# PROJECT REPORT

**NAME** **:** Anjali Verma

**EMAIL ID** **:** vermaanjali2000@gmail.com

**GITHUB ID** **:** anjaliverma-09

**PROJECT ID** **:** SPS_PRO_99

**PROJECT TITLE** **:** Intelligent Customer Help Desk with Smart Document Understanding

**INTERNSHIP AT** **:** smartinternz

# INDEX

# 1. INTRODUCTION

## 1.1 Overview:

We will be able to write an application that leverages multiple Watson AI Services (Discovery , Assistant, Cloud function and Node Red). By the end of the project, we'll learn best practices of combining Watson services, and how they can build interactive information retrieval systems with Discovery and Assistant.

## 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 owner's manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owner's 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 what text in the owner's manual is important and what is not. This will improve the answers returned from the queries.

### 1.2.1 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

# 2. LITERATURE SURVEY

## 2.1 Existing problem:

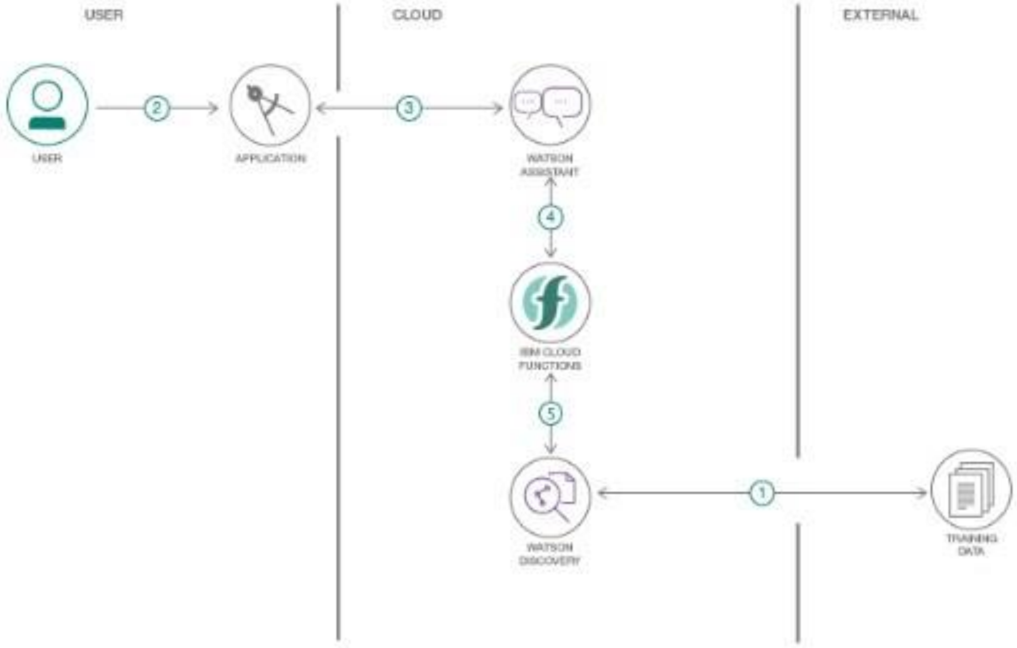
Generally Chatbots means getting input from users and getting only response questions and for some questions the output from bot will be like “try again”, “I don’t understand”, “will you repeat again”, and so on… and directs customer to customer agent but a good customer Chatbot should minimize involvement of customer agent to chat with customer to clarify his/her doubts. So to achieve this we should include an virtual agent in chatbot so that it will take care of real involvement of customer agent and customer can clarifies his doubts with fast chatbots.

## 2.2 Proposed solution:

For the above problem to get solved we have to put an virtual agent in chatbot so it can understand the queries that are posted by customers. The virtual agent should trained from some insight records based company background so it can answer queries based on the product or related to company. In this project I used Watson Discovery to achieve the above solution. And later including Assistant and Discovery on Node-RE

# 3. THEORITICAL ANALYSIS

## 3.1 Block Diagram:



1. The document is annotated using Watson Discovery SDU
2. 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.
3. 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.

### 3.2 Hardware / Software designing:

1. Create IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Create flow and configure node
6. Deploy and run Node Red app.

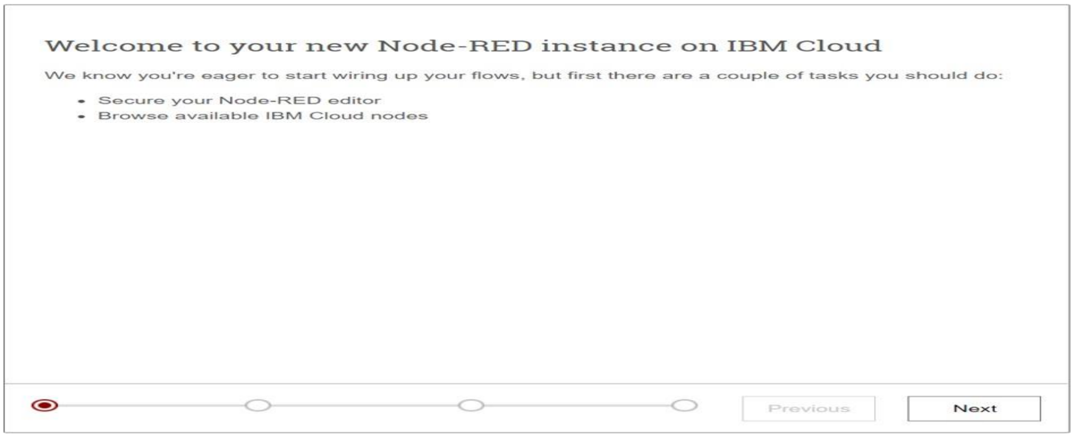
# 4. EXPERIMENTAL INVESTIGATIONS

**Create IBM Cloud services**

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

**Create Node-RED in IBM cloud:**

- Step-1: Login to IBM and go to the catalog
- Step-2: Search for node-red and select “Node-RED Starter “ Service
- Step-3: Enter the Unique name and click on create a button
- Step -4 Your Node-red service is starting
- Step -5: We have to configure Node red for the first time. Click on next to continue

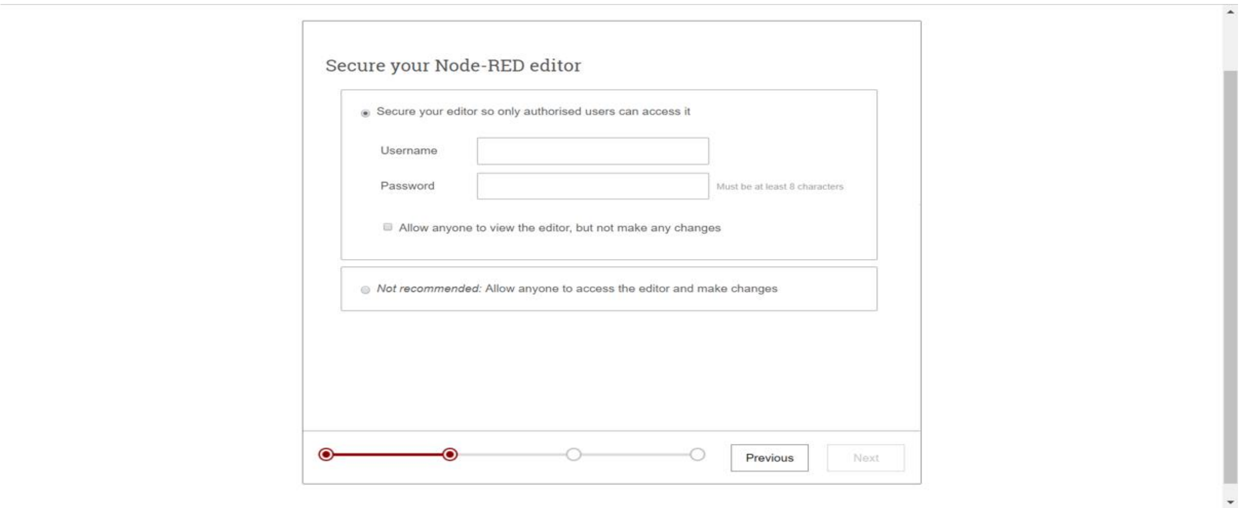


Welcome to your new Node-RED instance on IBM Cloud

We know you're eager to start wiring up your flows, but first there are a couple of tasks you should do:

- Secure your Node-RED editor
- Browse available IBM Cloud nodes

- Step – 6: Secure your node red editor by giving a username and password and click on Next



Secure your Node-RED editor

● Secure your editor so only authorised users can access it

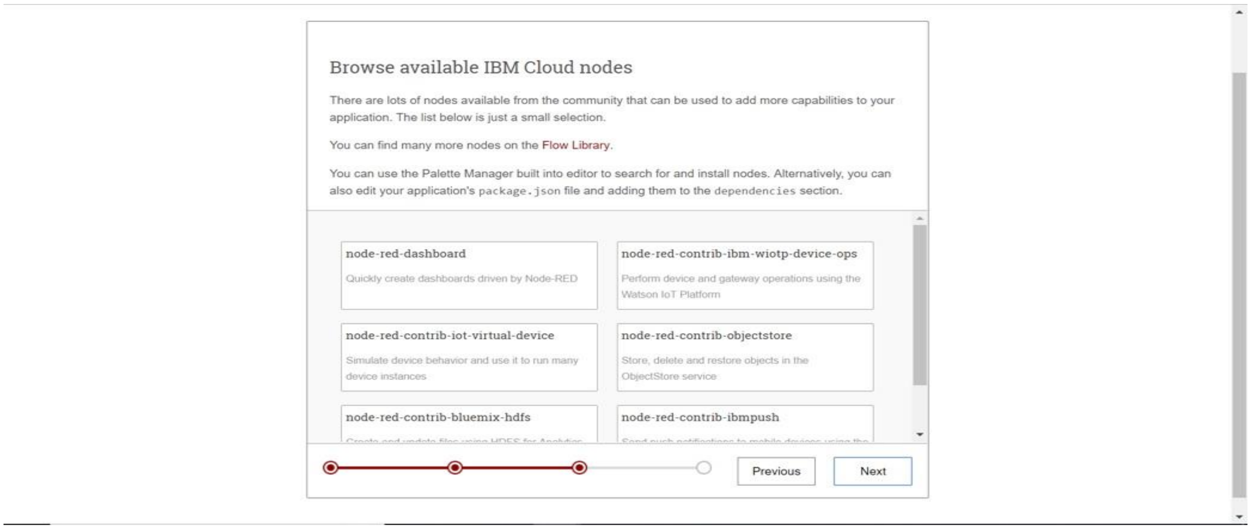
Username

Password — Must be at least 8 characters

□ Allow anyone to view the editor, but not make any changes

○ *Not recommended:* Allow anyone to access the editor and make changes

- Step – 7: Click Next to continue



Browse available IBM Cloud nodes

There are lots of nodes available from the community that can be used to add more capabilities to your application. The list below is just a small selection.

You can find many more nodes on the Flow Library.

You can use the Palette Manager built into editor to search for and install nodes. Alternatively, you can also edit your application's `package.json` file and adding them to the `dependencies` section.

node-red-dashboard
Quickly create dashboards driven by Node-RED

node-red-contrib-ibm-wiotp-device-ops
Perform device and gateway operations using the Watson IoT Platform

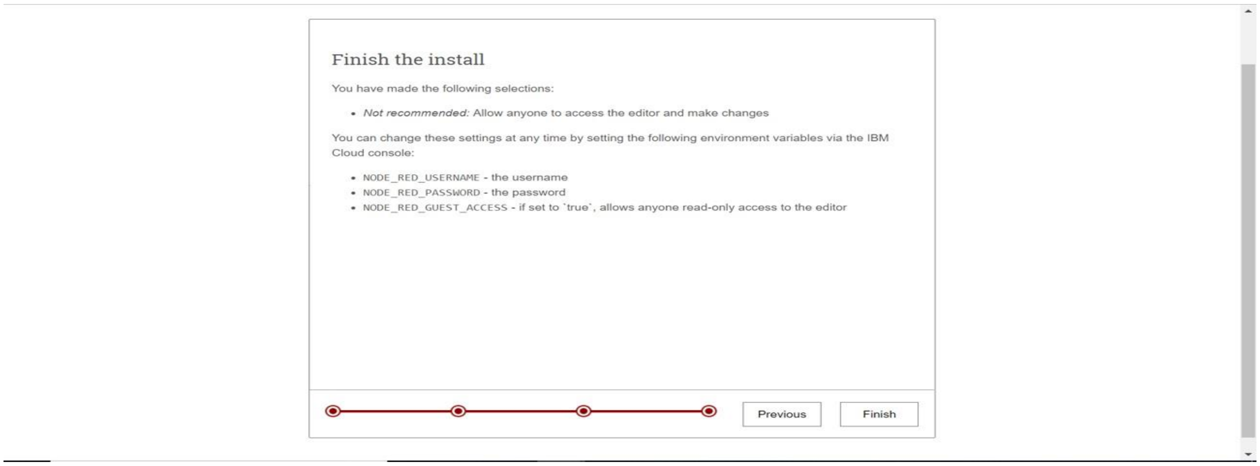
node-red-contrib-iot-virtual-device
Simulate device behavior and use it to run many device instances

node-red-contrib-objectstore
Store, delete and restore objects in the ObjectStore service

node-red-contrib-bluemix-hdfs

node-red-contrib-ibmpush

- Step – 8: Click Finish



Finish the install

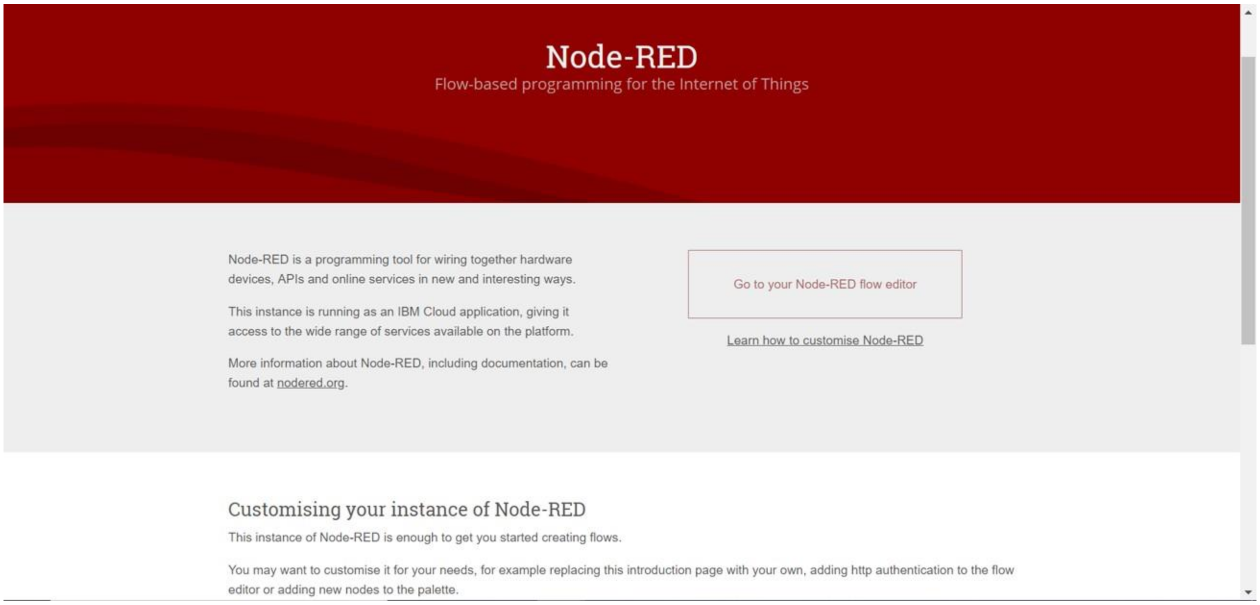
You have made the following selections:

- *Not recommended:* Allow anyone to access the editor and make changes

You can change these settings at any time by setting the following environment variables via the IBM Cloud console:

- `NODE_RED_USERNAME` - the username
- `NODE_RED_PASSWORD` - the password
- `NODE_RED_GUEST_ACCESS` - if set to `true`, allows anyone read-only access to the editor

- Step – 9: Click on Go to Node-Red flow editor to launch the flow editor



Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.

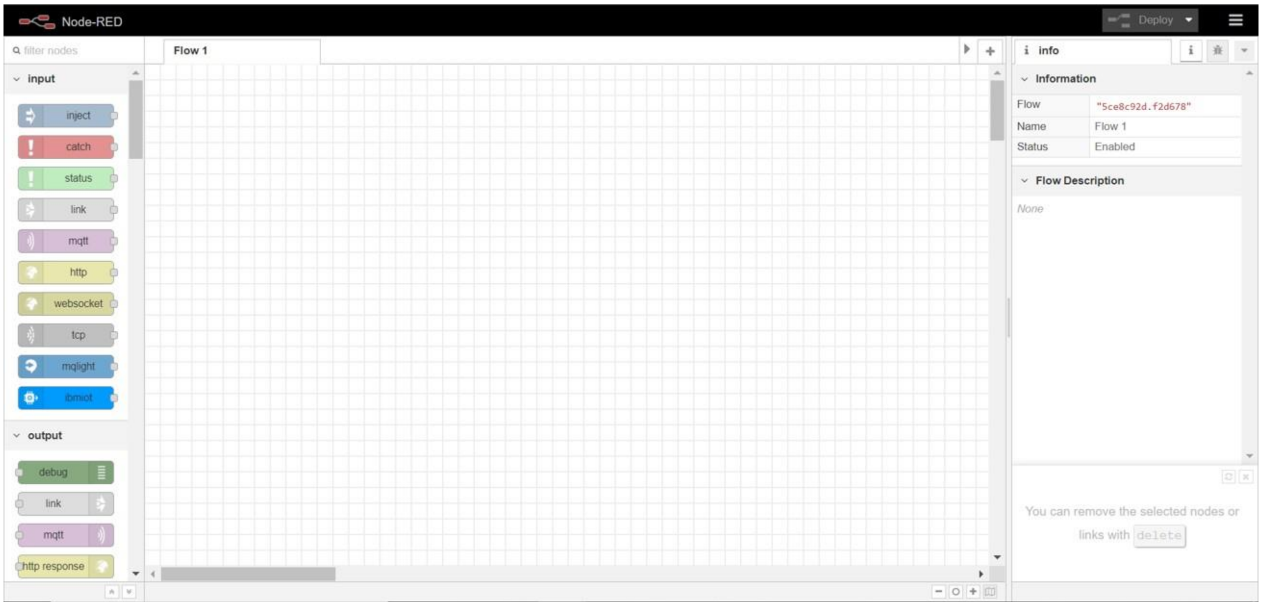
Go to your Node-RED flow editor
Learn how to customise Node-RED

Customising your instance of Node-RED

This instance of Node-RED is enough to get you started creating flows.

