

Project Report

on

Intelligent Customer Helpdesk with Smart Document Understanding

in

Machine learning / Artificial Intelligence

by

Shane Sam A

(0408briansam0408@gmail.com)

***Intelligent Customer
Help
Desk with Smart
Document
Understanding***

1. Introduction.....	4
1.1 Overview.....	4
1.2 Purpose.....	4
2. Literature Survey.....	5
2.1 Existing Problem.....	5
2.2 Proposed Solution.....	5
3. Theoretical Analysis.....	6
3.1 Block Diagram.....	6
3.2 Hardware / Software Designing.....	6
4. Experimental Investigations.....	7
5. Flowchart.....	9
6. Result.....	10
7. Advantages & Disadvantages.....	11
8. Applications.....	11
9. Conclusion.....	11
10. Future Scope.....	11
11. Bibliography.....	11
12. Appendix.....	12
A. Source Code.....	12

1. Introduction

1.1 Overview

We will build a chatbot that uses various Watson AI Services (Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-Red) to deliver an effective Web based UI through which we can chat with the assistant.

We will integrate the Watson Discovery service with Watson Assistant using webhooks.

1. Project Requirements : Node-RED, IBM Cloud, IBM Watson, Node JS
2. Functional Requirements : IBM Cloud
3. Technical Requirements : AI, ML, Watson AI, Node JS
4. Software Requirements : Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED
5. Project Deliverables : Intelligent Chatbot with Smart Document Understanding
6. Project Team : Shane Sam A
7. Project Duration : 30 Days

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. So unless and until 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.

1.2.1 Scope of Work

1. Create a customer care dialog skill in Watson Assistant
2. Use Smart Document Understanding to build an enhanced Watson Discovery collection
3. Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery
4. Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

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

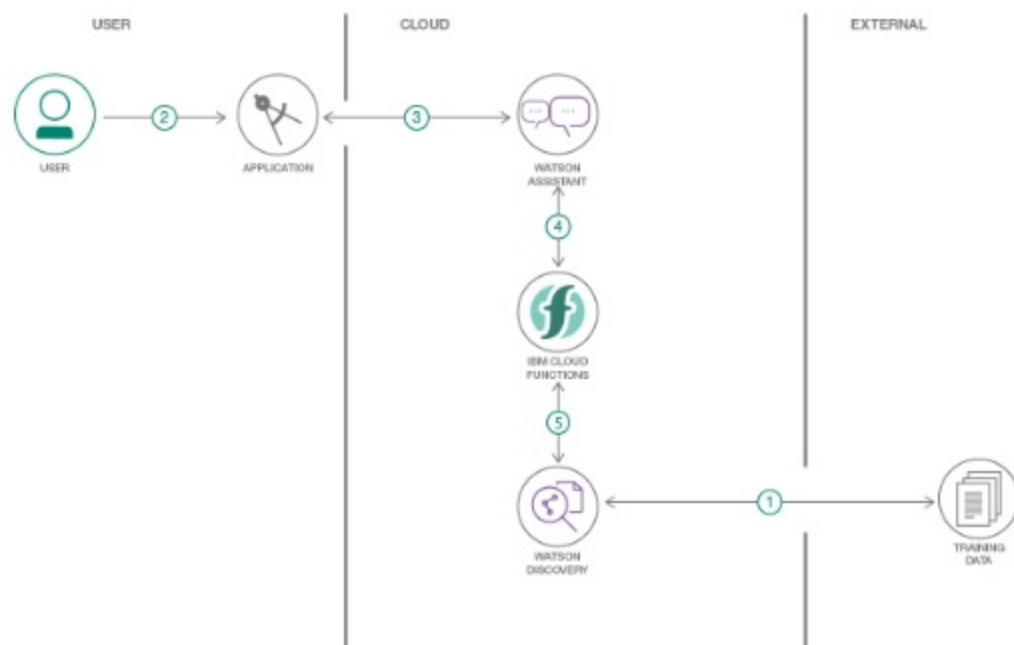
In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now,

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 until 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

