

PROJECT REPORT

Intelligent Customer Help Desk with Smart Document Understanding

**BY
NIKHIL KAUSHAL**

1. PROJECT SCOPE:

1.1 PROJECT OBJECTIVE

To build a Intelligent Customer Help Desk With Smart Document Understanding.

1.2 DELIVERABLES

- The typical customer care chatbot that 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.
- If the customer question is about the operation of a device, so it will return relevant sections of the owners manual to help solve our customers' problems.

1.3 MILESTONES

- Creating a customer care dialog skill in Watson Assistant.
- Using Smart Document Understanding to build an enhanced Watson Discovery collection.
- Creating an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery.
- Building a web application with integration to all these services & deploy the same on IBM Cloud Platform.

1.4 PROJECT REQUIREMENTS

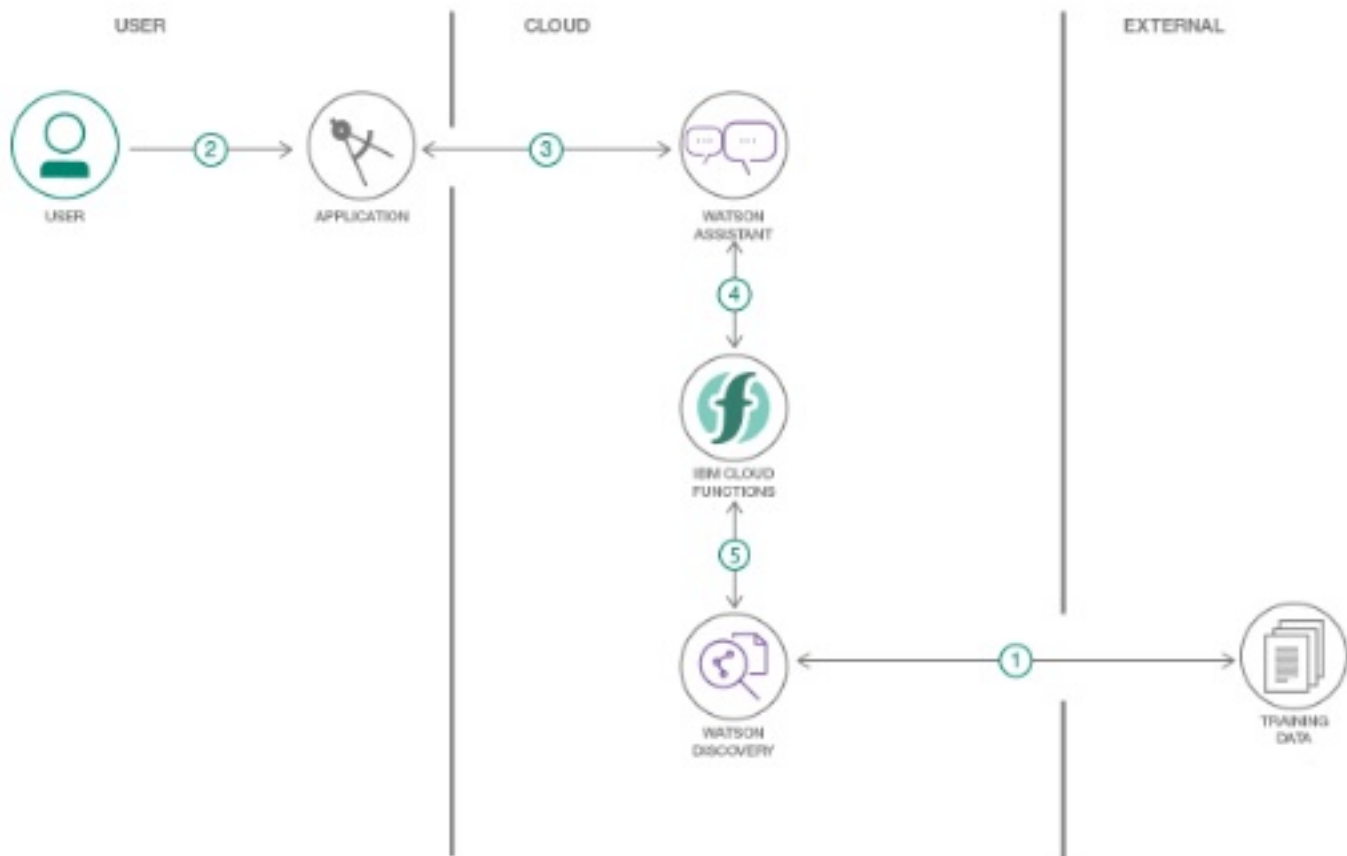
- IBM Cloud platform
- IBM Watson services
- IBM Cloud functions
- Node-RED web application for integrating all services.

2. Summary:

A typical customer care chatbot that can answer simple questions, such as store locations and hours, directions, and 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 passes 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?" it will return relevant sections of the owners manual to help solve our customers' problems.

To take it a step further, the project also uses 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 improved the answers returned from the queries.