You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow editor or adding new nodes to the palette.

- Step – 10: Node red editor has various nodes with the respective functionality



| | |
|---|---|
| Flow | "5ce8c92d.f2d678" |
| Name | Flow 1 |
| Status | Enabled |

None

You can remove the selected nodes or links with delete

## Creation of Watson discovery instance in IBM Cloud:

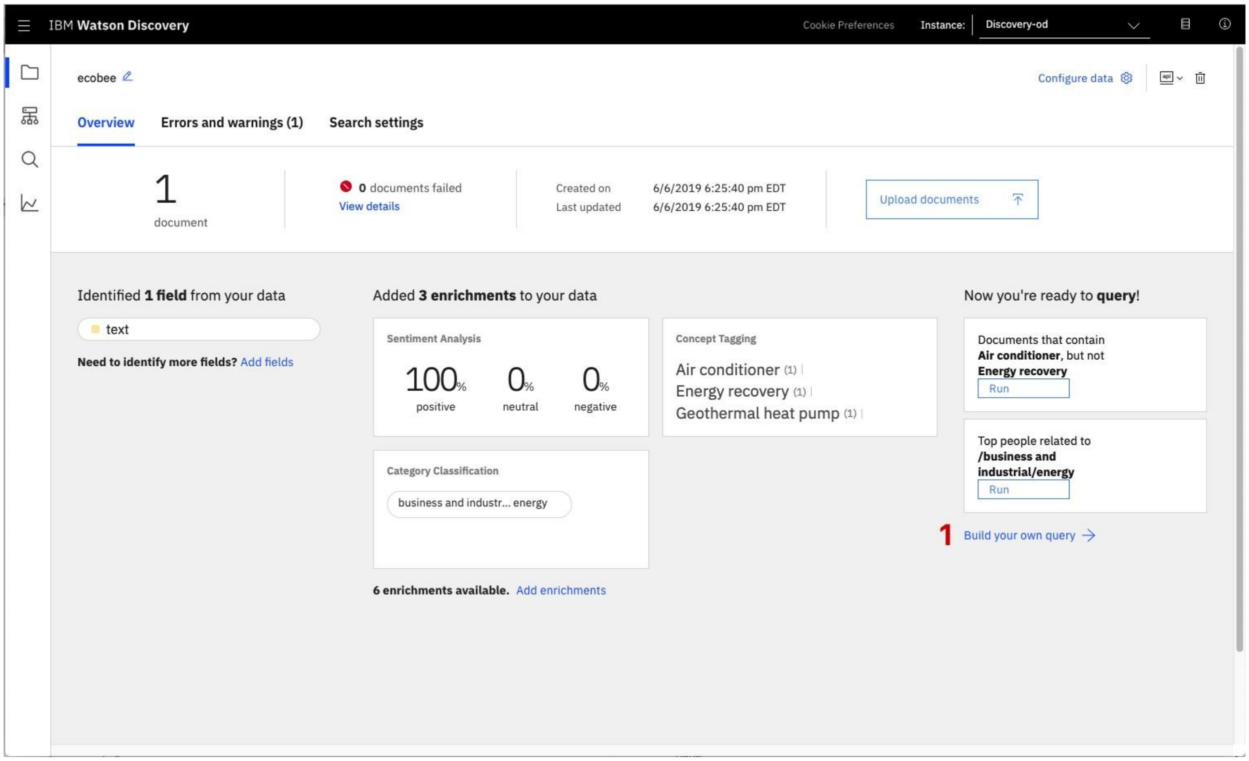
- Import the document

As shown below, launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option. Give the data collection a unique name.

When prompted, select and upload the *ecobee3_UserGuide.pdf*.

The Ecobee is a popular residential thermostat that has a wifi interface and multiple configuration options.

Before applying SDU to our document, lets do some simple queries on the data so that we can compare it to results found after applying SDU.



1 document

0 documents failed

| | |
|---|---|
| Created on | 6/6/2019 6:25:40 pm EDT |
| Last updated | 6/6/2019 6:25:40 pm EDT |

Identified **1 field** from your data

text

**Need to identify more fields?** Add fields

Added **3 enrichments** to your data

Sentiment Analysis

100% positive 0% neutral 0% negative

Concept Tagging

Air conditioner (1)
Energy recovery (1)
Geothermal heat pump (1)

Category Classification

business and industr... energy

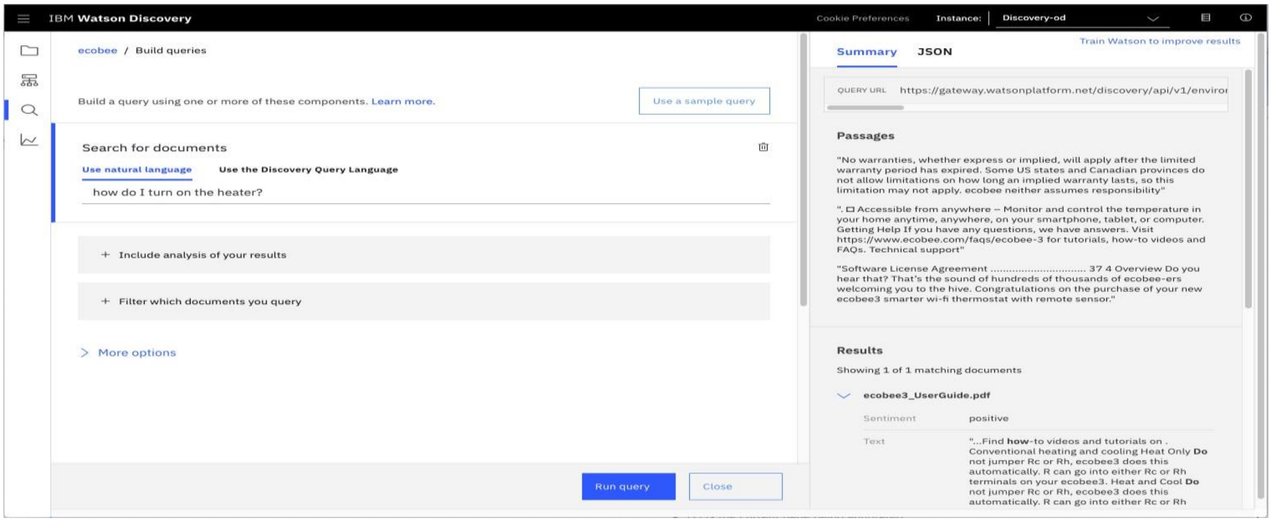
**6 enrichments available.** Add enrichments

Now you're ready to **query**!

Documents that contain **Air conditioner**, but not **Energy recovery**

Top people related to **/business and industrial/energy**

- Click the Build your own query [1] button.


IBM Watson Discovery
Cookie Preferences
Instance:
Discovery-od
ecobee / Build queries
Build a query using one or more of these components. Learn more.
Use a sample query
Search for documents
Use natural language
Use the Discovery Query Language
+ Include analysis of your results
+ Filter which documents you query
> More options
Run query
Close
Summary
JSON
Train Watson to improve results

QUERY URL https://gateway.watsonplatform.net/discovery/api/v1/enviro

how do I turn on the heater?

**Passages**

"No warranties, whether express or implied, will apply after the limited warranty period has expired. Some US states and Canadian provinces do not allow limitations on how long an implied warranty lasts, so this limitation may not apply. ecobee neither assumes responsibility"

". □ Accessible from anywhere – Monitor and control the temperature in your home anytime, anywhere, on your smartphone, tablet, or computer. Getting Help If you have any questions, we have answers. Visit https://www.ecobee.com/faqs/ecobee-3 for tutorials, how-to videos and FAQs. Technical support"

"Software License Agreement ............................... 37 4 Overview Do you hear that? That's the sound of hundreds of thousands of ecobee-ers welcoming you to the hive. Congratulations on the purchase of your new ecobee3 smarter wi-fi thermostat with remote sensor."

**Results**

Showing 1 of 1 matching documents

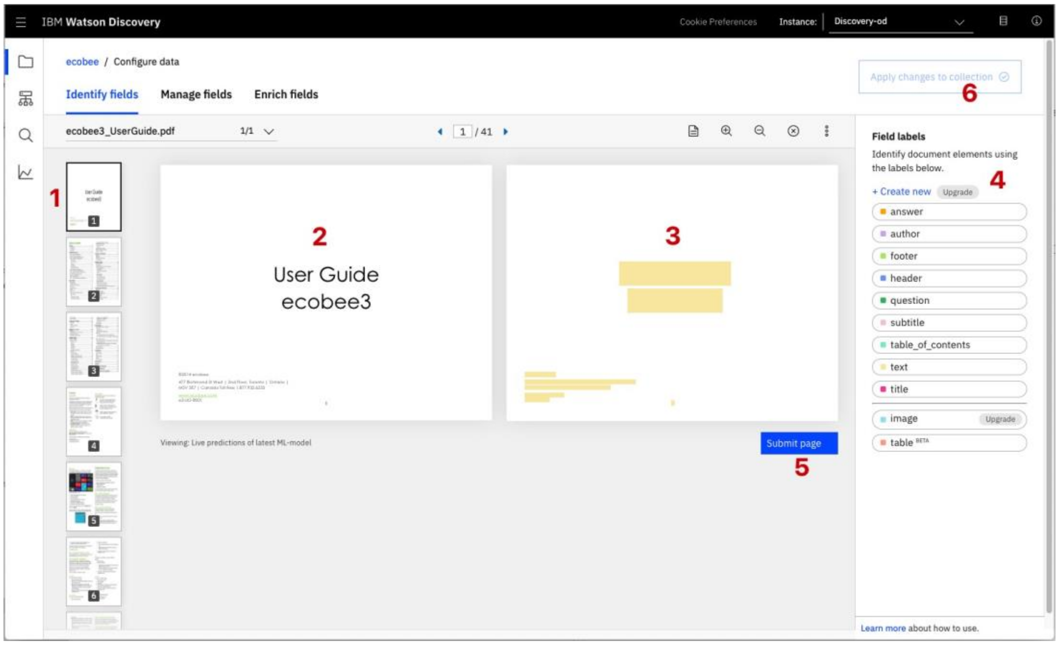
ecobee3_UserGuide.pdf

| | |
|---|---|
| Sentiment | positive |
| Text | "...Find **how**-to videos and tutorials on . Conventional heating and cooling Heat Only **Do** not jumper Rc or Rh, ecobee3 does this automatically. R can go into either Rc or Rh terminals on your ecobee3. Heat and Cool **Do** not jumper Rc or Rh, ecobee3 does this automatically. R can go into either Rc or Rh |

Enter queries related to the operation of the thermostat and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question. **Annotate with SDU**

Now let's apply SDU to our document to see if we can generate some better query responses. From the Discovery collection panel, click the Configure data button (located in the top right corner) to start the SDU process.

Here is the layout of the Identify fields tab of the SDU annotation panel:



The goal is to annotate all of the pages in the document so Discovery can learn what text is important, and what text can be ignored.

- [1] is the list of pages in the manual. As each is processed, a green check mark will appear on the page.
- [2] is the current page being annotated.
- [3] is where you select text and assign it a label.
- [4] is the list of labels you can assign to the page text.
- Click [5] to submit the page to Discovery.
- Click [6] when you have completed the annotation process.

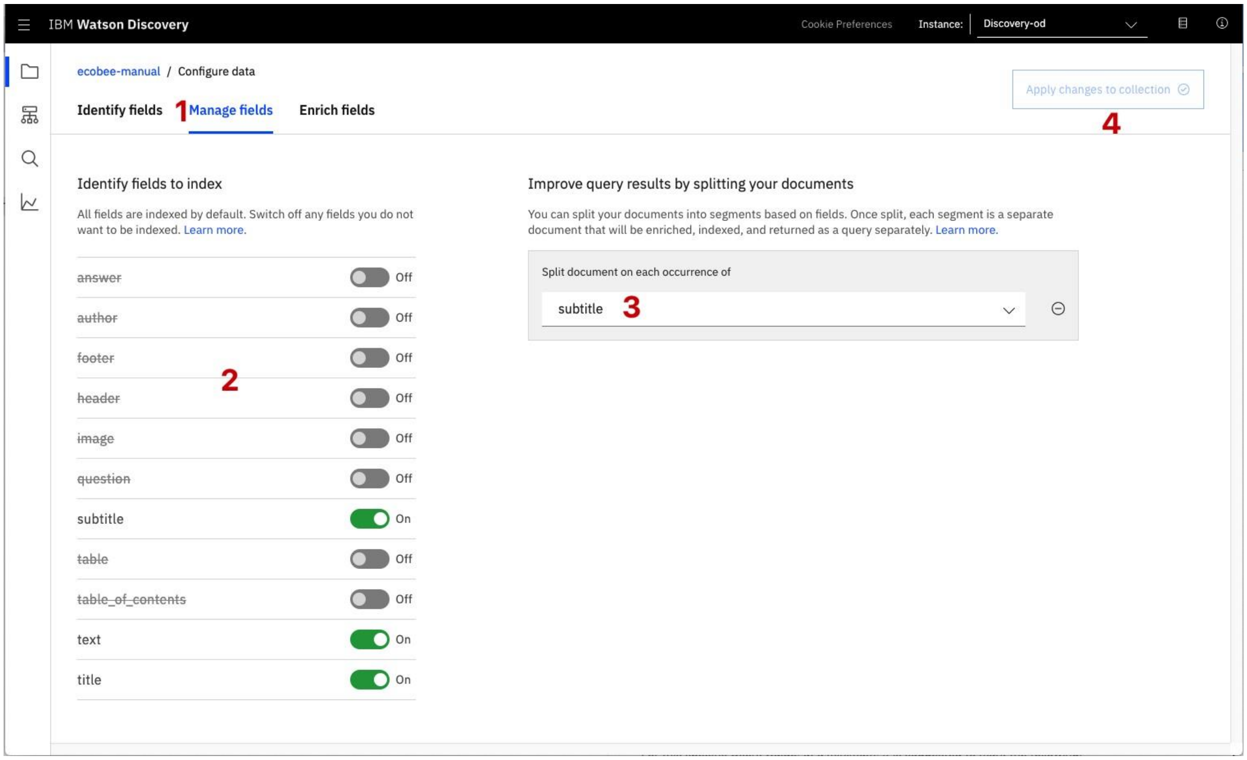
As you go though the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages. Once you get to a page that is already correctly annotated, you can stop, or simply click Submit [5] to acknowledge it is correct. The more pages you annotate, the better the model will be trained.

For this specific owner's manual, at a minimum, it is suggested to mark the following:

- The main title page as 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 infomation (located in the last few pages) as a footer
- All other text should be marked as text.

Once you click the Apply changes to collection button [6], you will be asked to reload the document. Choose the same owner's manual .pdf document as before.

Next, click on the Manage fields [1] tab.



ecobee-manual / Configure data

Identify fields to index

All fields are indexed by default. Switch off any fields you do not want to be indexed. Learn more.

| | |
|---|---|
| ~~answer~~ | Off |
| ~~author~~ | Off |
| ~~footer~~ | Off |
| ~~header~~ | Off |
| ~~image~~ | Off |
| ~~question~~ | Off |
| subtitle | On |
| ~~table~~ | Off |
| ~~table_of_contents~~ | Off |
| text | On |
| title | On |

Improve query results by splitting your documents

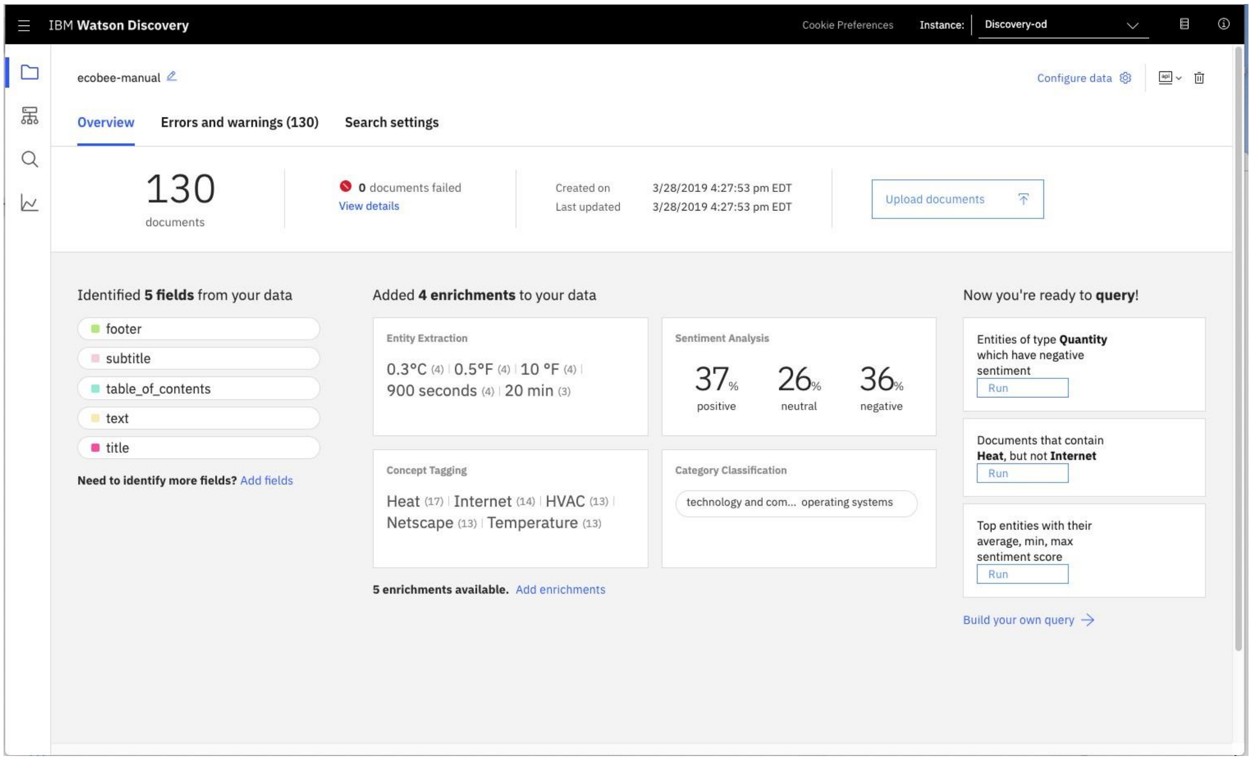
You can split your documents into segments based on fields. Once split, each segment is a separate document that will be enriched, indexed, and returned as a query separately. Learn more.