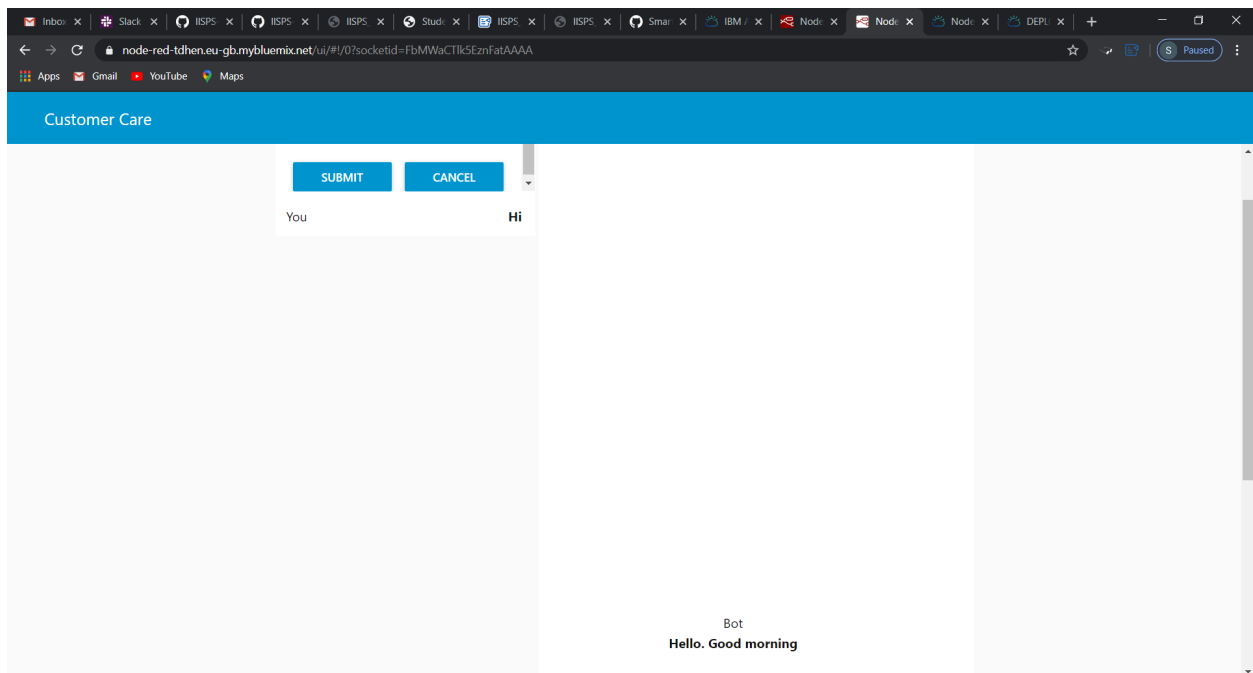


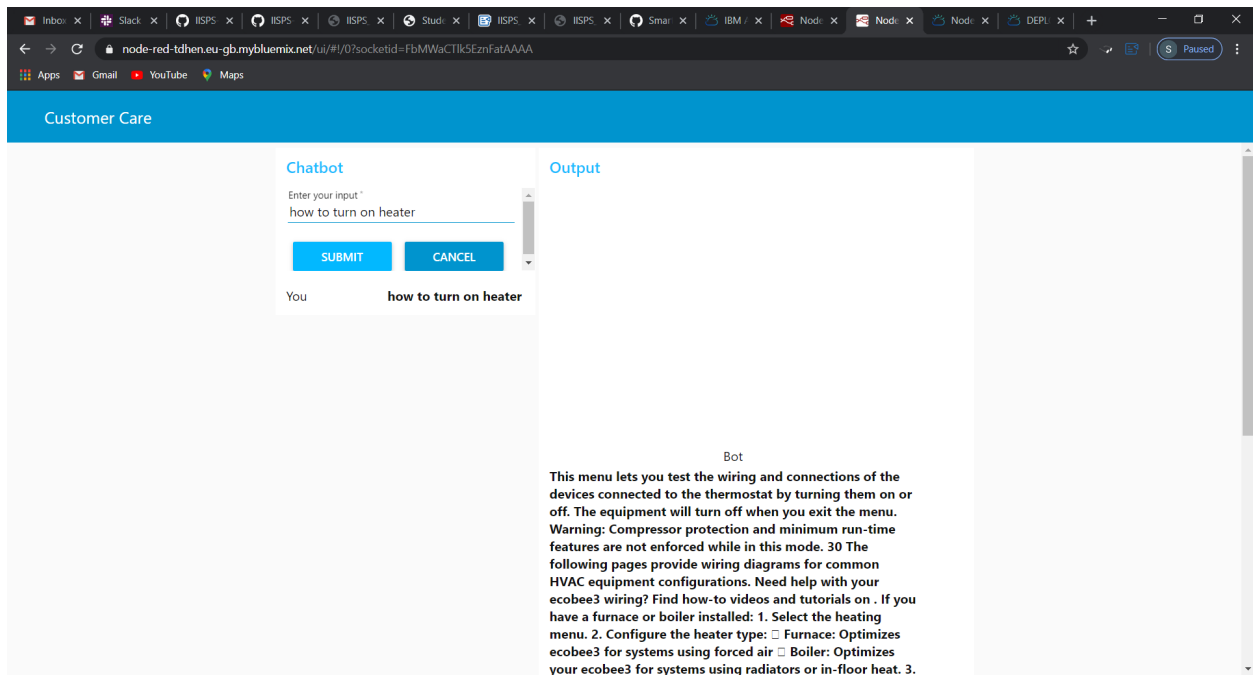