3. Experimental Investigations:

3.1. Creating IBM Cloud services

Create the following services:

- Watson Assistant
- Watson Discovery
- IBM cloud function
- Node Red

3.2. Creating Node-Red application on IBM Cloud

1. Go to IBM cloud and in catalogs, search for Node-Red.
2. Now, give a unique name to your application and configure the application.
3. As the application will be configured, you can go to Application URL.
4. Then a Node-Red wizard will open and ask if you want a password for it or not. You can choose according to your will.
5. Once the workspace will open you are all set for Node-red flow.

The screenshot displays the IBM Cloud Developer console interface. The browser address bar shows the URL: <https://cloud.ibm.com/developer/appservice/apps/31e0ab44-b132-42e1-969b-db21d792fd4a>. The page title is "Customer Desk ChatBot" with an "Add tags" link and an "Actions..." dropdown menu.

Details

App URL	https://customer-desk-chatbot.eu-gb.mybluemix.net
Source	https://eu-gb.git.cloud.ibm.com/nikhilkaushal0000/CustomerDeskChatBot
Resource group	Default
Deployment target	Customer Desk ChatBot
Created	5/25/2020

Services

Cloudant

[Open dashboard](#) [Documentation](#)

Credentials ▾

[Connect existing services](#) [Create service](#)

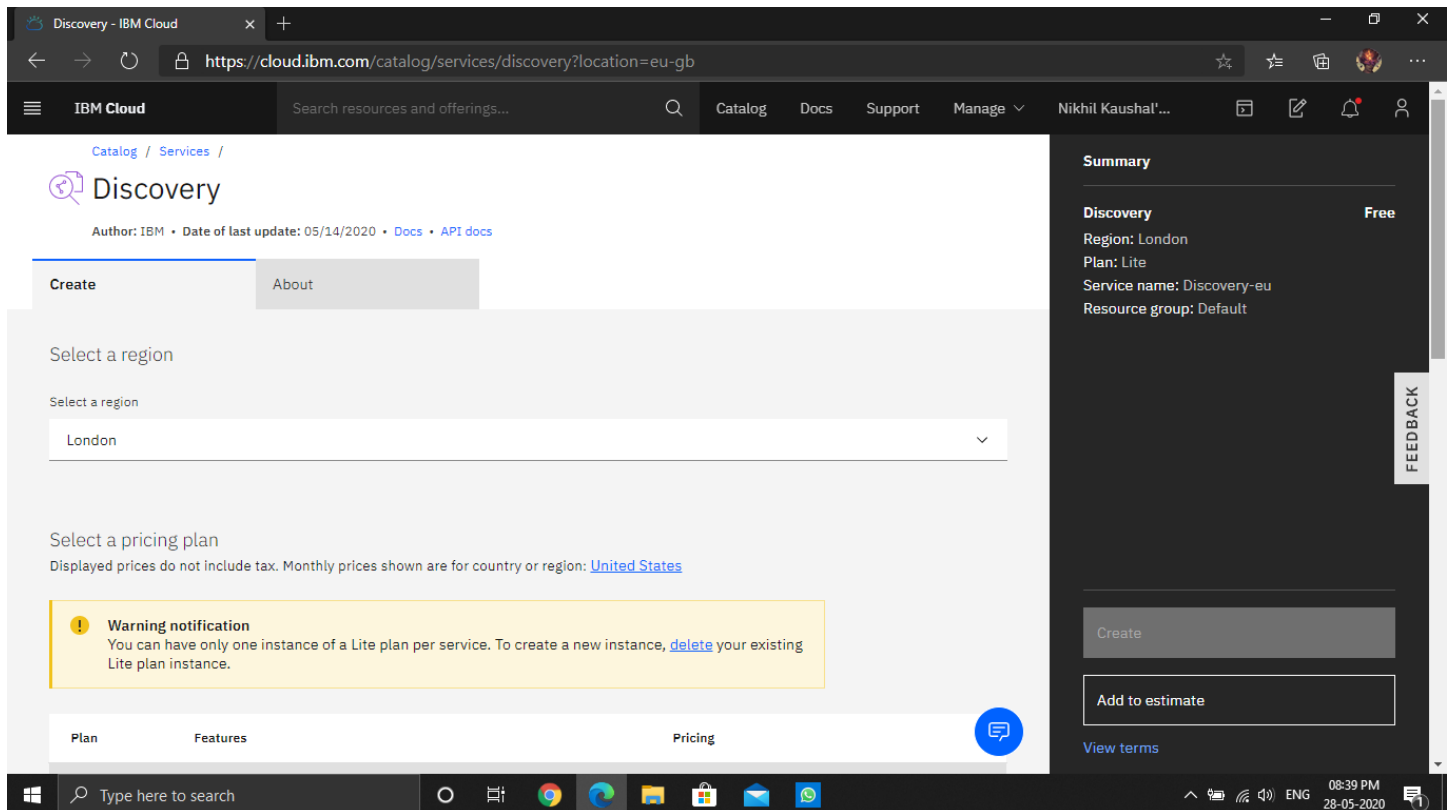
Deployment Automation

Name	CustomerDeskChatBot
Location	London
Tool integrations	
Delivery Pipelines	
Name	CustomerDeskChatBot
Status	Success
Last input	Last commit by IBM Cloud (3 days ago) Clone from zip

On the right side of the console, there are vertical buttons labeled "FEEDBACK" and "ASK A QUESTION". The Windows taskbar at the bottom shows the time as 08:25 PM on 28-05-2020.