Split document on each occurrence of: subtitle

- [2] Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except subtitles and text.
- [3] is telling Discovery to split the document apart, based on subtitle.
- Click [4] to submit your changes.

Once again, you will be asked to reload the document.

Now, as a result of splitting the document apart, your collection will look very different:



ecobee-manual

130 documents

0 documents failed

| | |
|---|---|
| Created on | 3/28/2019 4:27:53 pm EDT |
| Last updated | 3/28/2019 4:27:53 pm EDT |

Identified **5 fields** from your data

- footer
- subtitle
- table_of_contents
- text
- title

**Need to identify more fields?** Add fields

Added **4 enrichments** to your data

Entity Extraction: 0.3°C (4) | 0.5°F (4) | 10 °F (4) | 900 seconds (4) | 20 min (3)

Sentiment Analysis: 37% positive, 26% neutral, 36% negative

Concept Tagging: Heat (17) | Internet (14) | HVAC (13) | Netscape (13) | Temperature (13)

Category Classification: technology and com... operating systems

**5 enrichments available.** Add enrichments

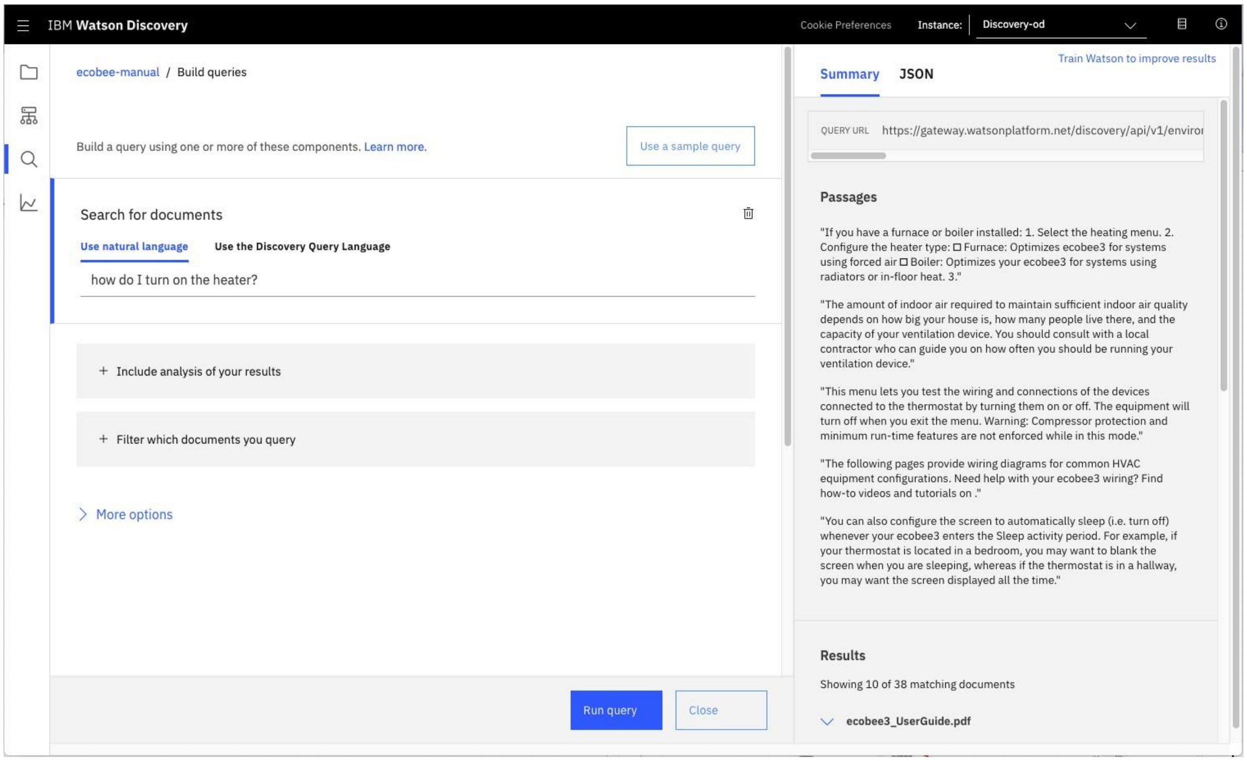
Now you're ready to **query**!

Entities of type **Quantity** which have negative sentiment

Documents that contain **Heat**, but not **Internet**

Top entities with their average, min, max sentiment score

Return to the query panel (click Build your own query) and see how much better the results are.



ecobee-manual / Build queries

Build a query using one or more of these components. Learn more.

Search for documents

how do I turn on the heater?

QUERY URL https://gateway.watsonplatform.net/discovery/api/v1/enviro

**Passages**

"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."

"The amount of indoor air required to maintain sufficient indoor air quality depends on how big your house is, how many people live there, and the capacity of your ventilation device. You should consult with a local contractor who can guide you on how often you should be running your ventilation device."

"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 following pages provide wiring diagrams for common HVAC equipment configurations. Need help with your ecobee3 wiring? Find how-to videos and tutorials on ."

"You can also configure the screen to automatically sleep (i.e. turn off) whenever your ecobee3 enters the Sleep activity period. For example, if your thermostat is located in a bedroom, you may want to blank the screen when you are sleeping, whereas if the thermostat is in a hallway, you may want the screen displayed all the time."

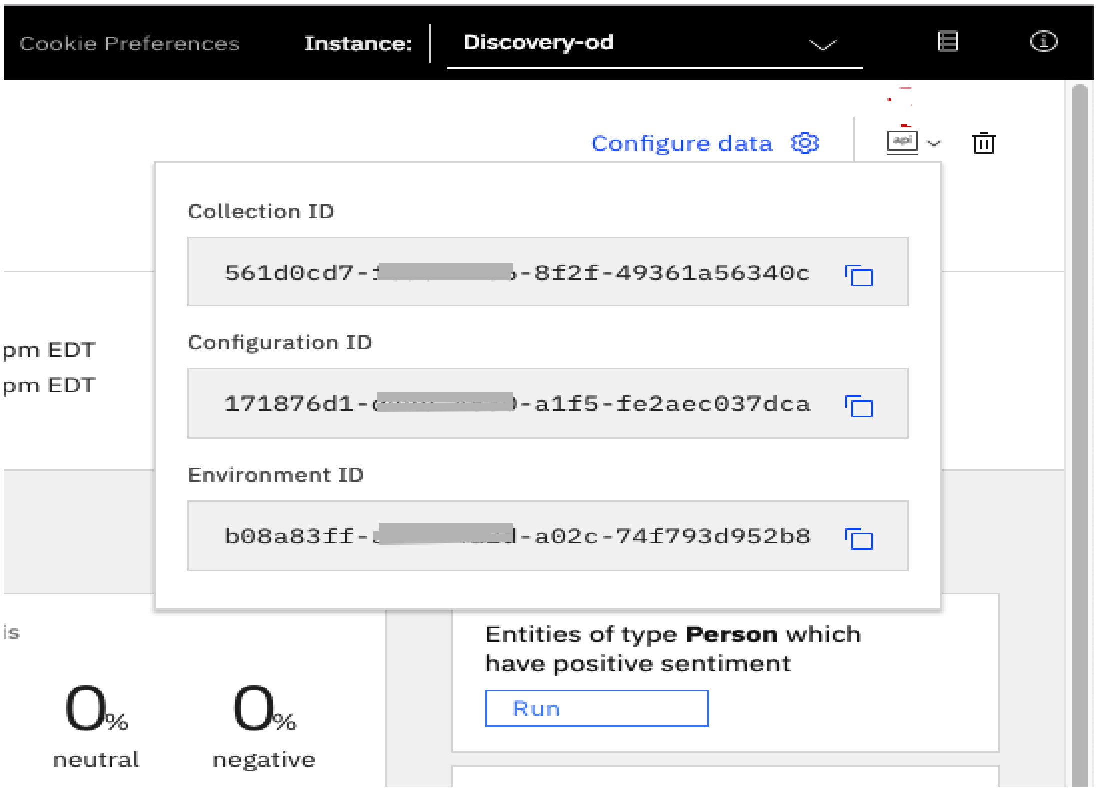
**Results**

Showing 10 of 38 matching documents

ecobee3_UserGuide.pdf

Store credentials for future use

In upcoming steps, you 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 [1] located at the top right side of your collection panel:



Collection ID

561d0cd7-[redacted]-8f2f-49361a56340c

Configuration ID

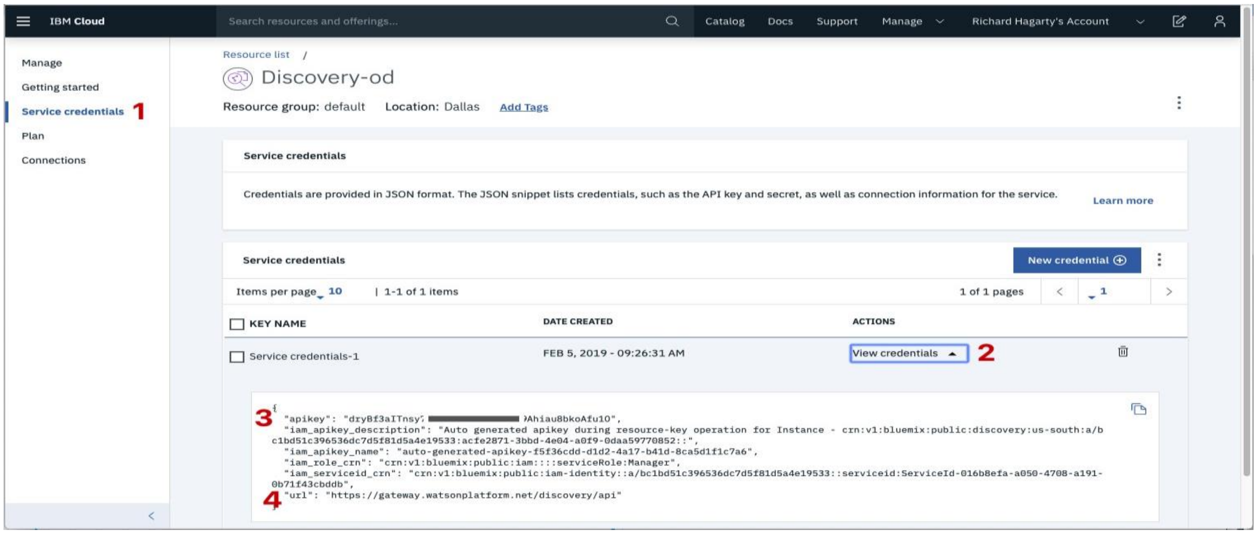
171876d1-[redacted]-a1f5-fe2aec037dca

Environment ID

b08a83ff-[redacted]-a02c-74f793d952b8

Entities of type **Person** which have positive sentiment

For credentials, return to the main panel of your Discovery service, and click the Service credentials [1] tab:



Resource group: default    Location: Dallas

**Service credentials**

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

**Service credentials**

Items per page 10 | 1-1 of 1 items    1 of 1 pages

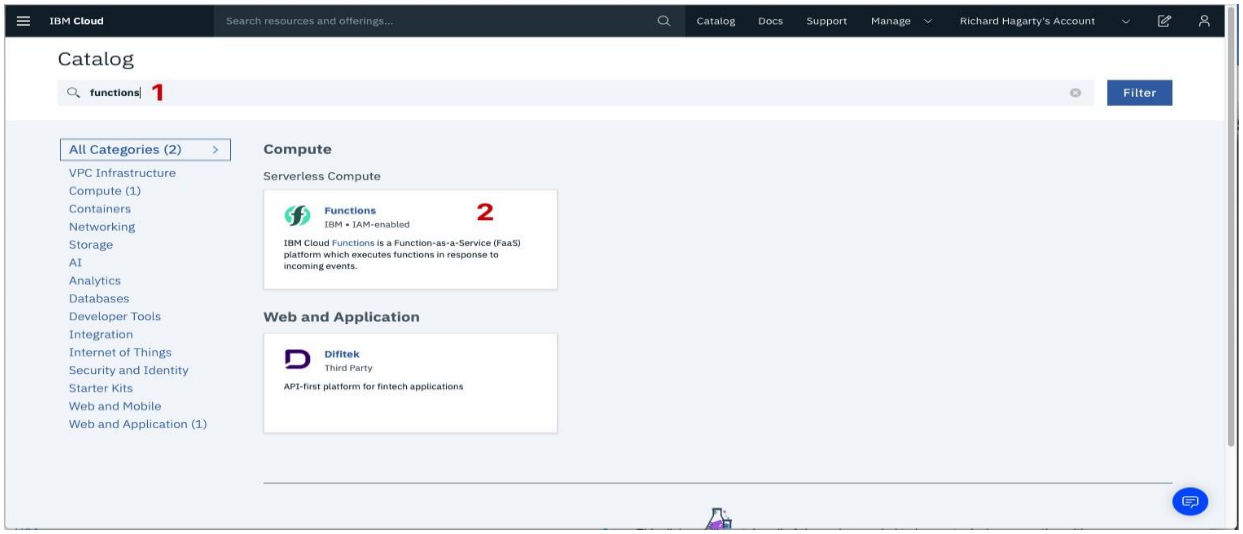
| □ KEY NAME | DATE CREATED | ACTIONS |
|---|---|---|
| □ Service credentials-1 | FEB 5, 2019 - 09:26:31 AM | View credentials |

```
{
  "apikey": "dryBf3aITnsy[redacted]Ahiau8bkoAfu10",
  "iam_apikey_description": "Auto generated apikey during resource-key operation for Instance - crn:v1:bluemix:public:discovery:us-south:a/b
c1bd51c396536dc7d5f81d5a4e19533:acfe2871-3bbd-4e04-a0f9-0daa59770852::",
  "iam_apikey_name": "auto-generated-apikey-f5f36cdd-d1d2-4a17-b41d-8ca5d1f1c7a6",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/bc1bd51c396536dc7d5f81d5a4e19533::serviceid:ServiceId-016b8efa-a050-4708-a191-
0b71f43cbddb",
  "url": "https://gateway.watsonplatform.net/discovery/api"
}
```

Click the View credentials [2] drop-down menu to view the IAM apikey [3] and URL endpoint [4] for your service.

## Creating IBM cloud functions:

Now let's 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 [1], then select the Functions card [2]:



Catalog

functions

Compute

Serverless Compute

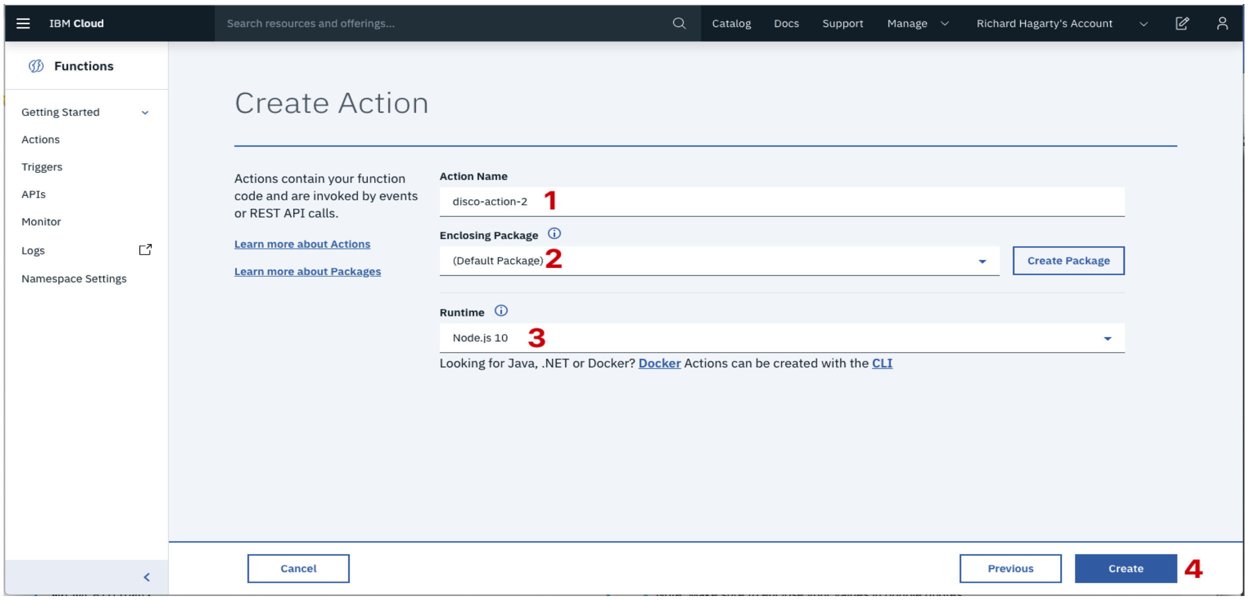
Functions
IBM • IAM-enabled
IBM Cloud Functions is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events.

Web and Application

Difitek
Third Party
API-first platform for fintech applications

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 [1], keep the default package [2], and select the Node.js 10 [3] runtime. Click the Create button [4] to create the action.



Create Action

Actions contain your function code and are invoked by events or REST API calls.