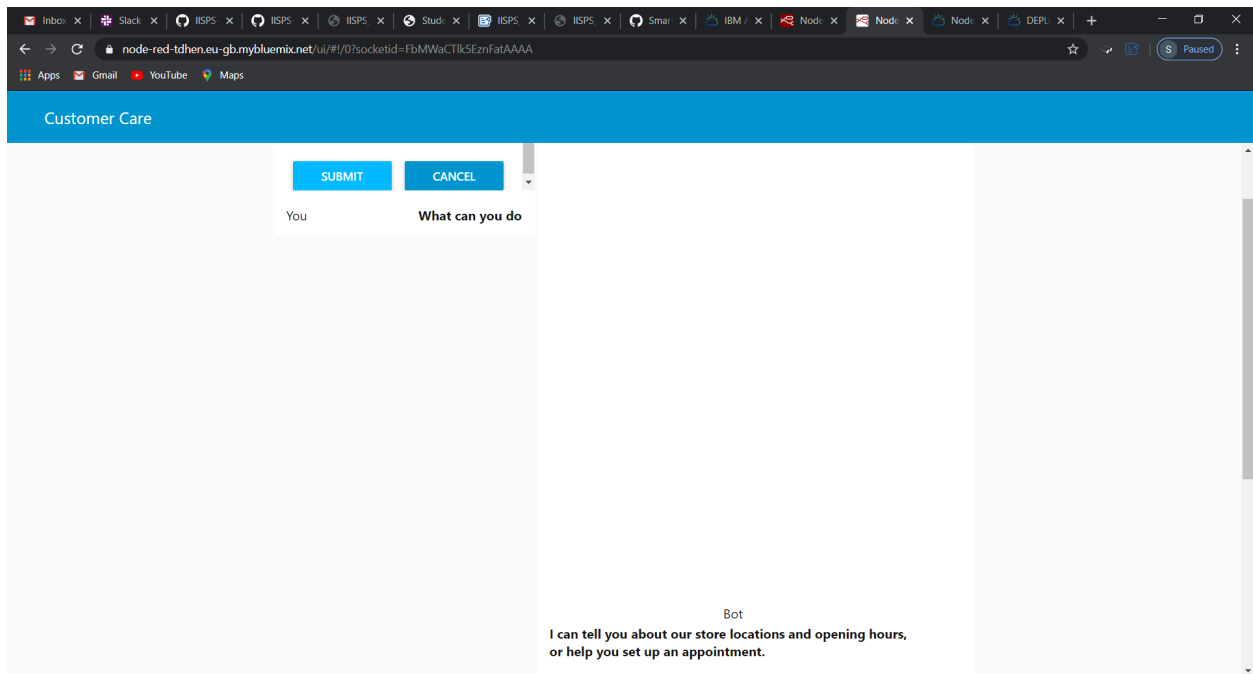
Hardware / Software Designing