3.3. Creation of Watson discovery instance in IBM Cloud

- In Watson Discovery you can add as many documents you want in order to get the specified result out of the documents.
- Also we can configure the data in the documents in order to get the better results and to improve the working of Watson Discovery.
- We can annoate the data in configuration as we will configure the sections page by page, the watson discovery will improve itself and there will be a time when you don't need to configure anymore because Watson discovery as already improved itself.
- We also need to split the documents by section in order to get better results for the queries.
- In upcoming steps, we will need to provide the credentials to access your Discovery collection. The values can be found in the following locations. The Collection ID and Environment ID values can be found by clicking the dropdown button located at the top right side of your collection panel.



3.4. Creating IBM cloud functions

1. Create the web action that will make queries to our Discovery collection. Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter, then select the Functions card.
2. 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.
3. Once your action is created, click on the Code tab.
4. In the code editor window, type the code - that will simply connects to the Discovery service, makes a query against the collection, then returns the response.
5. Add the following keys:
 - url
 - environment_id
 - collection_id
 - iam_apikey
6. Now that the credentials are set, return to the Code panel and press the Invoke button.
7. Then, go to Endpoints panel. Click the checkbox for Enable as Web Action. This will generate a public endpoint URL.

The screenshot shows the IBM Cloud Functions console interface. The browser address bar displays the URL: https://cloud.ibm.com/functions/details/action/nikhilkaushal0000%2540gmail.com_dev/discovery-functions/code. The page title is "discovery-functions" and it is identified as a "Web Action". The namespace is "nikhilkaushal0000@gmail.com_dev(London)".

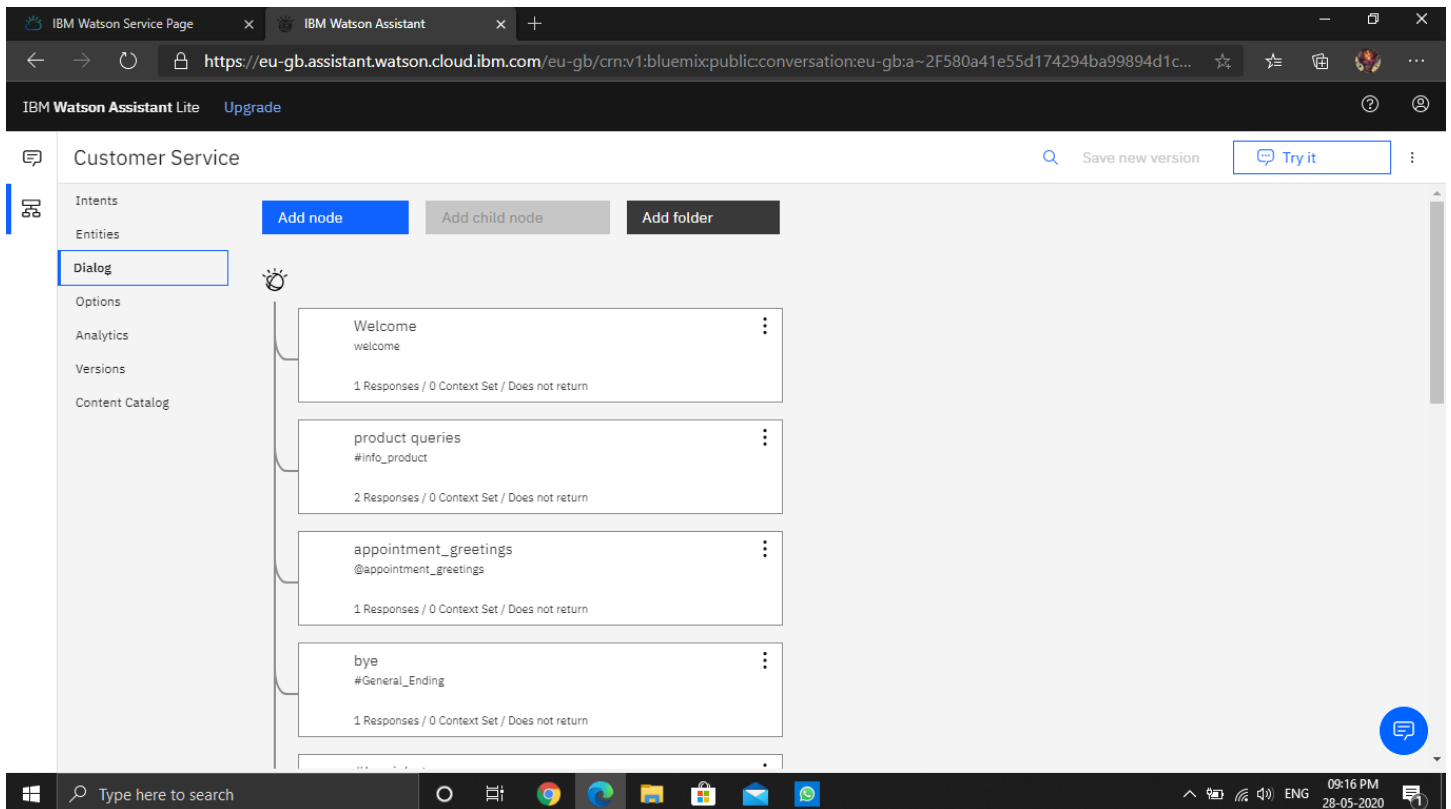
On the left sidebar, the "Code" tab is selected. Below it are links for Parameters, Runtime, Endpoints, Connected Triggers, Enclosing Sequences, and Logs.