Learn more about Actions

Learn more about Packages

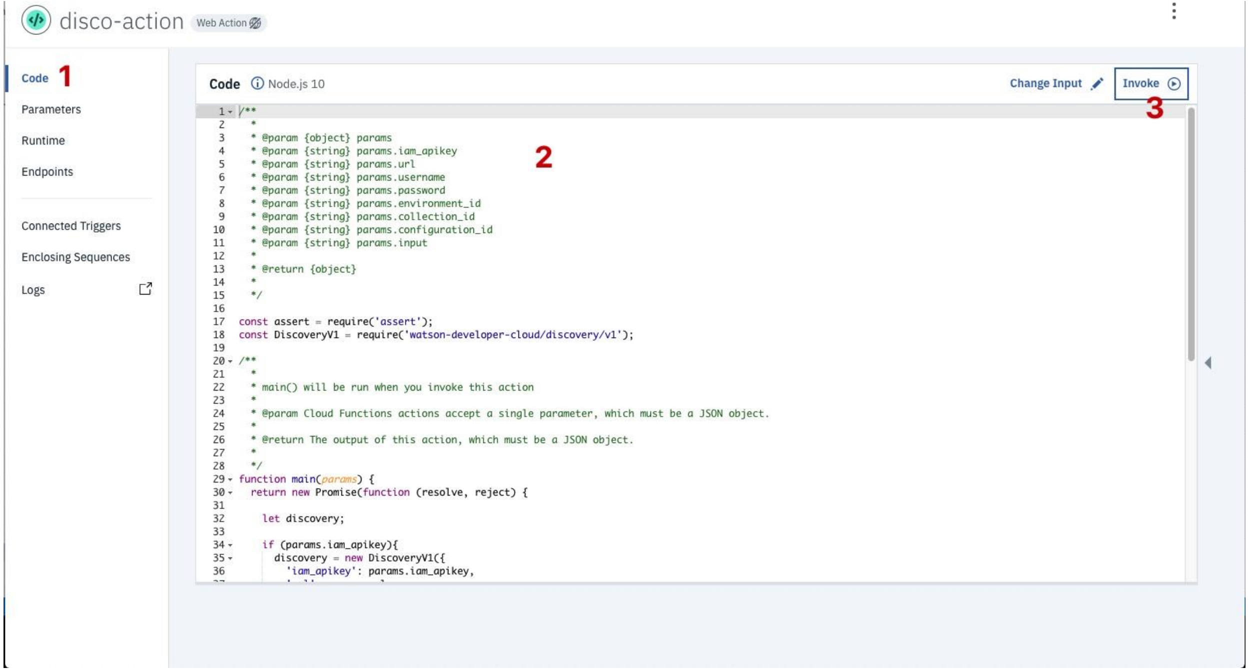
Action Name: disco-action-2

Enclosing Package: (Default Package)

Runtime: Node.js 10

Looking for Java, .NET or Docker? Docker Actions can be created with the CLI

Once your action is created, click on the Code tab [1]:

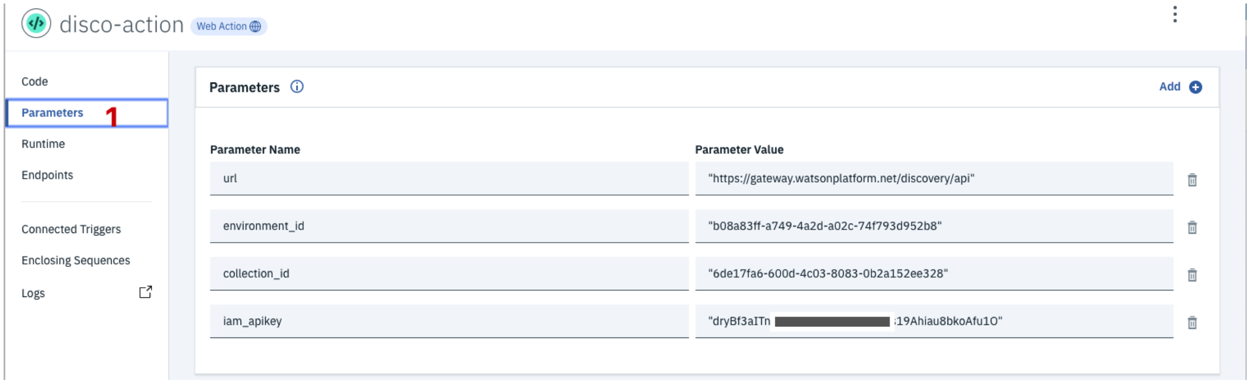


```
/**
  *
  * @param {object} params
  * @param {string} params.iam_apikey
  * @param {string} params.url
  * @param {string} params.username
  * @param {string} params.password
  * @param {string} params.environment_id
  * @param {string} params.collection_id
  * @param {string} params.configuration_id
  * @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,
```

In the code editor window [2], cut and paste in the code from the disco-action.js file found in the actions directory of your local repo. The code is pretty straight-forward - it simply connects to the Discovery service, makes a query against the collection, then returns the response.

If you press the Invoke button [3], it will fail due to credentials not being defined yet. We'll do this next.

Select the Parameters tab [1]:



| Parameter Name | Parameter Value |
|---|---|
| url | "https://gateway.watsonplatform.net/discovery/api" |
| environment_id | "b08a83ff-a749-4a2d-a02c-74f793d952b8" |
| collection_id | "6de17fa6-600d-4c03-8083-0b2a152ee328" |
| iam_apikey | "dryBf3aITn [redacted] :19Ahiau8bkoAfu1O" |

Add the following keys:

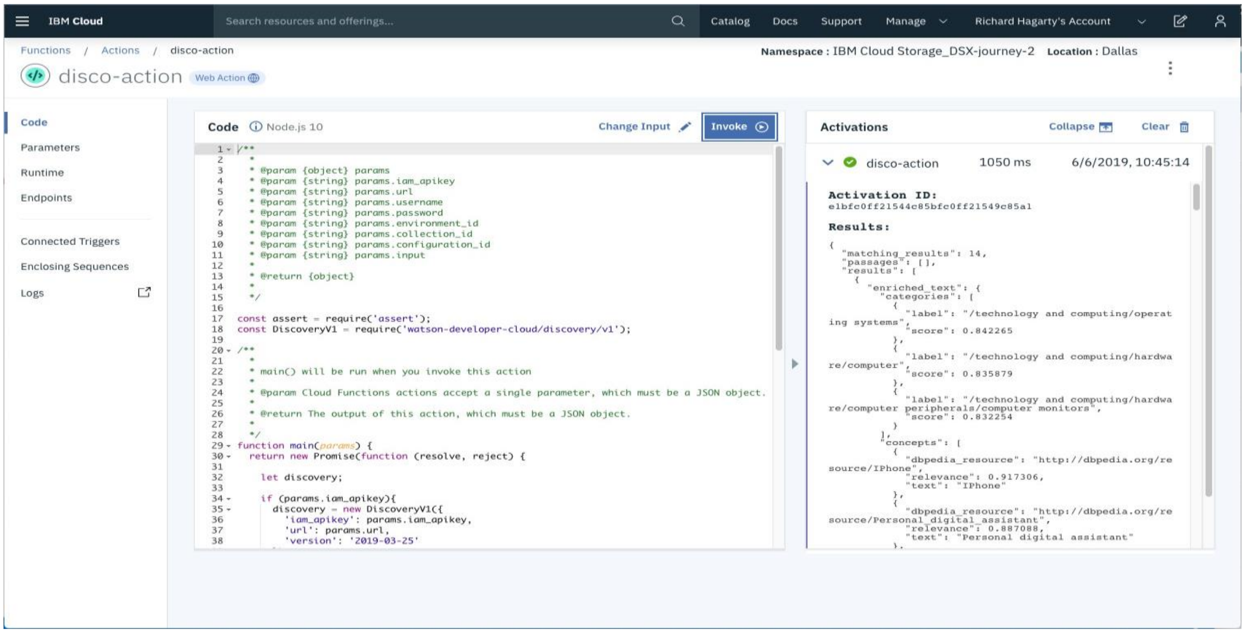
- url
- environment_id
- collection_id

- iam_apikey

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

Note: Make sure to enclose your values in double quotes.

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:



Namespace : IBM Cloud Storage_DSX-journey-2 Location : Dallas

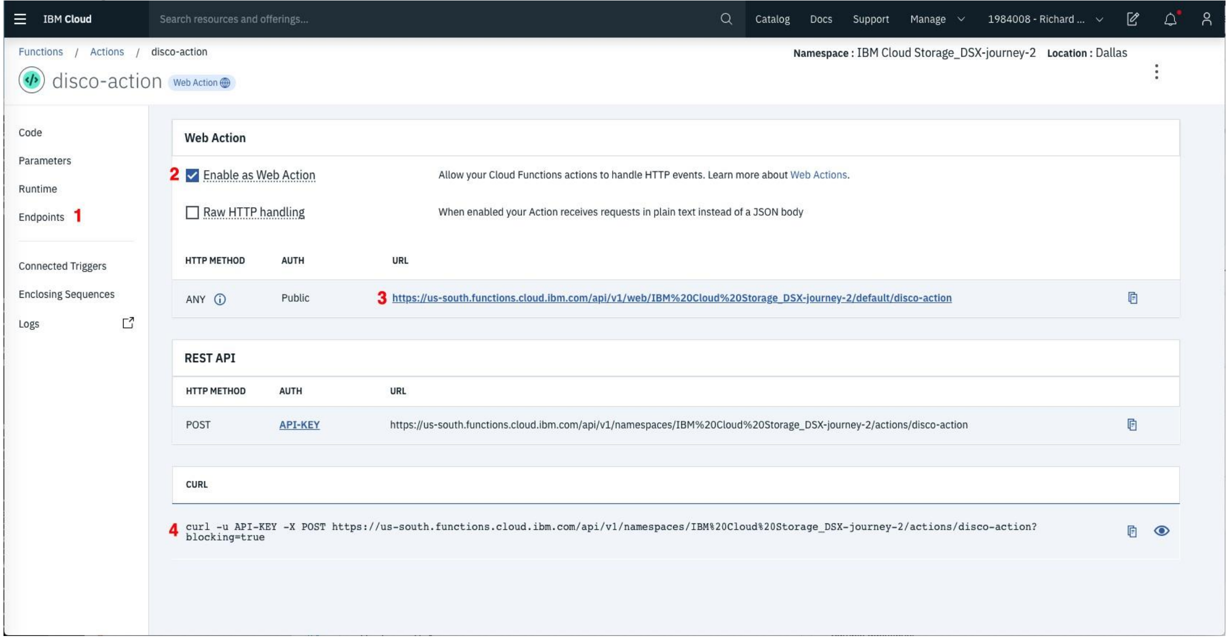
Code ⓘ Node.js 10

```
/**
 *
 * @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'
```

disco-action 1050 ms 6/6/2019, 10:45:14

```
Activation ID:
e1bfc0ff21544c85bfc0ff21549c85a1

Results:
{
  "matching_results": 14,
  "passages": [],
  "results": [
    {
      "enriched_text": {
        "categories": [
          {
            "label": "/technology and computing/operat
ing systems",
            "score": 0.842265
          },
          {
            "label": "/technology and computing/hardwa
re/computer",
            "score": 0.835879
          },
          {
            "label": "/technology and computing/hardwa
re/computer peripherals/computer monitors",
            "score": 0.832254
          }
        ],
        "concepts": [
          {
            "dbpedia_resource": "http://dbpedia.org/re
source/IPhone",
            "relevance": 0.917306,
            "text": "IPhone"
          },
          {
            "dbpedia_resource": "http://dbpedia.org/re
source/Personal_digital_assistant",
            "relevance": 0.887088,
            "text": "Personal digital assistant"
          },
```

Next, go to the Endpoints panel [1]:



Namespace : IBM Cloud Storage_DSX-journey-2 Location : Dallas

**Web Action**

☑ Enable as Web Action — Allow your Cloud Functions actions to handle HTTP events. Learn more about Web Actions.

□ Raw HTTP handling — When enabled your Action receives requests in plain text instead of a JSON body

| HTTP METHOD | AUTH | URL |
|---|---|---|
| ANY | Public | https://us-south.functions.cloud.ibm.com/api/v1/web/IBM%20Cloud%20Storage_DSX-journey-2/default/disco-action |

**REST API**

| HTTP METHOD | AUTH | URL |
|---|---|---|
| POST | API-KEY | https://us-south.functions.cloud.ibm.com/api/v1/namespaces/IBM%20Cloud%20Storage_DSX-journey-2/actions/disco-action |

CURL

```
curl -u API-KEY -X POST https://us-south.functions.cloud.ibm.com/api/v1/namespaces/IBM%20Cloud%20Storage_DSX-journey-2/actions/disco-action?
blocking=true
```

Click the checkbox for Enable as Web Action [2]. This will generate a public endpoint URL [3].

Take note of the URL value [3], as this will be needed by Watson Assistant in a future step.

To verify you have entered the correct Discovery parameters, execute the provied curl command [4]. If it fails, re-check your parameter values.

NOTE: An IBM Cloud Functions service will not show up in your dashboard resource list. To return to your defined Action, you will need to access Cloud Functions by selecting Create Resource from the main dashboard panel (as shown at the beginning of this step).

## Configure Wstaon Assistant:

As shown below, launch the Watson Assistant tool and create a new dialog skill. Select the Use sample skill option as your starting point.

This dialog skill contains all of the nodes needed to have a typical call center conversation with a user.

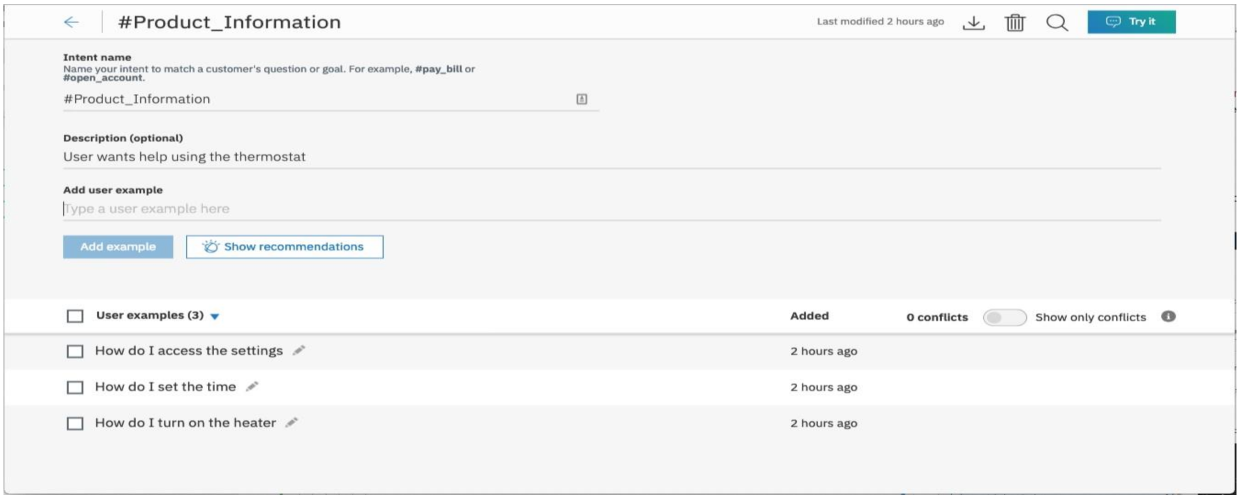
## Add new intent

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.



#Product_Information

**Intent name**
Name your intent to match a customer's question or goal. For example, **#pay_bill** or **#open_account**.

#Product_Information

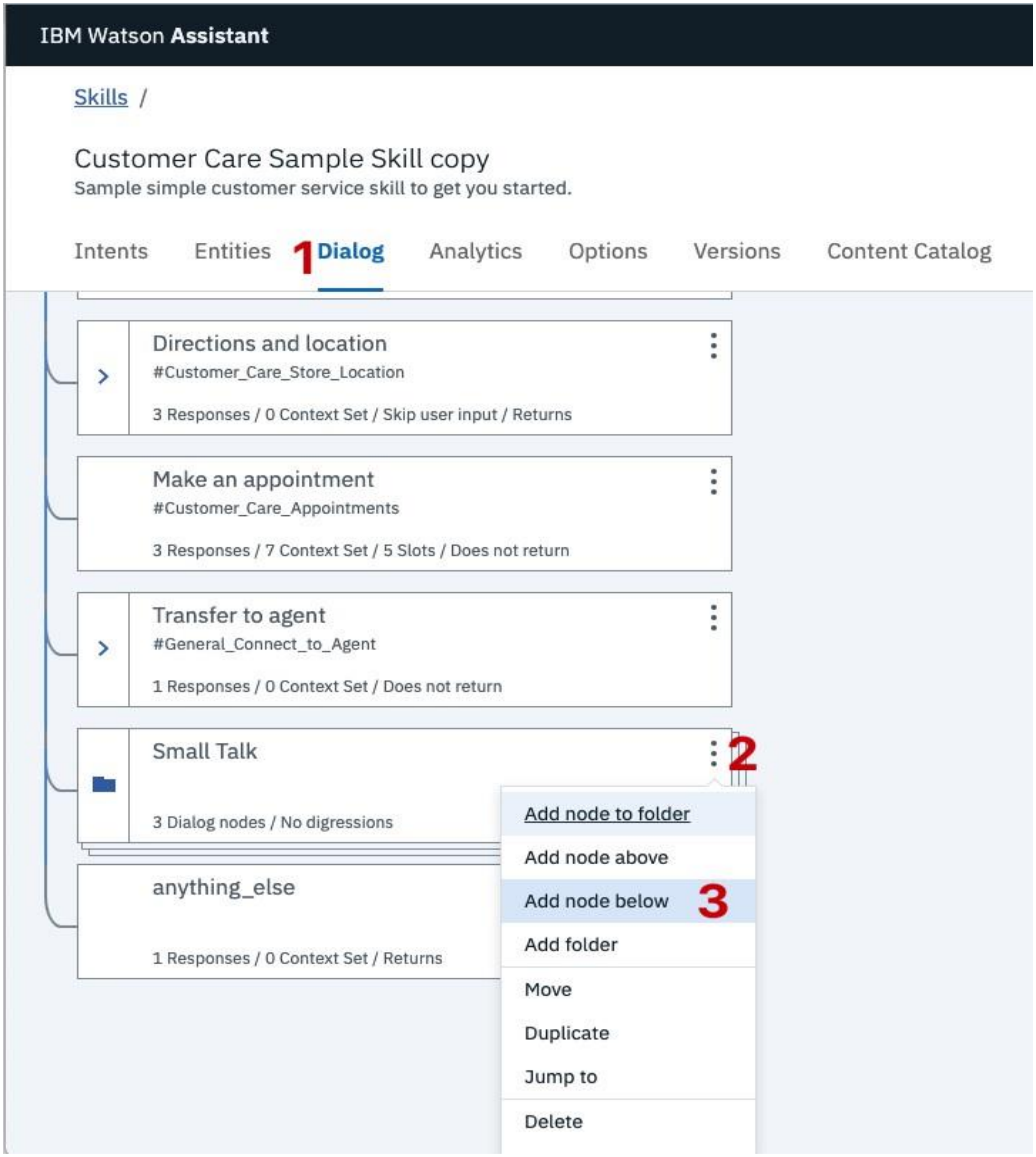
**Description (optional)**
User wants help using the thermostat

**Add user example**
Type a user example here

| □ User examples (3) | Added |
|---|---|
| □ How do I access the settings | 2 hours ago |
| □ How do I set the time | 2 hours ago |
| □ How do I turn on the heater | 2 hours ago |

