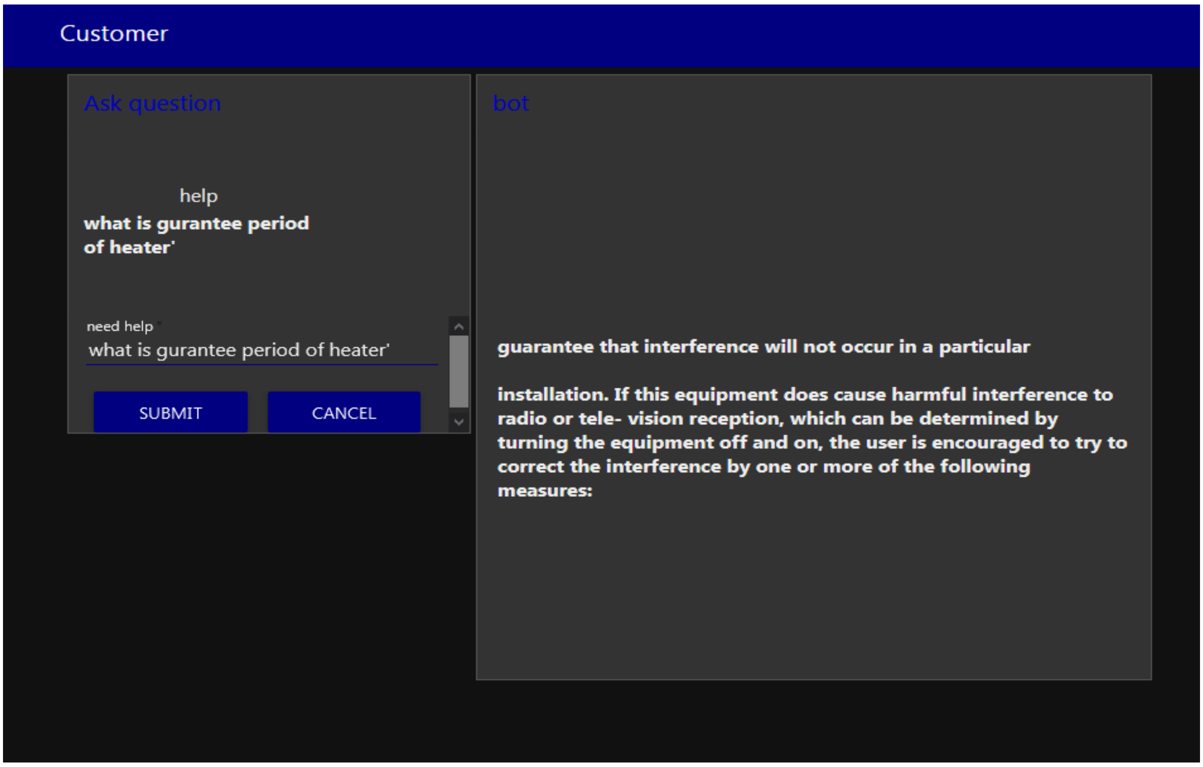
how to stop a hetaer

need help *

how to stop a hetaer

On Web:

1. Select Settings tile. 2. Select Preferences. 3. Select Hold Action. 4. Select the hold action from the list: 2 hours 4 hours Until the next scheduled activity Until you change it (default value) Decide at time of change Smart recovery lets your ecobee3 learn how your heating and cooling system works, taking into account weather and historical operating performance so that your home reaches the scheduled set point at the time in which the change occurs (i.e. not afterwards). For example, if you wake up at 6:00 AM, you do not need to schedule your Home period to start at 5:30 AM. Smart Recovery will start the HVAC equipment to ensure that at 6:00 AM, the house is at your desired temperature.