The main area shows the code editor for "Node.js 10". The code is as follows:

```
1  /**
2   *
3   * @param {object} params
4   * @param {string} params.iam_apikey
5   * @param {string} params.url
6   * @param {string} params.username
7   * @param {string} params.password
8   * @param {string} params.environment_id
9   * @param {string} params.collection_id
10  * @param {string} params.configuration_id
11  * @param {string} params.input
12  *
13  * @return {object}
14  *
15  */
16
17  const assert = require('assert');
18  const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');
19
20  /**
21   *
22   * main() will be run when you invoke this action
23   *
24   * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
25   *
26   * @return The output of this action, which must be a JSON object.
27   *
28   */
29  function main(params) {
30    return new Promise(function (resolve, reject) {
```

3.5. Configuring Watson Assistant

1. Launch the Watson Assistant tool and create a new dialog skill. Select the sample skills for beginning. This dialog skill contains almost all the typical questions that are being asked.
2. You can create the Intent so that if a particular phrase is being asked by the customer. Then a particular reply is being given associated with that.
3. We also need to add a dialog node for the actions that must be performed. We can add as much dialog nodes we can in order to improve our chatbot. We also have an option to jump over to any other node.



4. Enable webhook for the IBM Cloud Functions action you created. WebHooks are important to get the result from the Watson Discovery incase the user result for the query related to operation of the device.

IBM Watson Assistant

https://eu-gb.assistant.watson.cloud.ibm.com/eu-gb/crn:v1:bluemix:public:conversation:eu-gb:a~2F580a41e55d174294ba99894d1c...

IBM Watson Assistant Lite Upgrade

Assistants

Skills

product queries

Node name will be shown to customers for disambiguation so use something descriptive.

Key	Value
input	"<?input.text?>"