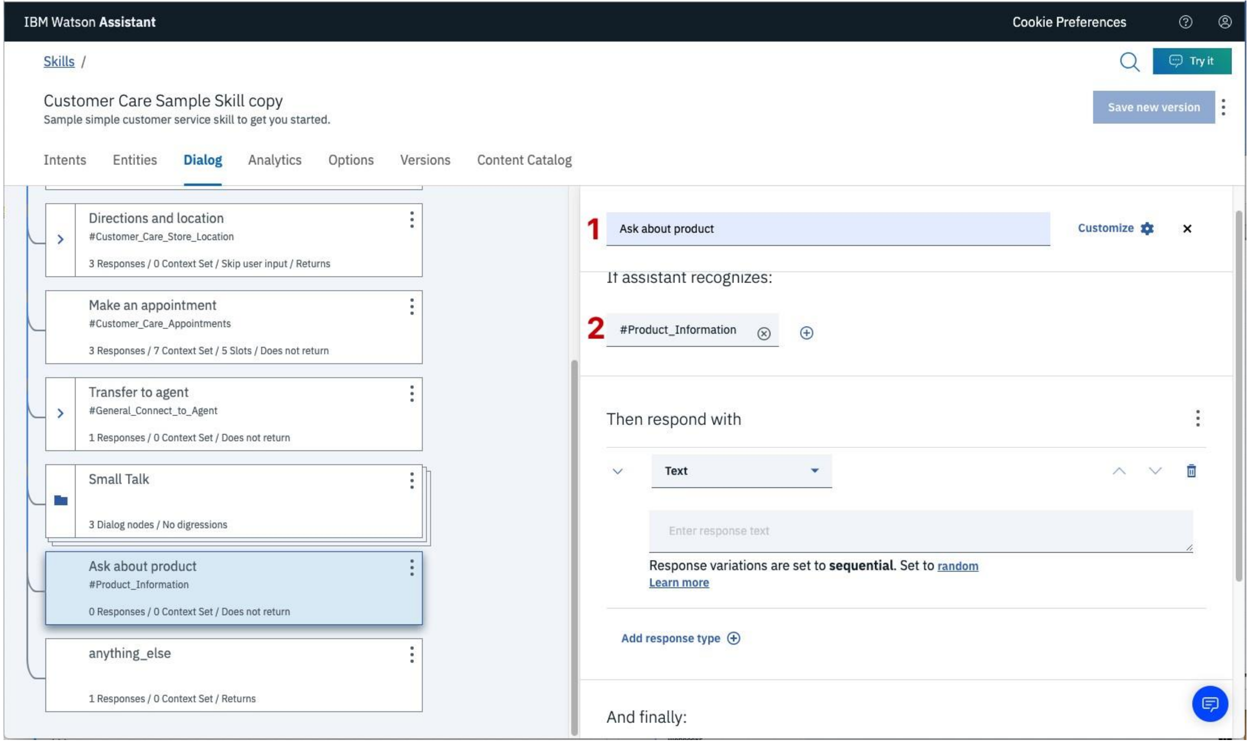
### Create new dialog node

Now we need to add a node to handle our intent. Click on the Dialog [1] tab, then click on the drop down menu for the Small Talk node [2], and select the Add node below [3] option.



IBM Watson Assistant
Skills /
Customer Care Sample Skill copy
Sample simple customer service skill to get you started.
Intents
Entities
1
Dialog
Analytics
Options
Versions
Content Catalog
Directions and location
#Customer_Care_Store_Location
3 Responses / 0 Context Set / Skip user input / Returns
Make an appointment
#Customer_Care_Appointments
3 Responses / 7 Context Set / 5 Slots / Does not return
Transfer to agent
#General_Connect_to_Agent
1 Responses / 0 Context Set / Does not return
Small Talk
3 Dialog nodes / No digressions
2
anything_else
1 Responses / 0 Context Set / Returns
Add node to folder
Add node above
Add node below
3
Add folder
Move
Duplicate
Jump to
Delete

Name the node "Ask about product" [1] and assign it our new intent [2].



Customer Care Sample Skill copy
Sample simple customer service skill to get you started.

Directions and location
#Customer_Care_Store_Location
3 Responses / 0 Context Set / Skip user input / Returns

Make an appointment
#Customer_Care_Appointments
3 Responses / 7 Context Set / 5 Slots / Does not return

Transfer to agent
#General_Connect_to_Agent
1 Responses / 0 Context Set / Does not return

Small Talk
3 Dialog nodes / No digressions

Ask about product
#Product_Information
0 Responses / 0 Context Set / Does not return

anything_else
1 Responses / 0 Context Set / Returns

Ask about product

If assistant recognizes:

#Product_Information

Then respond with

Enter response text

Response variations are set to **sequential**. Set to random
Learn more

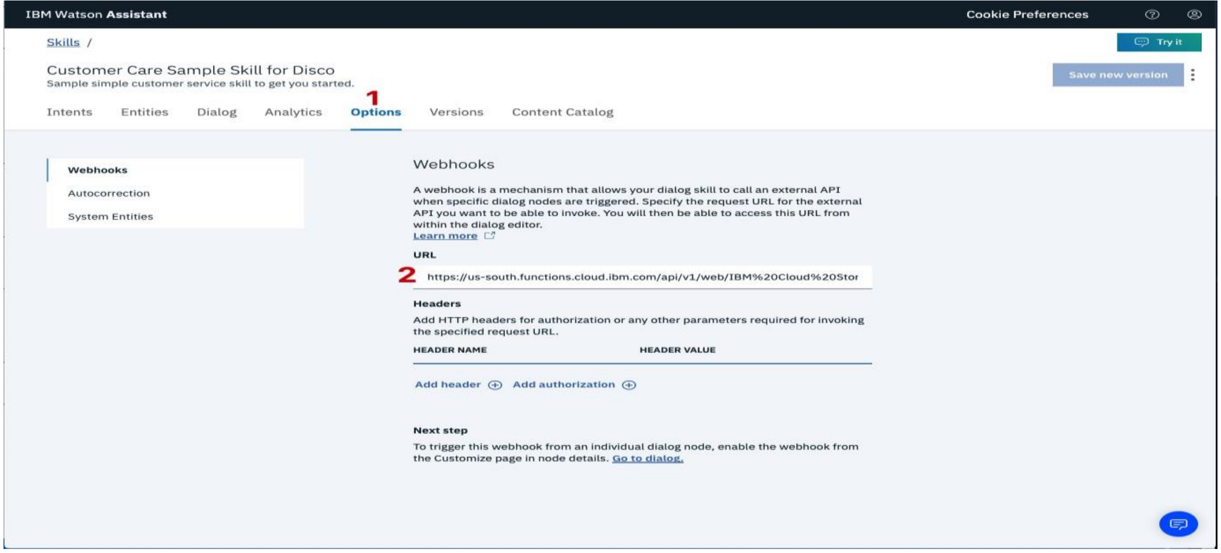
And finally:

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 [1]:



Customer Care Sample Skill for Disco
Sample simple customer service skill to get you started.

Webhooks

A webhook is a mechanism that allows your dialog skill to call an external API when specific dialog nodes are triggered. Specify the request URL for the external API you want to be able to invoke. You will then be able to access this URL from within the dialog editor.
Learn more

URL
https://us-south.functions.cloud.ibm.com/api/v1/web/IBM%20Cloud%20Stor

Headers
Add HTTP headers for authorization or any other parameters required for invoking the specified request URL.

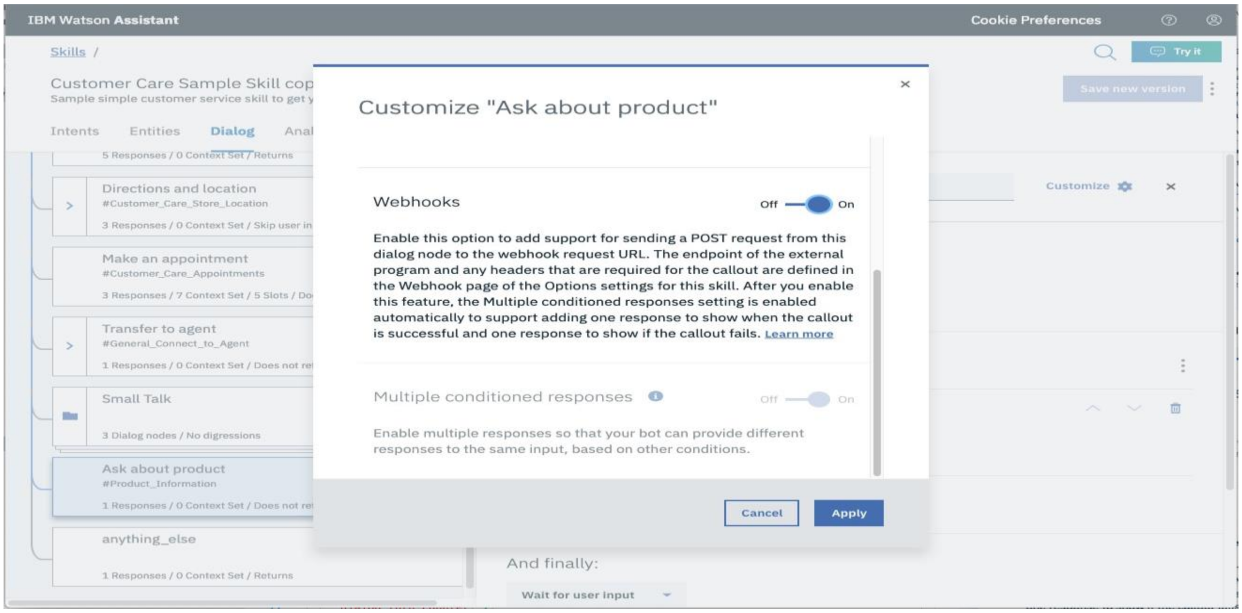
HEADER NAME HEADER VALUE

Next step
To trigger this webhook from an individual dialog node, enable the webhook from the Customize page in node details. Go to dialog.

Enter the public URL endpoint for your action [2].

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:



Customize "Ask about product"

Webhooks — Off ● On

Enable this option to add support for sending a POST request from this dialog node to the webhook request URL. The endpoint of the external program and any headers that are required for the callout are defined in the Webhook page of the Options settings for this skill. After you enable this feature, the Multiple conditioned responses setting is enabled automatically to support adding one response to show when the callout is successful and one response to show if the callout fails. Learn more

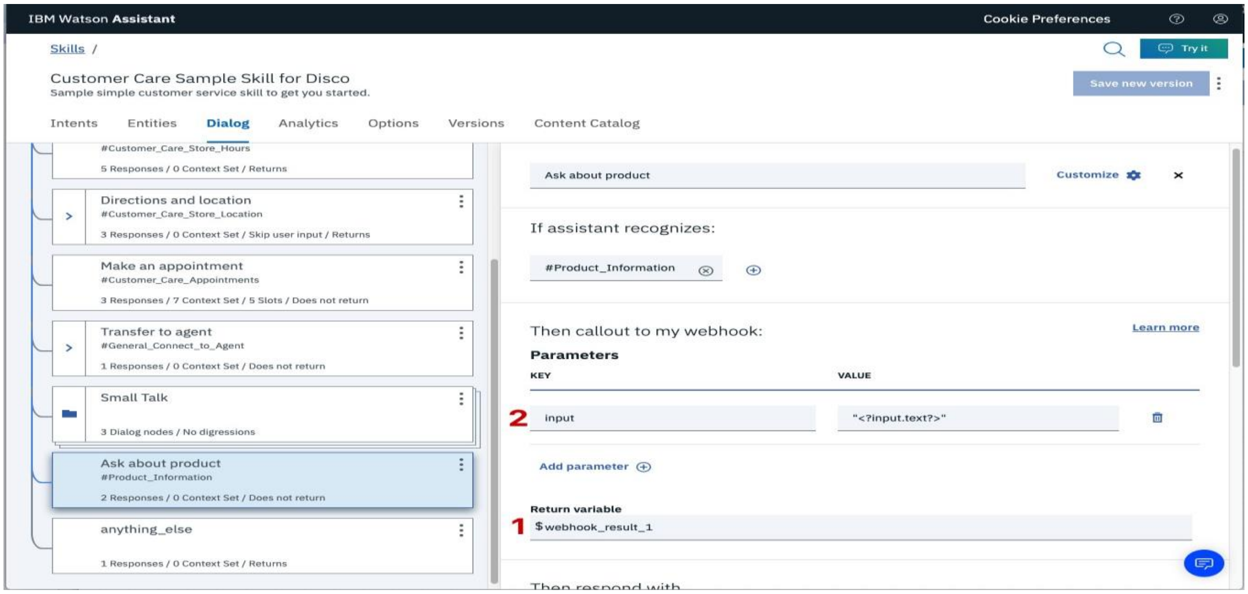
Multiple conditioned responses — Off ● On

Enable multiple responses so that your bot can provide different responses to the same input, based on other conditions.

IBM Watson Assistant
Cookie Preferences
Skills /
Try it
Customer Care Sample Skill cop
Sample simple customer service skill to get y
Save new version
Intents
Entities
Dialog
Anal
5 Responses / 0 Context Set / Returns
Directions and location
#Customer_Care_Store_Location
3 Responses / 0 Context Set / Skip user in
Make an appointment
#Customer_Care_Appointments
3 Responses / 7 Context Set / 5 Slots / Do
Transfer to agent
#General_Connect_to_Agent
1 Responses / 0 Context Set / Does not re
Small Talk
3 Dialog nodes / No digressions
Ask about product
#Product_Information
1 Responses / 0 Context Set / Does not re
anything_else
1 Responses / 0 Context Set / Returns
Customize
Cancel
Apply
And finally:
Wait for user input

Click Apply.

The dialog node should have a Return variable [1] set automatically to $webhook_result_1. This is the variable name you can use to access the result from the Discovery service query.



Ask about product

If assistant recognizes:

#Product_Information

Then callout to my webhook:

Parameters

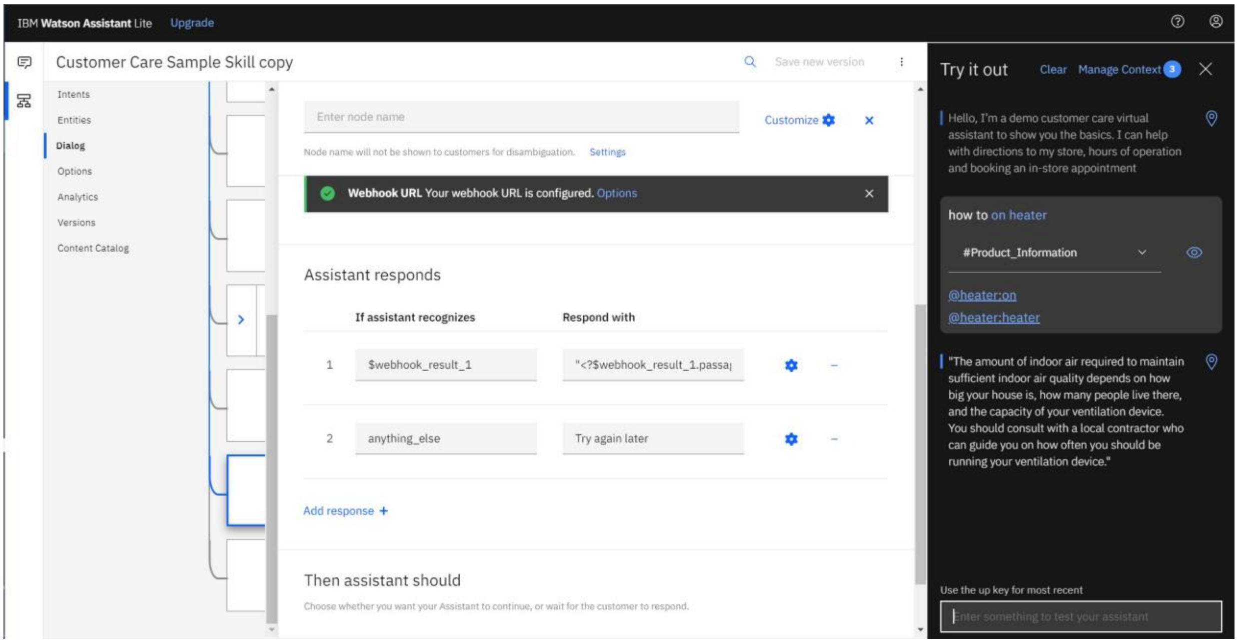
| KEY | VALUE |
|---|---|
| input | "<?input.text?>" |

Return variable

$webhook_result_1

IBM Watson Assistant
Cookie Preferences
Skills /
Try it
Customer Care Sample Skill for Disco
Sample simple customer service skill to get you started.
Save new version
Intents
Entities
Dialog
Analytics
Options
Versions
Content Catalog
#Customer_Care_Store_Hours
5 Responses / 0 Context Set / Returns
Directions and location
#Customer_Care_Store_Location
3 Responses / 0 Context Set / Skip user input / Returns
Make an appointment
#Customer_Care_Appointments
3 Responses / 7 Context Set / 5 Slots / Does not return
Transfer to agent
#General_Connect_to_Agent
1 Responses / 0 Context Set / Does not return
Small Talk
3 Dialog nodes / No digressions
Ask about product
#Product_Information
2 Responses / 0 Context Set / Does not return
anything_else
1 Responses / 0 Context Set / Returns
Customize
Learn more
2
Add parameter
1
Then respond with

You will also need to pass in the users 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:
Add Add “<?webhook_result_1.passages[0].passage_text?>” in respond with in Assistant responds block as shown below.



Enter node name

Node name will not be shown to customers for disambiguation.

**Webhook URL** Your webhook URL is configured. Options

Assistant responds

| | If assistant recognizes | Respond with |
|---|---|---|
| 1 | $webhook_result_1 | "<?$webhook_result_1.passa |
| 2 | anything_else | Try again later |

Then assistant should

Choose whether you want your Assistant to continue, or wait for the customer to respond.

Hello, I'm a demo customer care virtual assistant to show you the basics. I can help with directions to my store, hours of operation and booking an in-store appointment

how to on heater

#Product_Information

@heater:on
@heater:heater

"The amount of indoor air required to maintain sufficient indoor air quality depends on how big your house is, how many people live there, and the capacity of your ventilation device. You should consult with a local contractor who can guide you on how often you should be running your ventilation device."