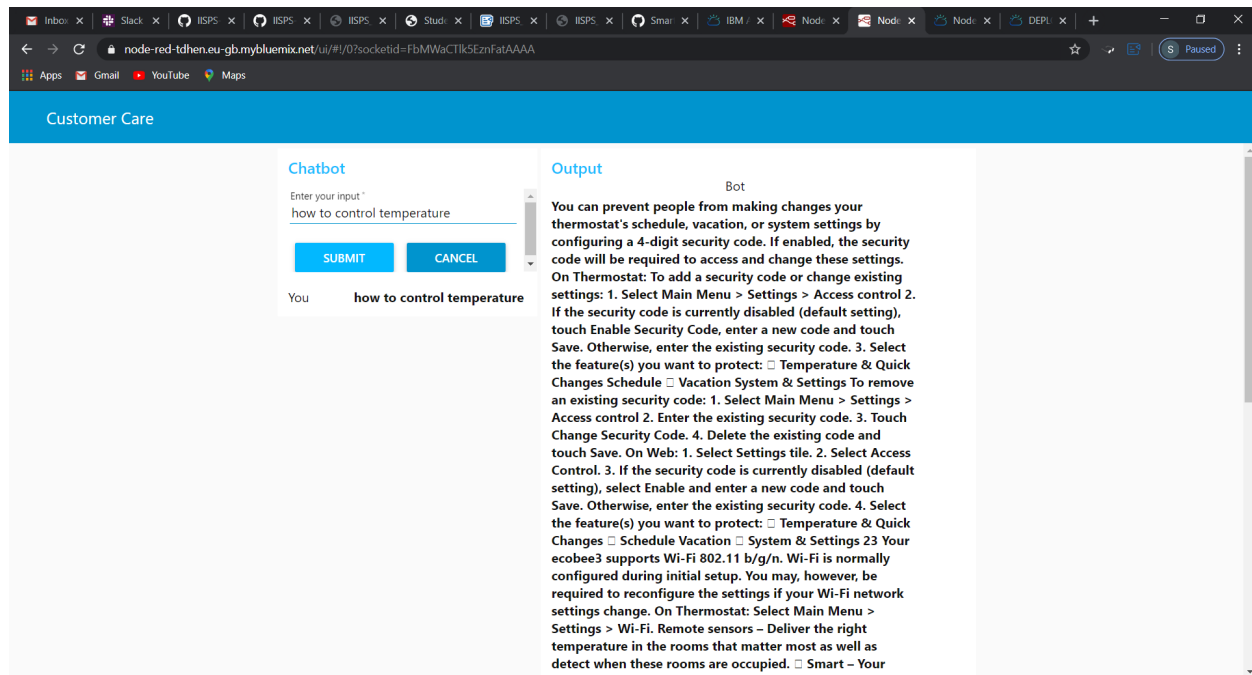
1. Create necessary Watson Services.
2. Configure Watson Discovery.
3. Create Watson Cloud Functions Action.