Add parameter +

Return variable

webhook_result_1

Webhook URL Your webhook URL is configured. Options

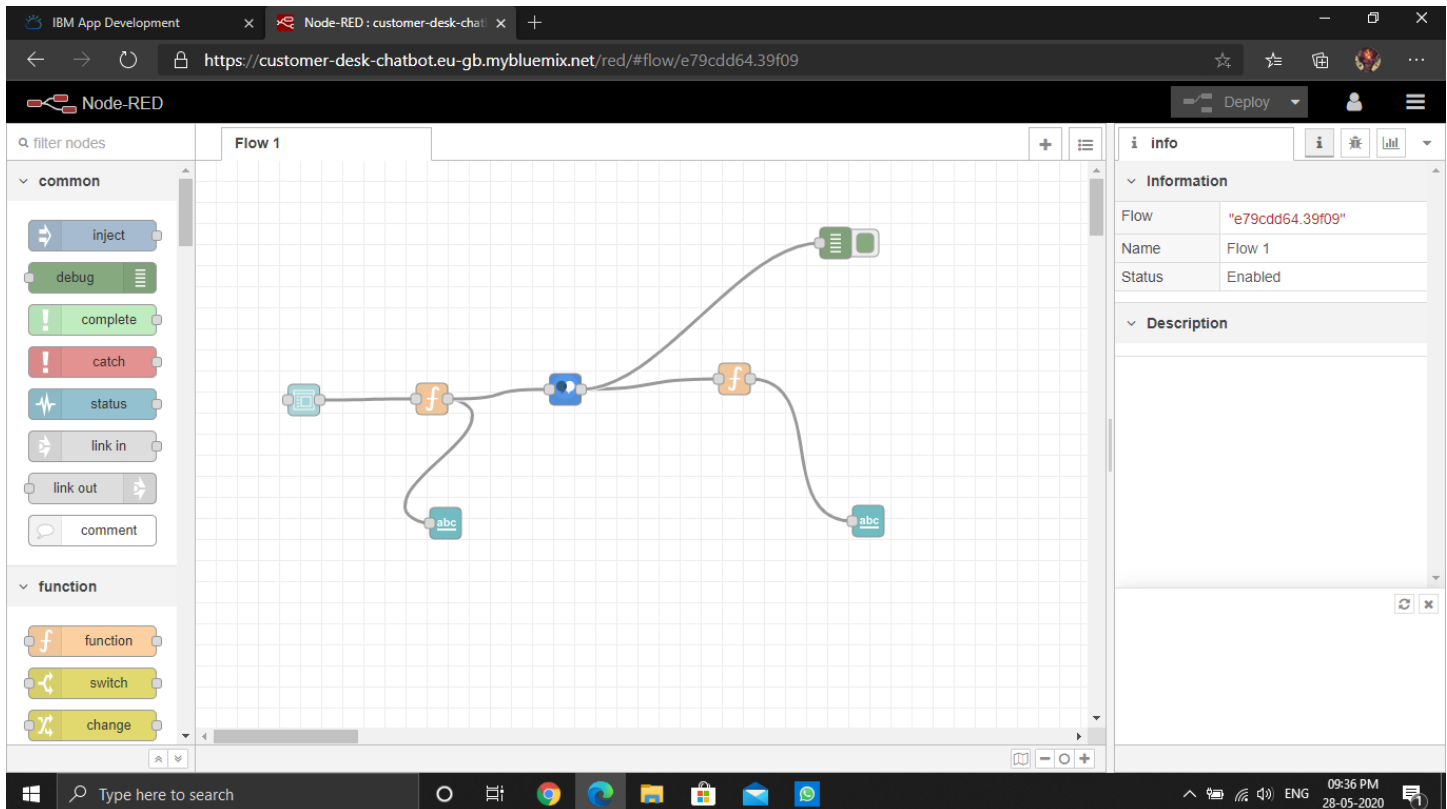
Assistant responds

Type here to search

09:22 PM 28-05-2020

3.6. Integration of Watson Assistant in Node-RED

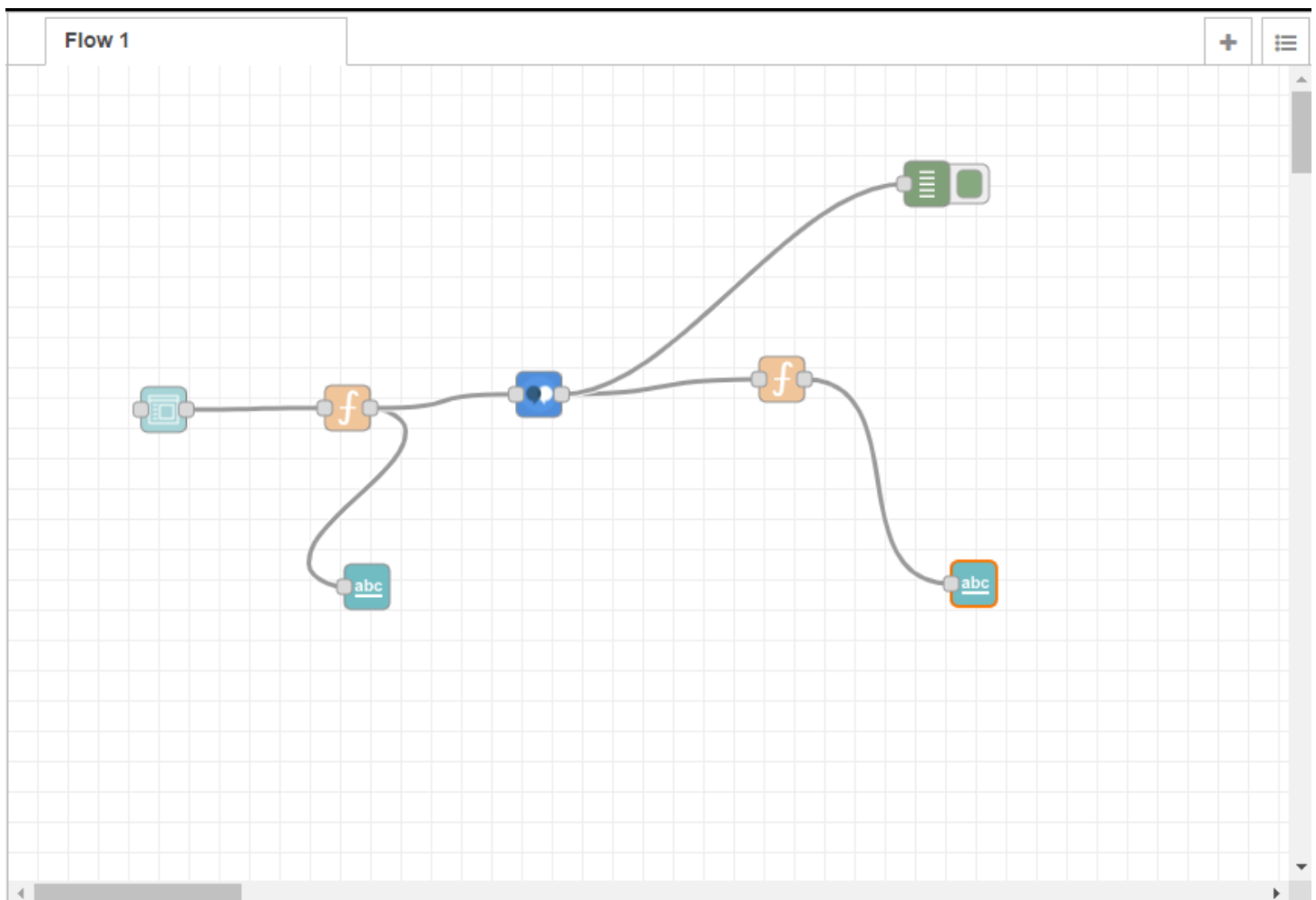
1. Double-click on the Watson Assistant node.
2. Give a name to your node and enter the username, password and workspace ID of your Watson Assistant service .
3. After entering all the information click on Done.
4. Drag Inject node on to the flow from the Input section.
5. Drag Debug node on to the flow from the Output section.
6. Double-click on the Inject node.
7. Select the Payload as a string.
8. Enter a sample input to be sent to the assistant service and click on Done .
9. Connect the nodes as shown below and click on Deploy.
10. Open Debug window as shown below.
11. Click on the button to send input text to the assistant node.
12. Observe the output from the assistant service node.
13. Drag the function node to parse the JSON data and get the bot response.
14. Double click on the function node and enter the JSON parsing code as shown below and click on done.
15. Connect the nodes as shown below, and click on Deploy.