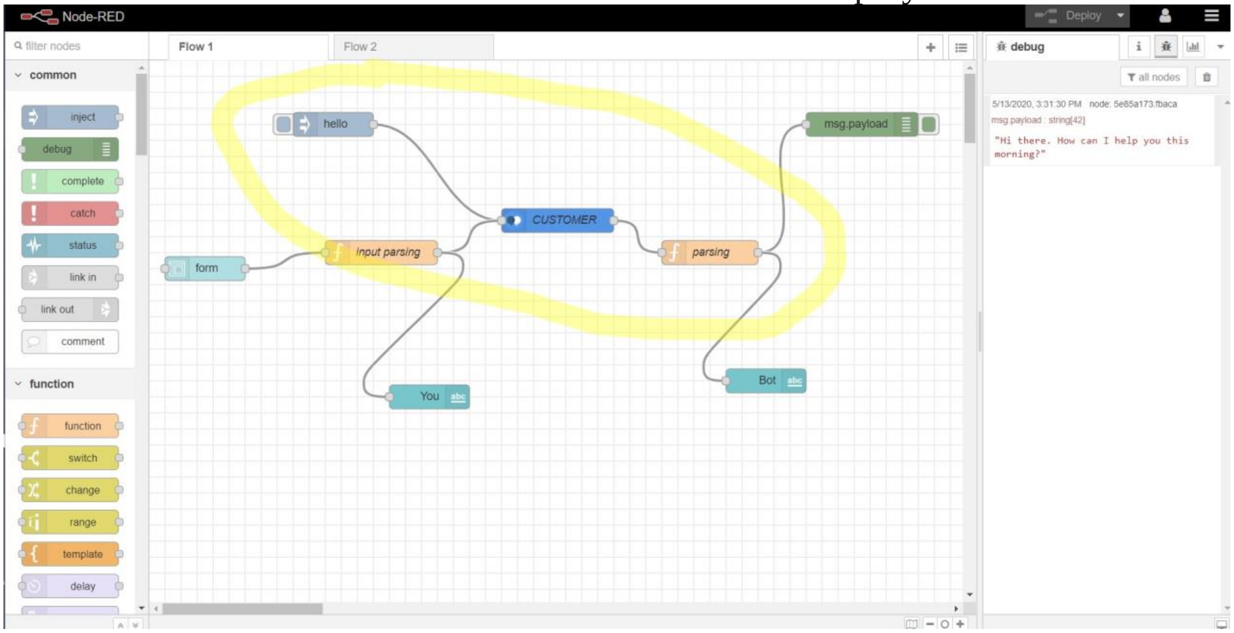
Use the up key for most recent

Enter something to test your assistant

## Integration of watson assistant in Node-RED

- Double-click on the Watson assistant node
- Give a name to your node and enter the username, password and workspace id of your Watson assistant service
- After entering all the information click on Done
- Drag Debug on to the flow from the output section
- Select the payload as a string
- Connect the nodes as shown below and click on Deploy
- Open Debug window as shown below
- Click on the button to send input text to the assistant node
- Observe the output from the assistant service node

- Drag the function node to parse the JSON data and get the results.
- Double click on the function node and enter the JSON parsing code as shown below and click on done
- Connect the nodes as shown below and click on Deploy



```
5/13/2020, 3:31:30 PM   node: 5e85a173.fbaca
msg.payload : string[42]
"Hi there. How can I help you this
morning?"
```

We are done integrating Watson assistant service to Node-red. In the next lab, we will create a web application using Node-red for the chatbot. For creating a web application UI we need dashboard nodes which should be installed manually.

- Go to navigation pane and click on manage palette
- Click on install
- Search for “node-red-dashboard” and click on install and again click on install on the prompt
- The following message indicates dashboard nodes are installed, close the manage palette
- Search for “Form” node and drag on to the flow
- Doube click on the “form” node to configure
- Click on the edit button to add the “Group” name and “Tab” name
- Click on the edit button to add tab name to web application
- Give sample tab name and click on add do the same thing for the group
- Give the label as “Enter your input”, Name as “text” and click on Done
- Drag a function node, double-click on it and enter the input parsing code

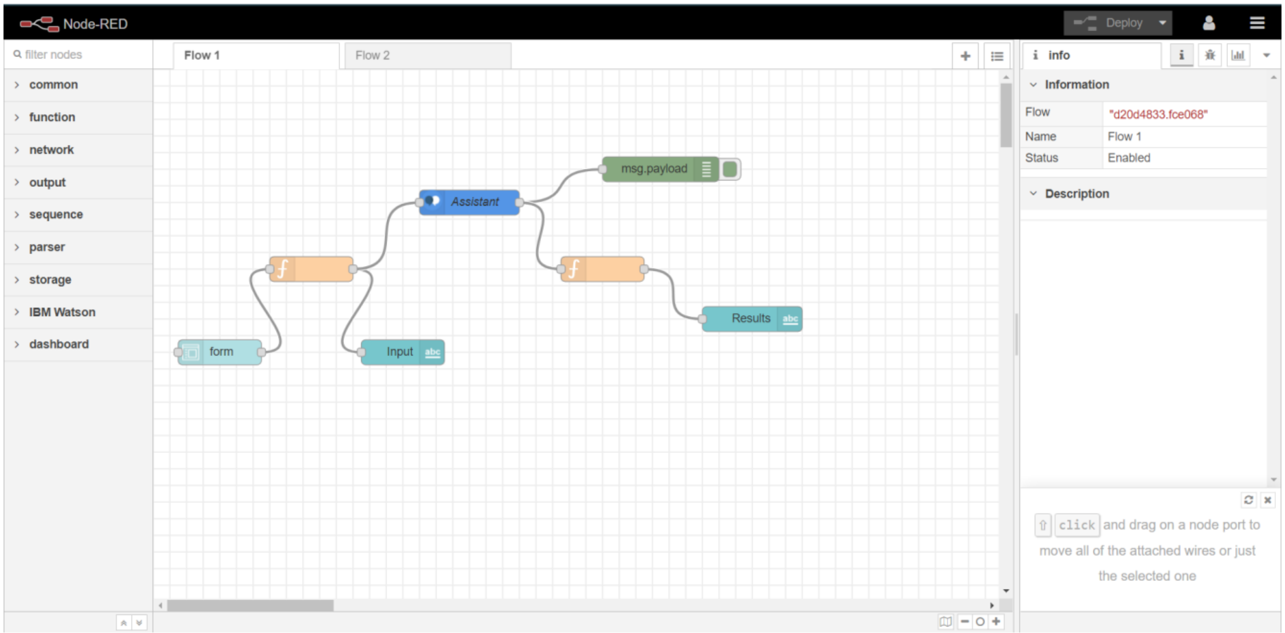
- Click on done
- Connect the form output to the input of the function node and output of the function to input of assistant node
- Search for “text” node from the “dashboard” section
- Drag two “text” nodes on to the flow
- Double click on the first text node, change the label as “Input” and click on Done
- Double click on the second text node, change the label as “Results” and click on Done
- Connect the output of input parsing function node to “ Input” text node and output of parsing function node to the input of “Results” text node
- Click on Deploy

# 5. FLOWCHART

Create flow and configure node:

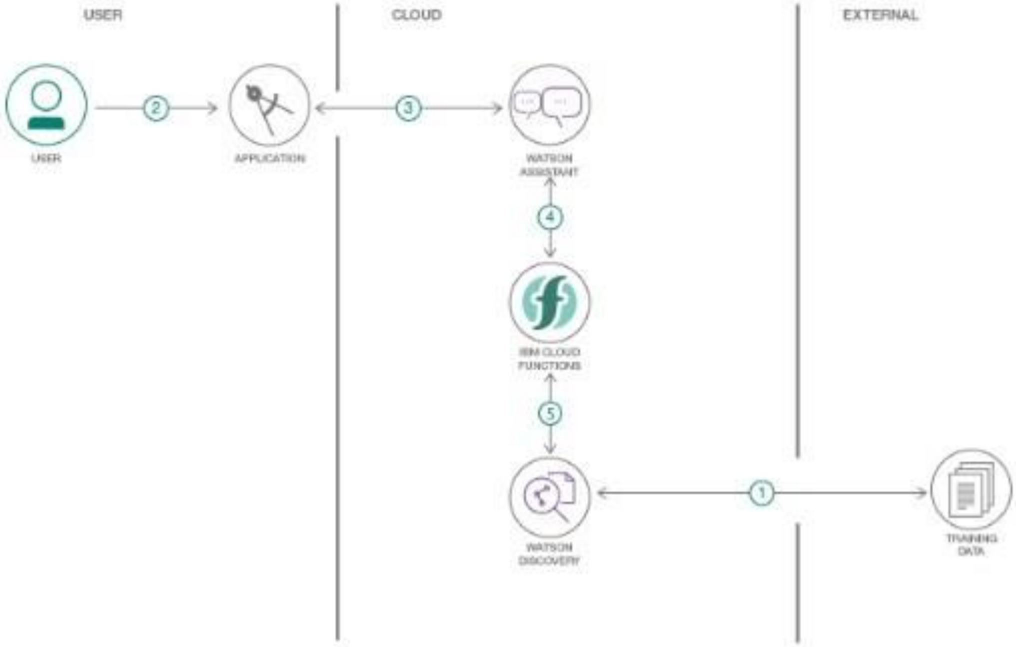
At first go to manage pallete and install dashboard. Now,Create the flow with the help of following node:

- Assistant
- Debug
- Function
- Ui_Form
- Ui_Text



| | |
|---|---|
| Flow | "d20d4833.fce068" |
| Name | Flow 1 |
| Status | Enabled |

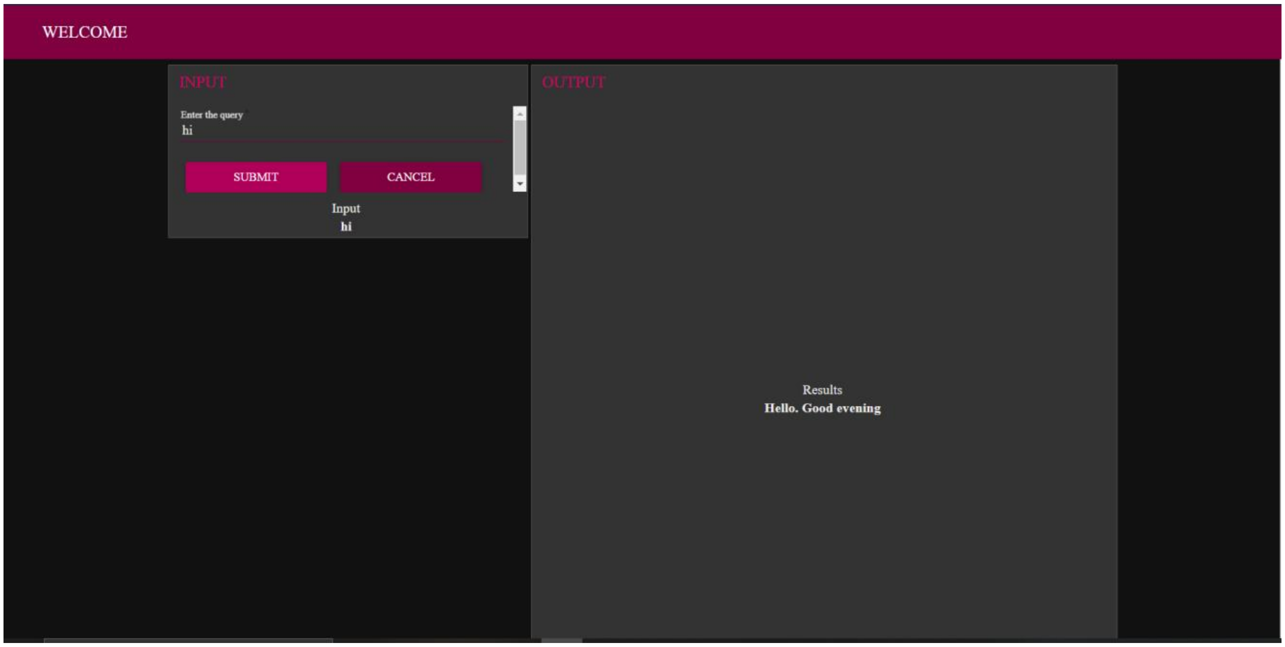
⇧ click and drag on a node port to move all of the attached wires or just the selected one


USER
CLOUD
EXTERNAL
USER
2
APPLICATION
3
WATSON ASSISTANT
4
IBM CLOUD FUNCTIONS
5
WATSON DISCOVERY
1
TRAINING DATA

# 6. RESULTS

Finally our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing the URL in the browser –

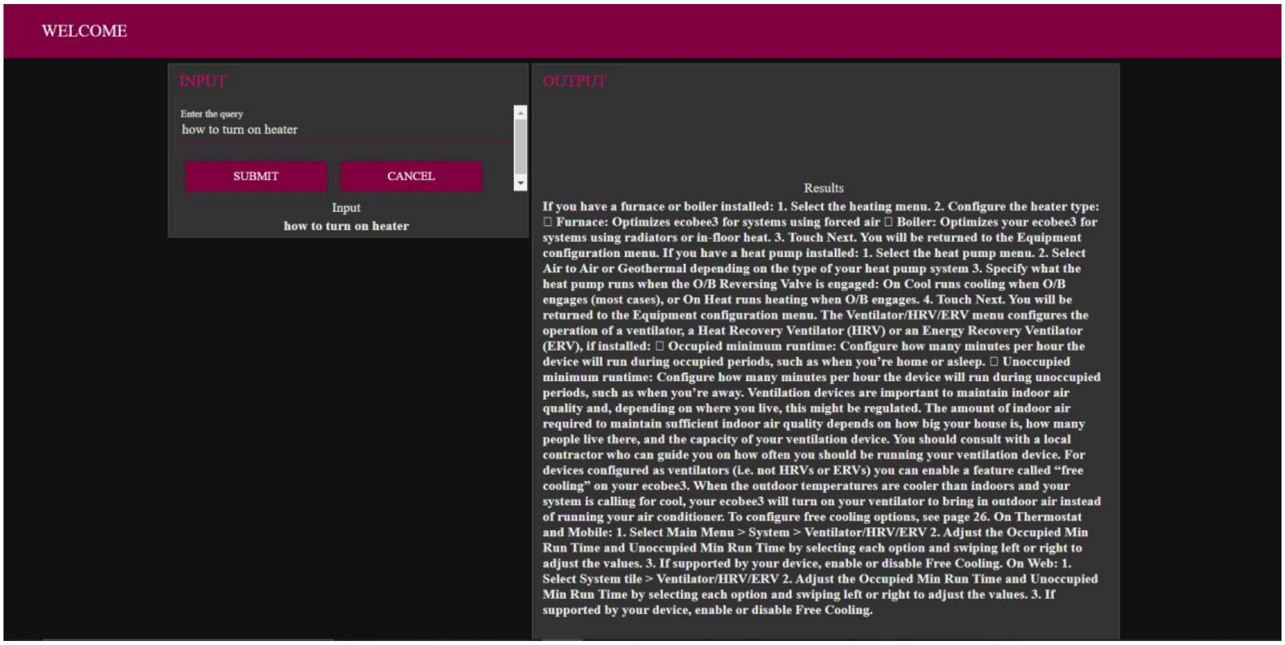
https://node-red-avav.eu-gb.mybluemix.net/ui



Enter the query
hi

Input
**hi**

Results
**Hello. Good evening**



Enter the query
how to turn on heater

Input
**how to turn on heater**

Results
**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. If you have a heat pump installed: 1. Select the heat pump menu. 2. Select Air to Air or Geothermal depending on the type of your heat pump system 3. Specify what the heat pump runs when the O/B Reversing Valve is engaged: On Cool runs cooling when O/B engages (most cases), or On Heat runs heating when O/B engages. 4. Touch Next. You will be returned to the Equipment configuration menu. The Ventilator/HRV/ERV menu configures the operation of a ventilator, a Heat Recovery Ventilator (HRV) or an Energy Recovery Ventilator (ERV), if installed: □ Occupied minimum runtime: Configure how many minutes per hour the device will run during occupied periods, such as when you're home or asleep. □ Unoccupied minimum runtime: Configure how many minutes per hour the device will run during unoccupied periods, such as when you're away. Ventilation devices are important to maintain indoor air quality and, depending on where you live, this might be regulated. The amount of indoor air required to maintain sufficient indoor air quality depends on how big your house is, how many people live there, and the capacity of your ventilation device. You should consult with a local contractor who can guide you on how often you should be running your ventilation device. For devices configured as ventilators (i.e. not HRVs or ERVs) you can enable a feature called "free cooling" on your ecobee3. When the outdoor temperatures are cooler than indoors and your system is calling for cool, your ecobee3 will turn on your ventilator to bring in outdoor air instead of running your air conditioner. To configure free cooling options, see page 26. On Thermostat and Mobile: 1. Select Main Menu > System > Ventilator/HRV/ERV 2. Adjust the Occupied Min Run Time and Unoccupied Min Run Time by selecting each option and swiping left or right to adjust the values. 3. If supported by your device, enable or disable Free Cooling. On Web: 1. Select System tile > Ventilator/HRV/ERV 2. Adjust the Occupied Min Run Time and Unoccupied Min Run Time by selecting each option and swiping left or right to adjust the values. 3. If supported by your device, enable or disable Free Cooling.**

# 7. ADVANTAGES & DISADVANTAGES

**Advantages:**

- **Reduced costs:** Chatbots eliminate the need for labor during online interaction with customers. This is obviously a great advantage for companies that receive multiple queries at once. In addition to saving costs with them, companies can align the chatbot with their objectives, and use them as a means to enhance customer conversion.
- **24/7 Availability:** Unlike humans, once we install a chatbot, it can handle queries at any time of day. Thus, the customer does not have to wait for a commercial of the company to help him. This also allows companies to monitor customer « traffic » during non-working hours and contact them later.
- **Learning and updating:** AI-based chatbots are able to learn from interactions and update independently. This is one of the main advantages. When you hire a new employee, you have to train them continuously. However, chatbots « form » themselves (with certain limitations, of course).
- **Management of multiple clients:** Humans can serve a limited number of customers at the same time. This restriction does not exist for chatbots, and they can manage all the necessary queries simultaneously. This is one of the main advantages of using chatbot, as no customer is left unattended and you are solving different problems at the same time. There are chatbots companies already working on developing voice chatbot services.