help

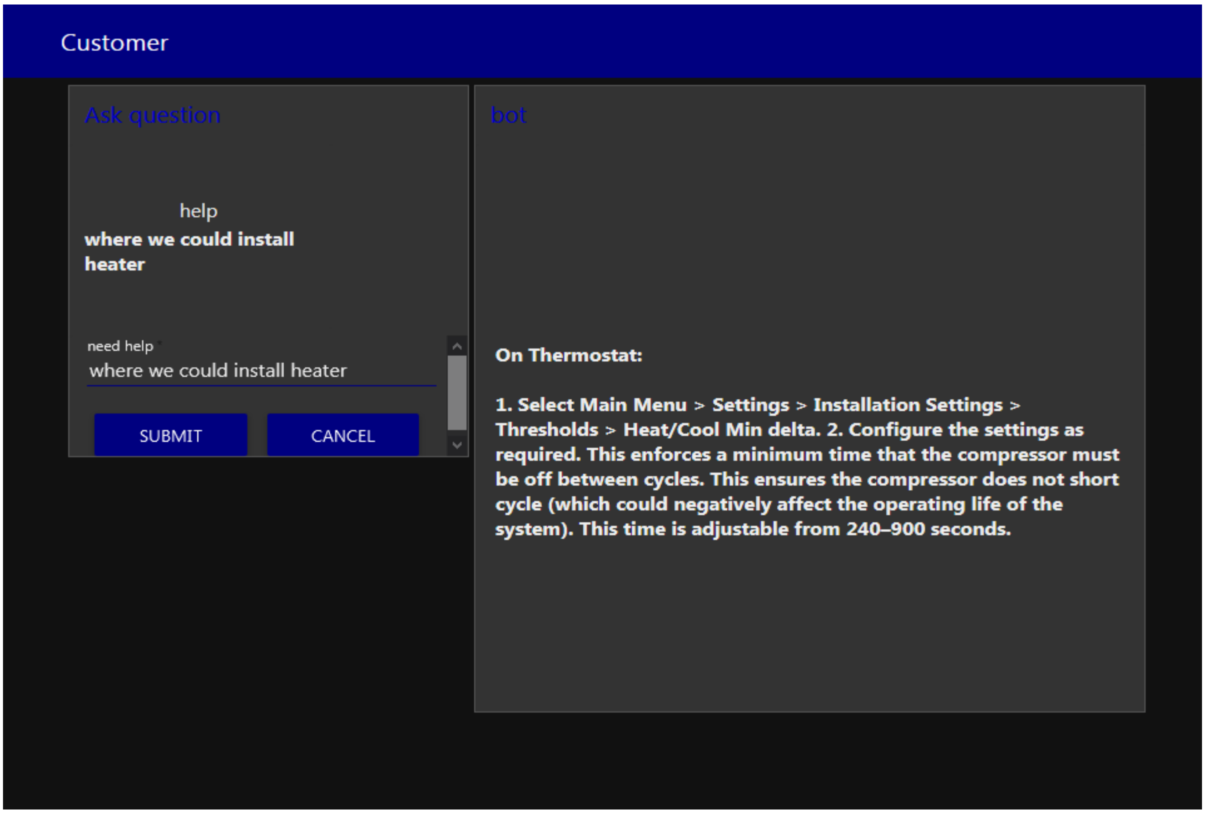
what is gurantee period of heater'

need help

what is gurantee period of heater'

guarantee that interference will not occur in a particular

installation. If this equipment does cause harmful interference to radio or tele- vision reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:



help

where we could install heater

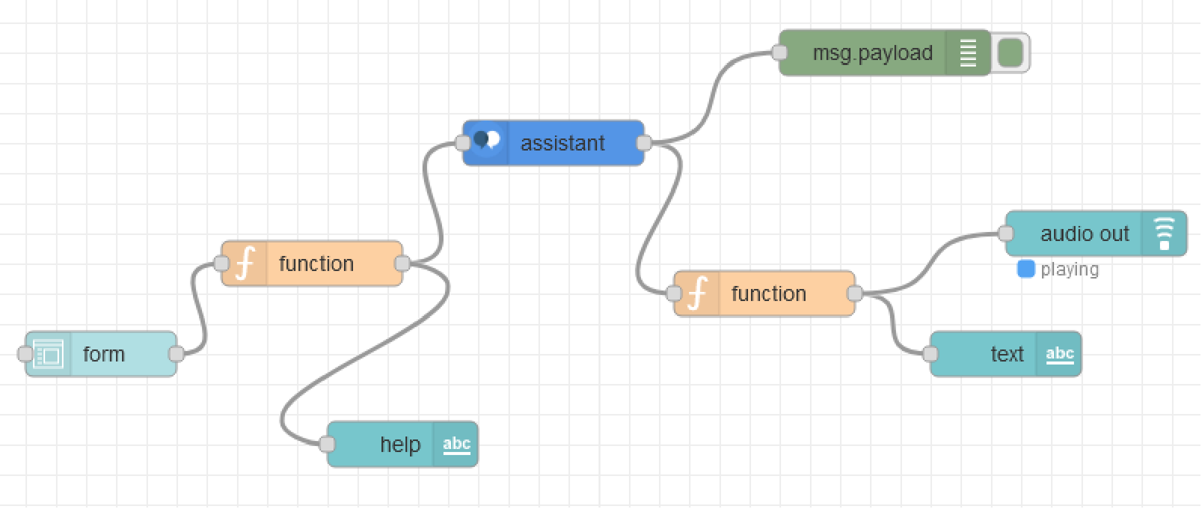
need help

where we could install heater

On Thermostat:

1. Select Main Menu > Settings > Installation Settings > Thresholds > Heat/Cool Min delta. 2. Configure the settings as required. This enforces a minimum time that the compressor must be off between cycles. This ensures the compressor does not short cycle (which could negatively affect the operating life of the system). This time is adjustable from 240–900 seconds.

5. Flowchart



Insert the following nodes into the flow in Node-RED.

- ui_form(1)
- ui_text(2)
- function(2)
- watson-conversation-v1(1)
- debug(1)

6. Results

Web based UI was developed by integrating all the services using Node-RED.
URL for UI Dashboard : https://node-red-fnsst.eu-gb.mybluemix.net/ui

7. Advantages & Disadvantages

7.1 Advantages

1. Reduces Man Power
2. Cost Efficient
3. Few calls will be diverted to Customer Representatives.

7.2 Disadvantages

1. Sometimes it can mislead cutomers as it tries to search irrelevant information in the manual.
2. It may also give same answers to different queries.
3. It may not respond correctly.

8. Applications

- This chatbot can be deployed to various websites as it can solve a lot of basic questions.
- It can be used to deploy as Customer Helpdesk for small scale products as their manual usually has the solution for the user's problems.

9. Conclusion

An Intelligent Customer Helpdesk Chatbot was created using various Watson services like Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-RED.

10. Future Scope

In the future, various other Watson services like Text-To-Speech and Speech-To-Text can be integrated in the chatbot. This can make the chatbot Hands-free.

.