FLOWCHART:

Create the flow using the following nodes :-

- function
- assistant
- debug
- form
- Ui_form
- Ui_text

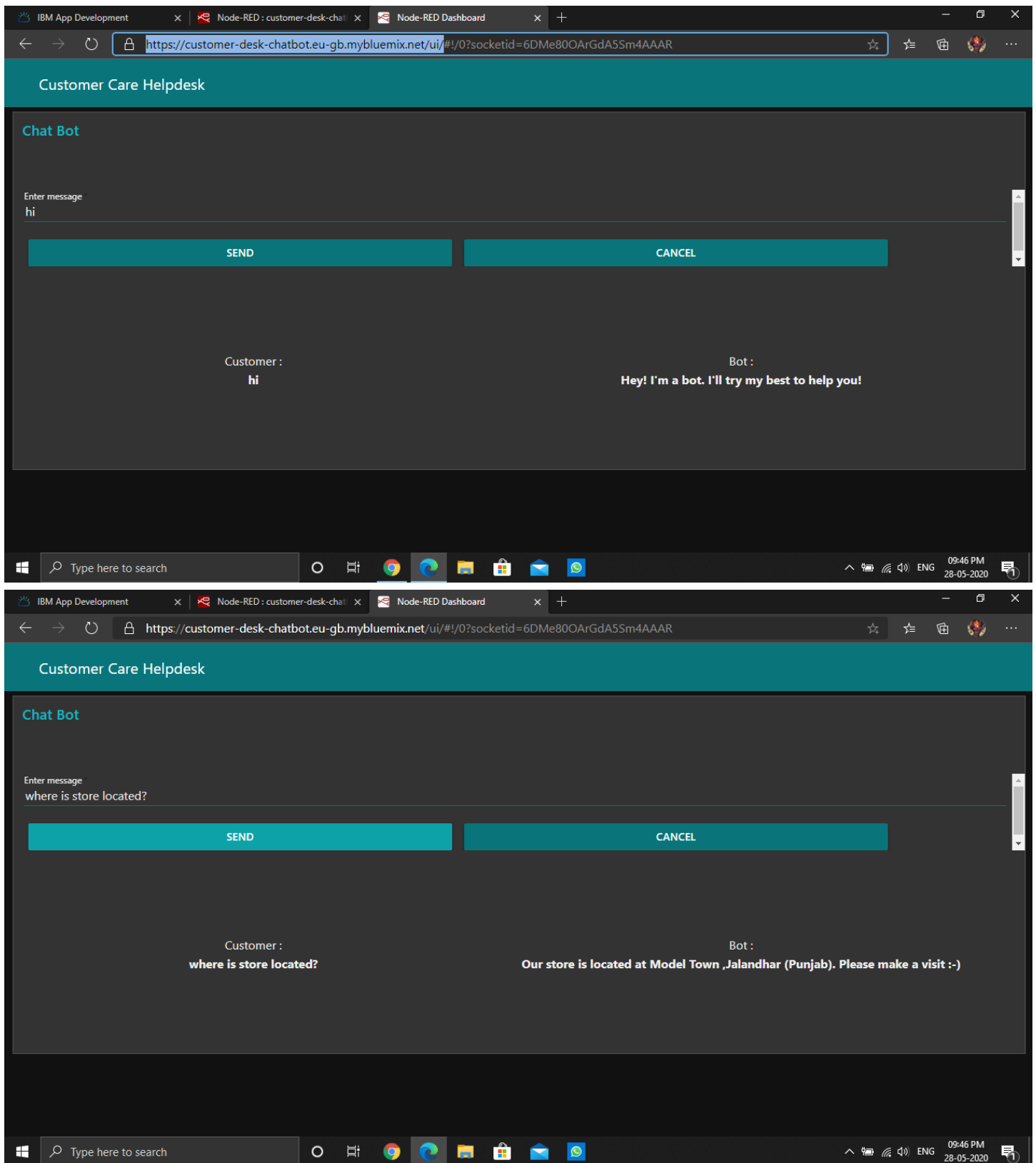


4. RESULT:

The Chatbot can be accessed by going to following URL:

<https://customer-desk-chatbot.eu-gb.mybluemix.net/ui/>

5. SCREENSHOTS:



IBM App Development x Node-RED : customer-desk-chat x Node-RED Dashboard x +

← → ↻ 🔒 https://customer-desk-chatbot.eu-gb.mybluemix.net/ui/#!/0?socketid=6DMe800ArGdA5Sm4AAAR ☆ ☆= 📄 🌐 ⋮

Customer Care Helpdesk

Chat Bot

Enter message
I want to talk to a real person

SEND CANCEL

Customer :
I want to talk to a real person

Bot :
A real person will reach out to you ! You'll soon receive a call.