**Disadvantages:**

- **Complex interface:** It is often considered that chatbots are complicated and need a lot of time to understand what you want in customer. Sometimes, it can also annoy the client about their slowness, or their difficulty in filtering responses.
- **They don't get you right:** Fixed chatbots can get stuck easily. If a query doesn't relate to something you've previously « taught » it, you won't understand it. This can lead to a frustrated customer and the loss of the sale. Other times they do understand you, but they need double (or triple) as many messages as one person, which spoils the user experience.
- **Time-consuming:** Chatbots are installed with the aim of speeding up responses and improving customer interaction. However, due to the limited availability of data and the time needed for self-updating, this process can be slow and costly. Therefore, there are times when instead of serving several customers at once, chatbots may become confused and not serve the customer well.
- **Installation cost:** Chatbots are useful programs that help you save a lot of labor by ensuring availability at all times and serving several customers at once. But unlike humans, each chatbot needs to be programmed differently for each business, which increases the initial installation cost. Considering the last-minute changes that can always occur, this is a risky investment, as updating the program will generate additional costs.
- **Null decision making:** Chatbots can attack the nerves of more than one because they are not able to make decisions.

## 8. APPLICATIONS

Following are the examples of ways one can use chatbots :

- Order Pizza
- Product Suggestions
- Customer Support
- Weather
- Personal Finance Assistance
- Schedule a Meeting
- Search for and Track Flights
- News
- Send Money
- Find a Restaurant

# 9. CONCLUSION

By following the above procedure, Intelligent Customer Help Desk with Smart Document Understanding using Watson assistant, Watson discovery, Node-RED and cloud-functions is successfully built. Chatbots are quickly making transformational changes and allowing businesses to thrive through customer interactions. The feedback and survey through chatbots strengthen the position of businesses as they analyze the reason behind different levels of customer approval. Use of conversational AI chatbots only means better engagement and relentless need for customer satisfaction in the near future.

# 10. FUTURE SCOPE

We can include watson studio text to speech and speech to text services to access the chatbot using voice instead of writing text and creating a more user friendly UI with additional interactive features.