11. Bibliography

1. Project Template:
   https://www.youtube.com/embed/LOCkV-mENq8

2.IBM Cloud:
   https://www.ibm.com/cloud/get-started

3. Node-RED Starter Application :
   https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

4. Create simple webpage
   https://www.w3schools.com/howto/howto_make_a_website.asp

5. Build your own AI assistant :
   https://www.youtube.com/watch?v=hitUOFNne14

6. How to use Watson Assistant with Webhooks :
   https://www.youtube.com/embed/5z3i5IsBVnk

7. Watson Discovery :
   https://developer.ibm.com/articles/introduction-watson-discovery/

Appendix

Source Code

Node RED Flow Code

```
[
    {
        "id": "31bdfe9c.c2dc72",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": ""
    },
    {
        "id": "f2f2649a.0d0d98",
        "type": "debug",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 590,
        "y": 40,
        "wires": []
    },
    {
        "id": "5de83f4d.ba0698",
        "type": "ui_form",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "label": "",
        "group": "d0bc5b73.db0cd",
        "order": 0,
        "width": 0,
        "height": 0,
        "options": [
            {
                "label": "need help",
                "value": "help",
                "type": "text",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "help": ""
        },
        "payload": "",
        "submit": "submit",
        "cancel": "cancel",
        "topic": "",
        "x": 70,
        "y": 240,
        "wires": [
            [
                "547e363.9c296c8"
            ]
        ]
    },
    {
        "id": "547e363.9c296c8",
        "type": "function",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "func": "msg.payload = msg.payload.help;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "x": 210,
        "y": 180,
        "wires": [
            [
                "f00e9dec.55098",
                "cc88e1f.e1e7b2"
            ]
        ]
    },
    {
        "id": "7e7afcb3.2a2e44",
        "type": "function",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "func": "msg.payload.text = \"\";\n\nif(msg.payload.context.webhook_result_1)\n{\n// for(var i in msg.payload.context.webhook_result_1.results)\n// {\n \n msg.payload.text=msg.payload.text+\"<br>\"+msg.payload.context.webhook_result_1.results[0].subtitle;\n msg.payload.text = msg.payload.text+\"<br>\"+\"<br>\"+msg.payload.context.webhook_result_1.results[0].text;\n \n \n // }\n msg.payload = msg.payload.text;\n}\nelse\nmsg.payload = msg.payload.output.text[0];\n\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "x": 510,
        "y": 200,
        "wires": [
            [
                "fab4d636.2848b8",
                "935bf652.0d23f8"
            ]
        ]
    },
    {
        "id": "f00e9dec.55098",
        "type": "watson-conversation-v1",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "workspaceid": "af4f367c-8878-44cb-802b-3b9a1ae8a70b",
        "multiuser": false,
        "context": true,
        "empty-payload": false,
        "service-endpoint": "https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/0baad8b1-d14d-414a-b83f-899e6eab07ec",
        "timeout": "",
        "optout-learning": false,
        "x": 370,
        "y": 100,
        "wires": [
            [
                "7e7afcb3.2a2e44",
                "f2f2649a.0d0d98"
            ]
        ]
    },
    {
        "id": "fab4d636.2848b8",
        "type": "ui_text",
        "z": "31bdfe9c.c2dc72",
        "group": "557e3913.22757",
        "order": 5,
        "width": "10",
        "height": "9",
        "name": "",
        "label": "",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 670,
        "y": 240,
        "wires": []
    },
    {
        "id": "cc88e1f.e1e7b2",
        "type": "ui_text",
        "z": "31bdfe9c.c2dc72",
        "group": "d0bc5b73.db0cd",
        "order": 4,
```

```json
        "width": "4",
        "height": "3",
        "name": "",
        "label": "help",
        "format": "{{msg.payload}}",
        "layout": "col-center",
        "x": 270,
        "y": 300,
        "wires": []
    },
    {
        "id": "935bf652.0d23f8",
        "type": "ui_audio",
        "z": "31bdfe9c.c2dc72",
        "name": "",
        "group": "557e3913.22757",
        "voice": "en-GB",
        "always": "",
        "x": 730,
        "y": 160,
        "wires": []
    },
    {
        "id": "d0bc5b73.db0cd",
        "type": "ui_group",
        "z": "",
        "name": "Ask question",
        "tab": "bad1931d.c1066",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false
    },
    {
        "id": "557e3913.22757",
        "type": "ui_group",
        "z": "",
        "name": "bot",
        "tab": "bad1931d.c1066",
        "order": 1,
        "disp": true,
        "width": "10",
        "collapse": false
    },
    {
        "id": "bad1931d.c1066",
        "type": "ui_tab",
        "z": "",
        "name": "Customer",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    }
]

```

Watson Cloud Function Action Code

```
const assert = require('assert');
const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');

function main(params) {
return new Promise(function (resolve, reject) {
let discovery;

if (params.iam_apikey){
```

```
discovery = new DiscoveryV1({
    'iam_apikey': params.iam_apikey,
    'url': params.url,
    'version': '2019-03-25'
    });
    }
    else {
    discovery = new DiscoveryV1({
    'username': params.username,
    'password': params.password,
    'url': params.url,
    'version': '2019-03-25'
    });
    }
    discovery.query({
    'environment_id': params.environment_id,
    'collection_id': params.collection_id,
    'natural_language_query': params.input,
    'passages': true,
    'count': 3,
    'passages_count': 3
    }, function(err, data) {
    if (err) {
    return reject(err);
    }
    return resolve(data);
    });
    });
    }
```