Windows Taskbar: Type here to search, Task View, Chrome, Edge, File Explorer, Microsoft Store, Mail, Messages, System Tray: Network, Volume, ENG, 09:47 PM, 28-05-2020, Notifications (1)

IBM App Development x Node-RED : customer-desk-chat x Node-RED Dashboard x +

← → ↻ 🔒 https://customer-desk-chatbot.eu-gb.mybluemix.net/ui/#!/0?socketid=6DMe800ArGdA5Sm4AAAR ☆ ☆= 📄 🌐 ⋮

Customer Care Helpdesk

Chat Bot

Enter message
i want to book an appointment

SEND CANCEL

Customer :
i want to book an appointment

Bot :
We will schedule an appointment.Please let us know the timings for the appointment.

Windows Taskbar: Type here to search, Task View, Chrome, Edge, File Explorer, Microsoft Store, Mail, Messages, System Tray: Network, Volume, ENG, 09:48 PM, 28-05-2020, Notifications (1)

Customer Care Helpdesk

Chat Bot

Enter message
how do i access the settings

SEND CANCEL

Customer :
how do i access the settings

Bot :
"If enabled, the security code will be required to access and change these settings. On Thermostat: To add a security code or change existing settings: 1. Select Main Menu > Settings > Access control 2. If the security code is currently disabled (default setting), touch Enable Security Code, enter a new code and touch Save."

Windows taskbar: Type here to search, icons for Edge, File Explorer, Mail, etc. System tray: 09:52 PM, 28-05-2020.

Customer Care Helpdesk

Chat Bot

Enter message
how do i set the time

SEND CANCEL

Customer :
how do i set the time

Bot :
"Sets the minimum equipment run time in cool mode: 1 to 20 min (default is 5 min). On Thermostat: 1. Select Main Menu > Settings > Installation Settings > Thresholds > Cool Min On Time. 2. Configure the settings as required."

Windows taskbar: Type here to search, icons for Edge, File Explorer, Mail, etc. System tray: 09:58 PM, 28-05-2020.

6. ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1. It can give the results about device operations in much efficient way.
2. It is easy to build because it only use predefined technologies.
3. No need of real human to facilitate the customers for simple tasks.

DISADVANTAGES:

1. Same answers to many questions.
2. Require much time for training Watson Assistant and Watson Discovery.
3. May mislead customers.

7. APPLICATIONS:

- Can be deployed to social media like Slack and Telegram.
- Can be used on personal websites.

8. CONCLUSION:

A person will be able to build a chatbot with a smart document understanding if he/she follows the above procedure carefully.

9. FUTURE SCOPE:

This chatbot can be improved by training it to a futhur extend. A person can add further functionalities to it and can also improve Watson Discovery by adding more documents.

10. BIBLIOGRAPHY:

APPENDIX:

Node-RED Flow Source Code

```
[
  {
    "id": "e79cdd64.39f09",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false,
    "info": ""
  },
  {
    "id": "680f88fd.178b28",
    "type": "ui_tab",
    "z": "",
    "name": "Customer Care Helpdesk",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
  },
  {
    "id": "63aa13e8.9523dc",
    "type": "ui_group",
    "z": "",
    "name": "Chat Bot",
    "tab": "680f88fd.178b28",
    "order": 1,
    "disp": true,
    "width": 25,
    "collapse": false
  },
  {
    "id": "3e639c3.2fd1964",
    "type": "ui_base",
    "theme": {
      "name": "theme-dark",
```



```
"lightTheme": {
  "default": "#0094CE",
  "baseColor": "#0094CE",
  "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
  "edited": true,
  "reset": false
},
"darkTheme": {
  "default": "#097479",
  "baseColor": "#097479",
  "baseFont": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
  "edited": true,
  "reset": false
},
"customTheme": {
  "name": "Untitled Theme 1",
  "default": "#4B7930",
  "baseColor": "#4B7930",
  "baseFont": "Century Gothic,CenturyGothic,AppleGothic,sans-serif",
  "reset": false
},
"themeState": {
  "base-color": {
    "default": "#097479",
    "value": "#097479",
    "edited": false
  },
  "page-titlebar-backgroundColor": {
    "value": "#097479",
    "edited": true
  },
  "page-backgroundColor": {
    "value": "#111111",
    "edited": false
  },
  "page-sidebar-backgroundColor": {
    "value": "#000000",
    "edited": false
  }
}
```

```

    },
    "group-textColor": {
      "value": "#0eb8c0",
      "edited": false
    },
    },
    "group-borderColor": {
      "value": "#555555",
      "edited": false
    },
    },
    "group-backgroundColor": {
      "value": "#333333",
      "edited": false
    },
    },
    "widget-textColor": {
      "value": "#eeeeee",
      "edited": false
    },
    },
    "widget-backgroundColor": {
      "value": "#097479",
      "edited": false
    },
    },
    "widget-borderColor": {
      "value": "#333333",
      "edited": false
    },
    },
    "base-font": {
      "value": "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
    }
  },
  "angularTheme": {
    "primary": "indigo",
    "accents": "blue",
    "warn": "red",
    "background": "grey"
  }
},
"site": {
  "name": "Node-RED Dashboard",
  "hideToolbar": "false",