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



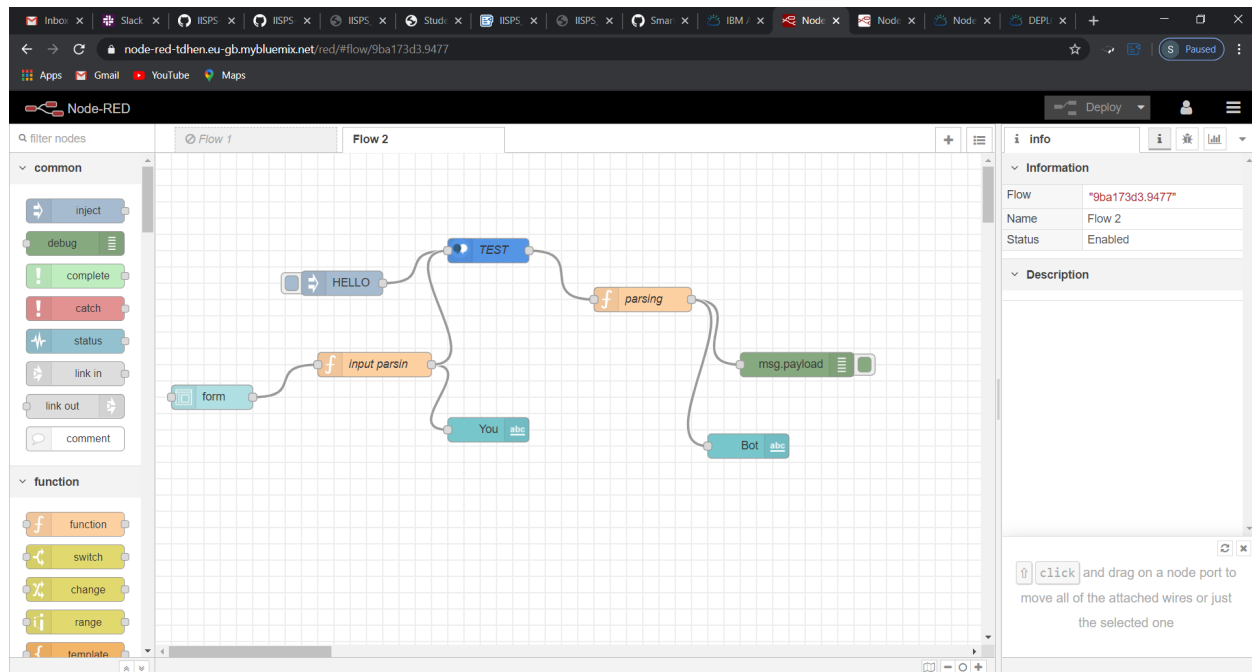




5. Flowchart

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

1. Inject
2. Debug
3. ui_Form
4. ui_Text
5. ui_Button
6. Function
7. Switch
8. Assistant



6. Results

Web based UI was developed by integrating all the services using Node-RED.

URL for UI Dashboard :

<https://node-red-tdhen.eu-gb.mybluemix.net/ui/#!/0?socketid=FbMWaCTIk5EznFatAAA>

7. Advantages & Disadvantages

Advantages

1. Reduces Man Power.
2. Reduces cost.
3. Less calls will be diverted to Customer Representatives.
4. It saves time.
5. Customer representatives will be able to focus on other aspects of their job.

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 needs regular correction and maintenance

8. Applications

1. This chatbot can be deployed in various websites as it can solve a lot of basic questions.
2. It can be deployed in the customer helpdesk of small regular-use products as their manual usually has the required solutions to the user's problems.

9. Conclusion

An Intelligent Customer Helpdesk Chatbot was created successfully 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 and eliminate the need of a human customer representative.

11. Bibliography

1. Node-RED Starter Application :
<https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>
2. Build your own AI assistant : <https://www.youtube.com/watch?v=hitUOFNne14>
3. How to use Watson Assistant with Webhooks :
<https://www.youtube.com/embed/5z3i5lsBVnk>
4. Watson Discovery :
<https://developer.ibm.com/articles/introduction-watson-discovery/>

Appendix

Source Code

Node-RED Flow code

```
[{ "id": "9ba173d3.9477", "type": "tab", "label": "Flow 2", "disabled": false, "info": "" }, { "id": "6aab2799.b8f1d8", "type": "watson-conversation-v1", "z": "9ba173d3.9477", "name": "TEST", "workspaceid": "ddfbbc5e-5cc6-4cc0-b7b7-ce0a1d1ded64", "multiuser": false, "context": false, "empty-payload": false, "service-endpoint": "https://api.au-syd.assistant.watson.cloud.ibm.com/instances/db321622-59d2-4766-b423-c8dad55e2b71", "timeout": "", "optout-learning": false, "x": 410, "y": 120, "wires": [ [ "b9488d37.6f13a" ] ] }, { "id": "b8d3bb4b.327a38", "type": "inject", "z": "9ba173d3.9477", "name": "", "topic": "", "payload": "HELLO", "payloadType": "str", "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "x": 230, "y": 160, "wires": [ [ "6aab2799.b8f1d8" ] ] }, { "id": "24816783.1eb338", "type": "debug", "z": "9ba173d3.9477", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "targetType": "msg", "x": 790, "y": 260, "wires": [] }, { "id": "b9488d37.6f13a", "type": "function", "z": "9ba173d3.9477", "name": "parsing", "func": "msg.payload.text=\"\\\";\\nif(msg.payload.context.webhook_result_1){\\nfor(var i in msg.payload.context.webhook_result_1.results){\\nmsg.payload.text=msg.payload.text+\"\\\"\\n\\\"+msg.payload.context.webhook_result_1.results[i].text;\\n    }\\nmsg.payload=msg.payload.text;\\n}\\nelse\\nmsg.payload=msg.payload.output.text[0];\\nreturn msg;\"", "outputs": 1, "noerr": 0, "x": 600, "y": 180, "wires": [ [ "24816783.1eb338", "99d8731c.a7d66" ] ] }, { "id": "26e3c06b.8f9bc", "type": "ui_form", "z": "9ba173d3.9477", "name": "", "label": "", "group": "a019c64b.3ed708",
```

```

"order": 1, "width": "0", "height": "0", "options": [{ "label": "Enter
your input", "value": "text", "type": "text", "required": true, "rows":
null }], "formValue": { "text": "" }, "payload": "", "submit": "submit",
"cancel": "cancel", "topic": "", "x": 70, "y": 300, "wires": [
    ["52a88724.f959a8"]
  ] }, { "id": "52a88724.f959a8", "type": "function", "z":
"9ba173d3.9477", "name": "input parsin", "func":
"msg.payload=msg.payload.text;\nreturn msg;", "outputs": 1, "noerr": 0,
"x": 270, "y": 260, "wires": [
    ["a718eac7.995fd8", "6aab2799.b8f1d8"]
  ] }, { "id": "99d8731c.a7d66", "type": "ui_text", "z":
"9ba173d3.9477", "group": "65126666.23dec8", "order": 2, "width": "9",
"height": "18", "name": "", "label": "Bot", "format": "{{msg.payload}}",
"layout": "col-center", "x": 730, "y": 360, "wires": [] }, { "id":
"a718eac7.995fd8", "type": "ui_text", "z": "9ba173d3.9477", "group":
"a019c64b.3ed708", "order": 2, "width": "0", "height": "0", "name": "",
"label": "You", "format": "{{msg.payload}}", "layout": "row-spread", "x":
410, "y": 340, "wires": [] }, { "id": "a019c64b.3ed708", "type":
"ui_group", "z": "", "name": "Chatbot", "tab": "2fb12e7e.9ecb62", "order":
1, "disp": true, "width": "6", "collapse": false }, { "id":
"65126666.23dec8", "type": "ui_group", "z": "", "name": "Output", "tab":
"2fb12e7e.9ecb62", "order": 2, "disp": true, "width": "10", "collapse":
false }, { "id": "2fb12e7e.9ecb62", "type": "ui_tab", "z": "", "name":
"Customer Care", "icon": "dashboard", "disabled": false, "hidden": false
}]

```

Watson Cloud Function Action Code

```

/**
 *
 * @param {object} params
 * @param {string} params.iam_apikey
 * @param {string} params.url
 * @param {string} params.username
 * @param {string} params.password

```

```

* @param {string} params.environment_id
* @param {string} params.collection_id
* @param {string} params.configuration_id
* @param {string} params.input
*
* @return {object}
*
*/

```

```

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

```

```

/**
 *
 * main() will be run when you invoke this action
 *
 * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
 *
 * @return The output of this action, which must be a JSON object.
 *
*/

```

```

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);  
});  
});  
}
```