```

```
        "allowSwipe": "false",
        "lockMenu": "false",
        "allowTempTheme": "true",
        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 0,
            "py": 0
        }
    },
    {
        "id": "e4856786.955b38",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "63aa13e8.9523dc",
        "order": 1,
        "width": 3,
        "height": 1
    },
    {
        "id": "f6f34f1d.8849",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "63aa13e8.9523dc",
        "order": 3,
        "width": 3,
        "height": 1
    },
    {
        "id": "a44c816c.d0f56",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "63aa13e8.9523dc",
```

```
    "order": 4,
    "width": 3,
    "height": 1
  },
  {
    "id": "55418810.a79c88",
    "type": "ui_spacer",
    "name": "spacer",
    "group": "63aa13e8.9523dc",
    "order": 5,
    "width": 3,
    "height": 1
  },
  {
    "id": "79067310.9cff4c",
    "type": "ui_spacer",
    "name": "spacer",
    "group": "63aa13e8.9523dc",
    "order": 7,
    "width": 1,
    "height": 1
  },
  {
    "id": "33e20640.17f42a",
    "type": "ui_spacer",
    "name": "spacer",
    "group": "63aa13e8.9523dc",
    "order": 9,
    "width": 1,
    "height": 1
  },
  {
    "id": "cec9ad08.8d31",
    "type": "ui_spacer",
    "name": "spacer",
    "group": "63aa13e8.9523dc",
    "order": 10,
    "width": 1,
    "height": 1
  }
```

```
},
{
  "id": "ee7b9226.b213d",
  "type": "ui_spacer",
  "name": "spacer",
  "group": "63aa13e8.9523dc",
  "order": 11,
  "width": 1,
  "height": 1
},
{
  "id": "5db8b4fa.b77eac",
  "type": "ui_spacer",
  "name": "spacer",
  "group": "63aa13e8.9523dc",
  "order": 12,
  "width": 1,
  "height": 1
},
{
  "id": "a5467bd8.b8bfc8",
  "type": "ui_form",
  "z": "e79cdd64.39f09",
  "name": "",
  "label": "",
  "group": "63aa13e8.9523dc",
  "order": 2,
  "width": "0",
  "height": "0",
  "options": [
    {
      "label": "Enter message",
      "value": "text",
      "type": "text",
      "required": true,
      "rows": null
    }
  ],
  "formValue": {
```

```

        "text": ""
    },
    "payload": "",
    "submit": "Send",
    "cancel": "Cancel",
    "topic": "",
    "x": 102.5,
    "y": 227,
    "wires": [
        [
            "e3164910.0ec338"
        ]
    ],
    "l": false
},
{
    "id": "e3164910.0ec338",
    "type": "function",
    "z": "e79cdd64.39f09",
    "name": "",
    "func": "msg.payload=msg.payload.text;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 224,
    "y": 226,
    "wires": [
        [
            "6b1355a8.0dfadc",
            "58cca53c.842f4c"
        ]
    ],
    "l": false
},
{
    "id": "bc053695.b7d668",
    "type": "function",
    "z": "e79cdd64.39f09",
    "name": "",
    "func": "msg.payload=msg.payload.output.text[0];\nreturn msg;",

```

```
    "outputs": 1,
    "noerr": 0,
    "x": 511,
    "y": 207,
    "wires": [
      [
        "d3906887.17ec98"
      ]
    ],
    "l": false
  },
  {
    "id": "6b1355a8.0dfadc",
    "type": "ui_text",
    "z": "e79cdd64.39f09",
    "group": "63aa13e8.9523dc",
    "order": 6,
    "width": 12,
    "height": 5,
    "name": "",
    "label": "Customer :",
    "format": "{{msg.payload}}",
    "layout": "col-center",
    "x": 237,
    "y": 344,
    "wires": [],
    "l": false
  },
  {
    "id": "72b74bfd.8fcdc4",
    "type": "debug",
    "z": "e79cdd64.39f09",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
```

```

        "x": 607,
        "y": 78,
        "wires": [],
        "l": false
    },
    {
        "id": "d3906887.17ec98",
        "type": "ui_text",
        "z": "e79cdd64.39f09",
        "group": "63aa13e8.9523dc",
        "order": 8,
        "width": 12,
        "height": 5,
        "name": "",
        "label": "Bot :",
        "format": "{{msg.payload}}",
        "layout": "col-center",
        "x": 638,
        "y": 342,
        "wires": [],
        "l": false
    },
    {
        "id": "58cca53c.842f4c",
        "type": "watson-conversation-v1",
        "z": "e79cdd64.39f09",
        "name": "",
        "workspaceid": "9b1f8555-d16c-443e-9581-f00a4a78123a",
        "multiuser": false,
        "context": true,
        "empty-payload": false,
        "service-endpoint":
"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/b084f97f-04fc-4913-a247-1f7c9f7c42a6",
        "timeout": "",
        "optout-learning": false,
        "x": 350.5,
        "y": 217,
        "wires": [
            [

```



```

        "bc053695.b7d668",
        "72b74bfd.8fcdc4"
    ]
  ],
  "l": false
}
]

```

Cloud Function Source 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');

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);
    });
});
}

```