# 11. BIBILOGRAPHY

# APPENDIX

## A. Source Code

### Watson Assistant : skill-Customer-Care-Sample-Skill.json

```
{
         "intents": [
           {
             "intent": "Product_Information",
             "examples": [
               {
                 "text": "How to turn on the heater"
               },
               {
                 "text": "How to use thermostat"
               }
             ],
             "description": "User wants help using the thermostat"
           },
           {
             "intent": "Cancel",
             "examples": [
               {
                 "text": "i don't want a table anymore anymore"
               },
               {
                 "text": "cancel the request"
               },
               {
                 "text": "forget it"
               },
               {
                 "text": "cancel that"
               },
               {
                 "text": "never mind"
               },
               {
                 "text": "i changed my mind"
               },
               {
                 "text": "nevermind"
               }
```

```
      ],
      "description": "Cancel the current request"
    },
    {
      "intent": "Thanks",
      "examples": [
        {
          "text": "much appreciated"
        },
        {
          "text": "thank you"
        },
        {
          "text": "i appreciate it"
        },
        {
          "text": "thx"
        },
        {
          "text": "thank you very much"
        },
        {
          "text": "that's nice of you"
        },
        {
          "text": "many thanks"
        },
        {
          "text": "thanks"
        }
      ],
      "description": "Thanks"
    },
    {
      "intent": "General_Greetings",
      "examples": [
        {
          "text": "hiya"
        },
        {
          "text": "yo"
        },
        {
          "text": "How are things going?"
        },
        {
```

```
  "text": "How are you today?"
},
{
  "text": "How have you been?"
},
{
  "text": "hi"
},
{
  "text": "How r u?"
},
{
  "text": "Looking good eve"
},
{
  "text": "Hey you"
},
{
  "text": "How is it going?"
},
{
  "text": "You there"
},
{
  "text": "Who is this?"
},
{
  "text": "What's up?"
},
{
  "text": "Good day"
},
{
  "text": "What's new?"
},
{
  "text": "Hi there"
},
{
  "text": "Hey twin"
},
{
  "text": "Hi advisor"
},
{
  "text": "Ok take me back"
```

```
        },
        {
          "text": "Hey there"
        },
        {
          "text": "Hey there all"
        },
        {
          "text": "Hey how are you doing"
        },
        {
          "text": "Hello I am looking for some help here"
        },
        {
          "text": "Hello"
        },
        {
          "text": "Hello Agent"
        },
        {
          "text": "Have you been well?"
        },
        {
          "text": "Greetings"
        },
        {
          "text": "Good to see you"
        },
        {
          "text": "Good morning"
        },
        {
          "text": "Good evening"
        }
      ],
      "description": "Greetings"
    },
    {
      "intent": "Goodbye",
      "examples": [
        {
          "text": "arrivederci"
        },
        {
          "text": "ciao"
        },
    {
      "text": "bye"
    },
    {
      "text": "so long"
    },
    {
      "text": "good bye"
    },
    {
      "text": "see ya"
    }
  ],
  "description": "Good byes"
},
{
  "intent": "Customer_Care_Store_Location",
  "examples": [
    {
      "text": "Looking for a location"
    },
    {
      "text": "What is the closest store to my address?"
    },
    {
      "text": "What is the nearest branch?"
    },
    {
      "text": "What is the store near my zip code?"
    },
    {
      "text": "Where is?"
    },
    {
      "text": "Find store"
    },
    {
      "text": "how do i get to your place"
    },
    {
      "text": "where are you located"
    },
    {
      "text": "can you give me directions"
    },
    {
```

```
  "text": "location please"
},
{
  "text": "how do i find you"
},
{
  "text": "what is the address"
},
{
  "text": "where are you"
},
{
  "text": "what's your location"
},
{
  "text": "give me directions"
},
{
  "text": "which cross streets are you on"
},
{
  "text": "how can i get to you from grand central"
},
{
  "text": "please suggest route from times square"
},
{
  "text": "Where are you located?"
},
{
  "text": "Where is your office?"
},
{
  "text": "how do i get to your business"
},
{
  "text": "Go to your company"
},
{
  "text": "I'd like to go to a store"
},
{
  "text": "I need help with find a store"
},
{
  "text": "I want to know about a store"
```

```
    }
  ],
  "description": "Locate a physical store location or an address."
},
{
  "intent": "General_Connect_to_Agent",
  "examples": [
    {
      "text": "How can I skip the recorded menu and go straight to a live person?"
    },
    {
      "text": "I dont want to talk to a computer"
    },
    {
      "text": "call the manager"
    },
    {
      "text": "I want to speak to a human"
    },
    {
      "text": "I want to talk to the manager"
    },
    {
      "text": "A real agent, please."
    },
    {
      "text": "Call agent"
    },
    {
      "text": "talk to a human"
    },
    {
      "text": "Agent help"
    },
    {
      "text": "Yes, take me to a real person"
    },
    {
      "text": "Where is the closest agent?"
    },
    {
      "text": "Send me to an agent"
    },
    {
      "text": "I don't want to speak with a robot"
    },
{
  "text": "get me a person"
},
{
  "text": "Can I connect to an agent?"
},
{
  "text": "Can I speak to a human please?"
},
{
  "text": "Can I speak to a live person?"
},
{
  "text": "Can I speak to an advisor?"
},
{
  "text": "Can I speak with somebody?"
},
{
  "text": "Can I talk to someone?"
},
{
  "text": "Can you assist me to connect to an agent?"
},
{
  "text": "Can you connect me with a real person?"
},
{
  "text": "Connect me to a live operator please."
},
{
  "text": "Contact person"
},
{
  "text": "Could you please transfer me to your master?"
},
{
  "text": "Customer service representative please."
},
{
  "text": "Do not want a robot?"
},
{
  "text": "Hi can you transfer me"
},
{
```

```
          "text": "I don't want to talk to a bot."
        },
        {
          "text": "I don't want to talk to you"
        },
        {
          "text": "Is there anyone there I can actually talk to for real?"
        },
        {
          "text": "I need to speak to a representative. How would I go about doing
so?"
        },
        {
          "text": "I want agent"
        },
        {
          "text": "I want a manager"
        },
        {
          "text": "I want an agent to help me"
        },
        {
          "text": "I want to talk to a person"
        },
        {
          "text": "I would like to speak to a human"
        },
        {
          "text": "I would like to speak to someone"
        },
        {
          "text": "Need help from human"
        },
        {
          "text": "Please assist me to get to an agent"
        },
        {
          "text": "Operator please"
        },
        {
          "text": "Please connect me to a live agent"
        },
        {
          "text": "Please let me talk to a human being."
        },
        {
          "text": "Pls connect"
        },
        {
          "text": "Put me through to someone"
        },
        {
          "text": "representative"
        },
        {
          "text": "I want to speak to a person"
        }
      ],
      "description": "Request a human agent."
    },
    {
      "intent": "Customer_Care_Store_Hours",
      "examples": [
        {
          "text": "will you open on christmas"
        },
        {
          "text": "how late y'all stay up till"
        },
        {
          "text": "how late are you there"
        },
        {
          "text": "how early do you open on Saturdays"
        },
        {
          "text": "Are you closed new Year's eve"
        },
        {
          "text": "Are you closing early today"
        },
        {
          "text": "What is the opening time for the washington store?"
        },
        {
          "text": "store open hours?"
        },
        {
          "text": "store open"
        },
        {
          "text": "store hours"
```

```
},
{
  "text": "store hrs"
},
{
  "text": "when can i visit your store"
},
{
  "text": "when does the store close"
},
{
  "text": "when do your stores open"
},
{
  "text": "open hours store"
},
{
  "text": "store open now"
},
{
  "text": "Hours of operation"
},
{
  "text": "What time do you close today"
},
{
  "text": "what time do you close on Sunday"
},
{
  "text": "What time do you open on Saturdays"
},
{
  "text": "What time do you close"
},
{
  "text": "when do you close"
},
{
  "text": "will you be open Memorial day"
},
{
  "text": "will you open for christmas"
},
{
  "text": "what are your hours"
},
{
  "text": "What time is your store open on saturday?"
},
{
  "text": "What time do stores close?"
},
{
  "text": "What time does the central manchester store shut on a saturday?"
},
{
  "text": "What time are you closing today?"
},
{
  "text": "What are your hous?"
},
{
  "text": "What are ur opening hours?"
},
{
  "text": "What are the saturday opening times for the local store?"
},
{
  "text": "What are the hours of operation?"
},
{
  "text": "What are the business hours of the store nearest to me?"
},
{
  "text": "Is the branch open now?"
},
{
  "text": "How long are you open?"
},
{
  "text": "How early do you open?"
},
{
  "text": "Does the store in the city center opens till 8pm on weekends?"
},
{
  "text": "Can you tell me how late the stores are open till?"
},
{
  "text": "At what hour can I swing by?"
},
{
```

```
          "text": "Are you open on sundays, and if so what are the hours?"
        },
        {
          "text": "Are you open on bank holidays?"
        },
        {
          "text": "Are you open during thanksgiving?"
        },
        {
          "text": "Are the stores open early?"
        },
        {
          "text": "are stores open on sunday"
        },
        {
          "text": "how late are you open tonight"
        },
        {
          "text": "how late are you open"
        },
        {
          "text": "Are you open on Sunday"
        }
      ],
      "description": "Find business hours."
    },
    {
      "intent": "Help",
      "examples": [
        {
          "text": "can you help"
        },
        {
          "text": "i need assistance"
        },
        {
          "text": "what can i do"
        },
        {
          "text": "what can i say"
        },
        {
          "text": "help me"
        },
        {
          "text": "help me decide"
        },
        {
          "text": "help"
        },
        {
          "text": "can you assist me"
        }
      ],
      "description": "Ask for help"
    },
    {
      "intent": "Customer_Care_Appointments",
      "examples": [
        {
          "text": "I would like to make an appointment to visit the nearest store to
my location."
        },
        {
          "text": "I would like to discuss my situation face to face"
        },
        {
          "text": "I want to talk in person with someone about my case"
        },
        {
          "text": "meet in store"
        },
        {
          "text": "are you available on tuesday"
        },
        {
          "text": "can i book for tonight"
        },
        {
          "text": "do you have availability next week"
        },
        {
          "text": "can i make an appointment"
        },
        {
          "text": "can you make an appointment for me"
        },
        {
          "text": "i'd like to make an appointment"
        },
        {
          "text": "What time can I meet the staff?"
        },
        {
          "text": "Want to change my visit"
        },
        {
          "text": "When can I meet with one of your employees at your store?"
        },
        {
          "text": "Store appointment"
        },
        {
          "text": "Set up an appt"
        },
        {
          "text": "Make an appointment"
        },
        {
          "text": "i'd like to come in for an appointment"
        },
        {
          "text": "I prefer a face to face visit"
        },
        {
          "text": "Can I book an in person session"
        },
        {
          "text": "Could I speak to someone in the store next tuesday?"
        }
      ],
      "description": "Schedule or manage an in-store appointment."
    }
  ],
  "entities": [
    {
      "entity": "specialist",
      "values": [
        {
          "type": "synonyms",
          "value": "Maria",
          "synonyms": []
        },
        {
          "type": "synonyms",
          "value": "Derrik",
          "synonyms": [
            "derek",
        "derik",
        "derrik",
        "derrick"
      ]
    },
    {
      "type": "synonyms",
      "value": "Brenda",
      "synonyms": []
    },
    {
      "type": "synonyms",
      "value": "Barbara",
      "synonyms": [
        "barbra"
      ]
    },
    {
      "type": "synonyms",
      "value": "Nicholas",
      "synonyms": [
        "nick"
      ]
    },
    {
      "type": "synonyms",
      "value": "Robert",
      "synonyms": [
        "bob"
      ]
    }
  ]
},
{
  "entity": "holiday",
  "values": [
    {
      "type": "synonyms",
      "value": "new years eve",
      "synonyms": [
        "12-31",
        "12/31",
        "dec 31",
        "dec 31st",
        "new year's eve"
      ]
```

```
},
{
  "type": "synonyms",
  "value": "christmas eve",
  "synonyms": [
    "x mas eve",
    "x-mas eve",
    "xmas eve"
  ]
},
{
  "type": "synonyms",
  "value": "labor day",
  "synonyms": []
},
{
  "type": "synonyms",
  "value": "valentine's day",
  "synonyms": [
    "valentine day",
    "valentines day"
  ]
},
{
  "type": "synonyms",
  "value": "independence day",
  "synonyms": [
    "7/4",
    "fourth of july",
    "july 4",
    "july 4th",
    "july fourth"
  ]
},
{
  "type": "synonyms",
  "value": "halloween",
  "synonyms": []
},
{
  "type": "synonyms",
  "value": "christmas",
  "synonyms": [
    "christmas day",
    "x man day",
    "xmas",
            "x mas",
            "x-mas",
            "x-mas day",
            "xmas day"
          ]
        },
        {
          "type": "synonyms",
          "value": "thanksgiving",
          "synonyms": [
            "turkey day"
          ]
        },
        {
          "type": "synonyms",
          "value": "memorial day",
          "synonyms": []
        },
        {
          "type": "synonyms",
          "value": "new years",
          "synonyms": [
            "1/1",
            "jan 1",
            "jan 1st",
            "jan first",
            "january 1",
            "january 1st",
            "january first",
            "new year",
            "new year day",
            "new years day"
          ]
        }
      ]
    },
    {
      "entity": "phone",
      "values": [
        {
          "type": "patterns",
          "value": "US Phone pattern",
          "patterns": [
            "(\\d{3})-(\\d{3})-(\\d{4})"
          ]
        }
```

```
  ]
},
{
  "entity": "sys-date",
  "values": []
},
{
  "entity": "sys-number",
  "values": []
},
{
  "entity": "sys-time",
  "values": []
},
{
  "entity": "reply",
  "values": [
    {
      "type": "synonyms",
      "value": "no",
      "synonyms": [
        "definitely not",
        "don't think so",
        "dont think so",
        "i think not",
        "nope",
        "not at this time",
        "not now"
      ]
    },
    {
      "type": "synonyms",
      "value": "yes",
      "synonyms": [
        "definitely",
        "go for it",
        "let's do it",
        "ok",
        "please",
        "sure",
        "why not",
        "yeah",
        "yes",
        "you bet",
        "you betcha",
        "yep"
```

```
          ]
        }
      ]
    },
    {
      "entity": "zip_code",
      "values": [
        {
          "type": "patterns",
          "value": "US Zip",
          "patterns": [
            "(\\b|\\s)\\d{5}(\\b|\\s)"
          ]
        }
      ]
    },
    {
      "entity": "landmark",
      "values": [
        {
          "type": "synonyms",
          "value": "grand central",
          "synonyms": []
        },
        {
          "type": "synonyms",
          "value": "times square",
          "synonyms": [
            "time sqaure",
            "time square",
            "times sqaure"
          ]
        },
        {
          "type": "synonyms",
          "value": "empire state building",
          "synonyms": [
            "empire state",
            "emprire state"
          ]
        }
      ],
      "fuzzy_match": true
    }
  ],
  "metadata": {
    "api_version": {
      "major_version": "v2",
      "minor_version": "2018-11-08"
    }
  },
  "webhooks": [
    {
      "url": "https://eu-
gb.functions.cloud.ibm.com/api/v1/web/vermaanjali2000%40gmail.com_dev/default/MyAction
.json",
      "name": "main_webhook",
      "headers": []
    }
  ],
  "dialog_nodes": [
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "To get to our business from Grand Central, take the 4,5 or 6 train
downtown to Union Square."
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "@landmark:(grand central)",
      "dialog_node": "node_4_1522439442155",
      "previous_sibling": "node_8_1482459217052"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "To get to our business from the Empire State Building, walk to Herald
Square and take the N train to Union Square"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "@landmark:(empire state building)",
      "dialog_node": "node_7_1482459200886",
      "previous_sibling": "node_3_1522439390442"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "To get to our business from Times Square, take the N train downtown to
Union Square"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "@landmark:(times square)",
      "dialog_node": "node_8_1482459217052",
      "previous_sibling": "node_7_1482459200886"
    },
    {
      "type": "standard",
      "title": "Provide location",
      "output": {
        "text": {
          "values": [
            "We're located by Union Square on the corner of 13th and Broadway"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "true",
      "dialog_node": "node_3_1522439390442"
    },
    {
      "type": "event_handler",
      "output": {
        "text": {
          "values": [
            "We only accept appointments between 11am and 5pm"
          ]
        }
      },
      "parent": "slot_105_1498132552870",
      "metadata": {},
      "next_step": {
        "behavior": "reprompt"
      },
      "conditions": "$time.after('17:30:30') || $time.before('10:59:59')",
      "event_name": "filled",
      "dialog_node": "handler_1_1509694458589",
      "previous_sibling": "handler_106_1498132552870"
    },
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_105_1498132552870",
      "context": {
        "time": "@sys-time"
      },
      "metadata": {},
      "conditions": "@sys-time",
      "event_name": "input",
      "dialog_node": "handler_106_1498132552870",
      "previous_sibling": "handler_107_1498132552870"
    },
    {
      "type": "event_handler",
      "output": {
        "text": "What time on <? $date.reformatDateTime(\"EEEEE\") ?> do you want to
come in?"
      },
      "parent": "slot_105_1498132552870",
      "metadata": {},
      "event_name": "focus",
      "dialog_node": "handler_107_1498132552870"
    },
    {
      "type": "slot",
      "output": {},
      "parent": "Reservation using slots",
      "metadata": {},
      "variable": "$specialist",
      "dialog_node": "slot_12_1522596437268",
      "previous_sibling": "slot_105_1498132552870"
    },
    {
      "type": "slot",
      "output": {},
      "parent": "Reservation using slots",
  "metadata": {
    "_customization": {}
  },
  "variable": "$phone",
  "dialog_node": "slot_22_1522444583114",
  "previous_sibling": "slot_8_1509132875735"
},
{
  "type": "event_handler",
  "output": {
    "text": {
      "values": [
        "OK. Canceling your request..."
      ]
    }
  },
  "parent": "Reservation using slots",
  "context": {
    "date": null,
    "time": null,
    "phone": null,
    "confirm": null,
    "specialist": null,
    "user_cancelled": true
  },
  "metadata": {},
  "next_step": {
    "behavior": "skip_all_slots"
  },
  "conditions": "#Cancel",
  "event_name": "generic",
  "dialog_node": "handler_16_1509133697261",
  "previous_sibling": "handler_3_1501275087289"
},
{
  "type": "event_handler",
  "output": {
    "text": {
      "values": []
    }
  },
  "parent": "Reservation using slots",
  "disabled": true,
  "metadata": {},
  "event_name": "focus",
  "dialog_node": "handler_7_1509696539866",
      "previous_sibling": "handler_16_1509133697261"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "Let me know how else I can help"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Reservation using slots",
      "context": {},
      "metadata": {},
      "conditions": "$user_cancelled",
      "dialog_node": "node_10_1509697567474",
      "previous_sibling": "node_25_1522598839584"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "Let me check availability...  [Use IBM Cloud Functions to connect to
backend systems]"
          ]
        }
      },
      "parent": "Reservation using slots",
      "context": {},
      "metadata": {},
      "conditions": "true",
      "dialog_node": "node_3_1519173961259",
      "previous_sibling": "node_10_1509697567474"
    },
    {
      "type": "slot",
      "output": {},
      "parent": "Reservation using slots",
      "metadata": {
        "_customization": {
          "mcr": true
        }
      },
      "variable": "$date",
  "dialog_node": "slot_102_1498132501942",
  "previous_sibling": "node_3_1519173961259"
},
{
  "type": "slot",
  "output": {},
  "parent": "Reservation using slots",
  "metadata": {
    "_customization": {
      "mcr": true
    }
  },
  "variable": "$time",
  "dialog_node": "slot_105_1498132552870",
  "previous_sibling": "slot_102_1498132501942"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "[Use IBM Cloud Functions to connect to to backend systems]"
      ]
    }
  },
  "parent": "Reservation using slots",
  "metadata": {},
  "conditions": "$user_needs_help",
  "dialog_node": "node_25_1522598839584",
  "previous_sibling": "handler_7_1509696539866"
},
{
  "type": "slot",
  "output": {},
  "parent": "Reservation using slots",
  "metadata": {},
  "variable": "$confirm",
  "dialog_node": "slot_8_1509132875735",
  "previous_sibling": "slot_12_1522596437268"
},
{
  "type": "event_handler",
  "output": {
    "text": {
      "values": [
            "I see you need help making an appointment. Let me transfer you to an
agent..."
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Reservation using slots",
      "context": {
        "date": null,
        "time": null,
        "phone": null,
        "confirm": null,
        "specialist": null,
        "user_needs_help": true
      },
      "metadata": {},
      "next_step": {
        "behavior": "skip_all_slots"
      },
      "conditions": "#Help",
      "event_name": "generic",
      "dialog_node": "handler_3_1501275087289"
    },
    {
      "type": "event_handler",
      "output": {
        "text": {
          "values": [
            "Perfect!"
          ]
        }
      },
      "parent": "slot_8_1509132875735",
      "metadata": {},
      "conditions": "@reply:yes",
      "event_name": "filled",
      "dialog_node": "handler_14_1509133469904",
      "previous_sibling": "handler_9_1509132875735"
    },
    {
      "type": "event_handler",
      "output": {
        "text": {
          "values": [
            "Sorry... let's try again"
          ]
        }
      },
      "parent": "slot_8_1509132875735",
      "context": {
        "date": null,
        "time": null,
        "confirm": null
      },
      "metadata": {},
      "conditions": "@reply:no",
      "event_name": "filled",
      "dialog_node": "handler_17_1509135162089",
      "previous_sibling": "handler_14_1509133469904"
    },
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_8_1509132875735",
      "context": {
        "confirm": "@reply && slot_in_focus"
      },
      "metadata": {},
      "conditions": "@reply && slot_in_focus",
      "event_name": "input",
      "dialog_node": "handler_9_1509132875735",
      "previous_sibling": "handler_10_1509132875735"
    },
    {
      "type": "event_handler",
      "output": {
        "text": "Let me confirm: You want an appointment for <?
$date.reformatDateTime(\"EEEEE\") ?> at <? $time.reformatDateTime(\"h a\") ?>. Is this
correct?"
      },
      "parent": "slot_8_1509132875735",
      "metadata": {},
      "event_name": "focus",
      "dialog_node": "handler_10_1509132875735"
    },
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_102_1498132501942",
      "context": {
        "date": "@sys-date"
      },
  "metadata": {},
  "conditions": "@sys-date",
  "event_name": "input",
  "dialog_node": "handler_103_1498132501942",
  "previous_sibling": "handler_104_1498132501942"
},
{
  "type": "event_handler",
  "output": {
    "text": {
      "values": [
        "Looks like you're trying to make a reservation in the past. Try again."
      ]
    }
  },
  "parent": "slot_102_1498132501942",
  "metadata": {},
  "next_step": {
    "behavior": "reprompt"
  },
  "conditions": "$date.before(now())",
  "event_name": "filled",
  "dialog_node": "handler_6_1509695999145",
  "previous_sibling": "handler_103_1498132501942"
},
{
  "type": "event_handler",
  "output": {
    "text": "What day would you like to come in?"
  },
  "parent": "slot_102_1498132501942",
  "metadata": {},
  "event_name": "focus",
  "dialog_node": "handler_104_1498132501942"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Hello",
        "Hi there",
        "Hi. How can I help"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_13_1502484041694",
  "metadata": {},
  "dialog_node": "node_28_1522448362216",
  "previous_sibling": "node_15_1488295465298"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Hello. Good evening",
        "Hi. Good evening",
        "Hello. How can I help this evening?"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_13_1502484041694",
  "metadata": {},
  "conditions": "now().after('17:00:00')",
  "dialog_node": "node_15_1488295465298",
  "previous_sibling": "node_1_1495022305143"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Hello. Good afternoon",
        "Hi there. It's a beautiful afternoon",
        "Good afternoon. How can I help?"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_13_1502484041694",
  "metadata": {},
  "conditions": "now().after('12:00:00') && now().before('16:59:59')",
  "dialog_node": "node_1_1495022305143",
  "previous_sibling": "node_16_1488295517679"
},
{
  "type": "response_condition",
  "output": {
    "text": {
          "values": [
            "Hello. Good morning",
            "It's a beautiful morning. Hello",
            "Hi there. How can I help you this morning?"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "node_13_1502484041694",
      "metadata": {},
      "conditions": "now().after('04:00:00') && now().before('11:59:59')",
      "dialog_node": "node_16_1488295517679"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "Our hours on <? @sys-date.reformatDateTime(\"EEEEE\") ?> are 11am to
6pm."
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Hours of Operation",
      "context": {},
      "metadata": {},
      "conditions": "@sys-date.reformatDateTime(\"EEEEE\") == \"Saturday\" || @sys-
date.reformatDateTime(\"EEEEE\") == \"Sunday\"",
      "dialog_node": "node_2_1482424204936",
      "previous_sibling": "node_5_1482426503106"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "We are open on @holiday regular hours"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Hours of Operation",
      "context": {},
      "metadata": {},
      "conditions": "@holiday",
      "dialog_node": "node_5_1482426503106",
      "previous_sibling": "node_1_1522387330204"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "We are open on <? @sys-date.reformatDateTime(\"EEEEE\") ?> from 10am
until 8pm"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Hours of Operation",
      "context": {},
      "metadata": {},
      "conditions": "@sys-date.reformatDateTime(\"EEEEE\") == \"Monday\" || @sys-
date.reformatDateTime(\"EEEEE\") == \"Tuesday\" || @sys-
date.reformatDateTime(\"EEEEE\") == \"Wednesday\" || @sys-
date.reformatDateTime(\"EEEEE\") == \"Thursday\" || @sys-
date.reformatDateTime(\"EEEEE\") == \"Friday\"",
      "dialog_node": "node_1_1522387330204",
      "previous_sibling": "node_4_1482425833988"
    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "Our hours are Monday to Friday 10am to 8pm and Friday and Saturday 11am
to 6pm."
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Hours of Operation",
      "context": {},
      "metadata": {},
      "conditions": " true",
      "dialog_node": "node_6_1482426521282",
      "previous_sibling": "node_2_1482424204936"
    },
    {
      "type": "response_condition",
      "output": {
```

```
    "text": {
     "values": [
      "We are closed on @holiday"
     ],
     "selection_policy": "sequential"
    }
   },
   "parent": "Hours of Operation",
   "context": {},
   "metadata": {},
   "conditions": "@holiday:christmas || @holiday:thanksgiving || @holiday:(new
years)",
   "dialog_node": "node_4_1482425833988"
  },
  {
   "type": "event_handler",
   "output": {},
   "parent": "slot_22_1522444583114",
   "context": {
    "phone": "@phone"
   },
   "metadata": {},
   "conditions": "@phone",
   "event_name": "input",
   "dialog_node": "handler_23_1522444583114",
   "previous_sibling": "handler_24_1522444583114"
  },
  {
   "type": "event_handler",
   "output": {
    "text": {
     "values": [
      "Thanks"
     ],
     "selection_policy": "sequential"
    }
   },
   "parent": "slot_22_1522444583114",
   "context": {},
   "metadata": {},
   "conditions": "true",
   "event_name": "filled",
   "dialog_node": "handler_22_1522598191131",
   "previous_sibling": "handler_23_1522444583114"
  },
  {
      "type": "event_handler",
      "output": {
        "text": "I'll just need a phone to hold your reservation"
      },
      "parent": "slot_22_1522444583114",
      "metadata": {},
      "event_name": "focus",
      "dialog_node": "handler_24_1522444583114"
    },
    {
      "type": "response_condition",
      "output": {
        "generic": [
          {
            "values": [
              {
                "text": "Try again later"
              }
            ],
            "response_type": "text",
            "selection_policy": "sequential"
          }
        ]
      },
      "parent": "node_2_1589752166191",
      "conditions": "anything_else",
      "dialog_node": "response_5_1589752214517",
      "previous_sibling": "response_1_1589752211979"
    },
    {
      "type": "response_condition",
      "output": {
        "generic": [
          {
            "values": [
              {
                "text": "$webhook_result_1"
              }
            ],
            "response_type": "text",
            "selection_policy": "sequential"
          }
        ]
      },
      "parent": "node_2_1589752166191",
      "conditions": "$webhook_result_1",
   "dialog_node": "response_1_1589752211979"
  },
  {
   "type": "standard",
   "output": {
    "text": "OK. Let me know how I can help"
   },
   "parent": "node_22_1467833484410",
   "metadata": {},
   "conditions": "@reply:no",
   "dialog_node": "node_21_1468350173406",
   "previous_sibling": "node_19_1468350024009"
  },
  {
   "type": "standard",
   "output": {
    "text": {
     "values": [
      "OK. Transferring... [Use IBM Cloud Functions to connect to backend
systems]"
     ]
    }
   },
   "parent": "node_22_1467833484410",
   "metadata": {},
   "conditions": "@reply:yes",
   "dialog_node": "node_19_1468350024009"
  },
  {
   "type": "event_handler",
   "output": {},
   "parent": "slot_12_1522596437268",
   "context": {
    "specialist": "@specialist"
   },
   "metadata": {},
   "conditions": "@specialist",
   "event_name": "input",
   "dialog_node": "handler_13_1522596437268",
   "previous_sibling": "handler_14_1522596437268"
  },
  {
   "type": "event_handler",
   "output": {
    "text": {
     "values": [
        "We'll do our best to book you with @specialist"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "slot_12_1522596437268",
  "event_name": "filled",
  "dialog_node": "handler_15_1522596463593",
  "previous_sibling": "handler_13_1522596437268"
},
{
  "type": "event_handler",
  "output": {},
  "parent": "slot_12_1522596437268",
  "event_name": "focus",
  "dialog_node": "handler_14_1522596437268"
},
{
  "type": "frame",
  "title": "I want to make an appointment",
  "output": {},
  "metadata": {
    "fallback": "leave",
    "_customization": {
      "mcr": true
    }
  },
  "conditions": "#Customer_Care_Appointments",
  "digress_in": "does_not_return",
  "dialog_node": "Reservation using slots",
  "digress_out": "allow_all",
  "previous_sibling": "Directions",
  "digress_out_slots": "allow_all"
},
{
  "type": "standard",
  "output": {},
  "metadata": {},
  "conditions": "#General_Greetings",
  "digress_in": "does_not_return",
  "dialog_node": "node_13_1502484041694",
  "previous_sibling": "Reservation using slots"
},
{
  "type": "standard",
  "title": "Please transfer me to an agent",
      "output": {
        "text": {
          "values": [
            "Would you like me to transfer you to a representative?"
          ],
          "selection_policy": "sequential"
        }
      },
      "metadata": {},
      "conditions": "#General_Connect_to_Agent",
      "digress_in": "does_not_return",
      "dialog_node": "node_22_1467833484410",
      "digress_out": "allow_all_never_return",
      "previous_sibling": "node_2_1468243505617"
    },
    {
      "type": "standard",
      "title": "Ask about product",
      "actions": [
        {
          "name": "main_webhook",
          "type": "webhook",
          "parameters": {
            "input": "<?input.text?>"
          },
          "result_variable": "webhook_result_1"
        }
      ],
      "metadata": {
        "_customization": {
          "mcr": true
        }
      },
      "conditions": "#Product_Information",
      "dialog_node": "node_2_1589752166191",
      "previous_sibling": "node_4_1570050459690"
    },
    {
      "type": "standard",
      "output": {
        "text": {
          "values": [
            "You're welcome. Just let me know if you need anything else",
            "No problem. Just let me know if you need anything else",
            "My pleasure. Just let me know if you need anything else"
          ],
          "selection_policy": "sequential"
        }
      },
      "metadata": {},
      "conditions": "#Thanks",
      "digress_in": "does_not_return",
      "dialog_node": "node_2_1468243505617",
      "previous_sibling": "node_12_1468329566917"
    },
    {
      "type": "standard",
      "output": {
        "text": {
          "values": [
            "So long",
            "See ya",
            "Good bye"
          ],
          "selection_policy": "sequential"
        }
      },
      "metadata": {},
      "conditions": "#Goodbye",
      "digress_in": "does_not_return",
      "dialog_node": "node_12_1468329566917",
      "previous_sibling": "node_13_1502484041694"
    },
    {
      "type": "standard",
      "title": "Where are you located?",
      "output": {},
      "metadata": {},
      "next_step": {
        "behavior": "skip_user_input"
      },
      "conditions": "#Customer_Care_Store_Location",
      "digress_in": "does_not_return",
      "dialog_node": "Directions",
      "digress_out": "allow_all",
      "previous_sibling": "Hours of Operation"
    },
    {
      "type": "standard",
      "output": {
        "text": {
          "values": [
            "I didn't understand can you try again"
          ],
          "selection_policy": "sequential"
        }
      },
      "metadata": {},
      "conditions": "anything_else",
      "digress_in": "returns",
      "dialog_node": "node_2_1467831978407",
      "digress_out": "allow_all",
      "previous_sibling": "node_2_1589752166191",
      "disambiguation_opt_out": true
    },
    {
      "type": "standard",
      "title": "What can I do",
      "output": {
        "generic": [
          {
            "values": [
              {
                "text": "I can tell you about our store locations and opening hours,
or help you set up an appointment."
              },
              {
                "text": "You could also ask me to connect you to an agent."
              }
            ],
            "response_type": "text",
            "selection_policy": "sequential"
          }
        ]
      },
      "conditions": "#Help",
      "dialog_node": "node_4_1570050459690",
      "previous_sibling": "node_22_1467833484410"
    },
    {
      "type": "standard",
      "title": "What are your hours?",
      "output": {},
      "metadata": {},
      "next_step": {
        "behavior": "jump_to",
        "selector": "body",
        "dialog_node": "node_3_1522439390442"
```

```
   },
   "conditions": "#Customer_Care_Store_Hours",
   "digress_in": "does_not_return",
   "dialog_node": "Hours of Operation",
   "digress_out": "allow_all",
   "previous_sibling": "Opening"
  },
  {
   "type": "standard",
   "title": "Opening",
   "output": {
    "text": {
     "values": [
      "Hello, I’m a demo customer care virtual assistant to show you the basics.
I can help with directions to my store, hours of operation and booking an in-store
appointment"
     ],
     "selection_policy": "sequential"
    }
   },
   "context": {
    "no_reservation": true
   },
   "metadata": {},
   "conditions": "welcome",
   "dialog_node": "Opening"
  }
 ],
 "counterexamples": [],
 "system_settings": {
  "tooling": {
   "store_generic_responses": true
  },
  "off_topic": {
   "enabled": true
  },
  "disambiguation": {
   "prompt": "Did you mean:",
   "enabled": true,
   "randomize": true,
   "max_suggestions": 5,
   "suggestion_text_policy": "title",
   "none_of_the_above_prompt": "None of the above."
  },
  "system_entities": {
   "enabled": true
```

```
    },
    "spelling_auto_correct": true
  },
  "learning_opt_out": false,
  "name": "Customer Care Sample Skill",
  "language": "en",
  "description": "Sample simple customer service skill to get you started."
}
```

## Cloud Function : node.js

```
/**
  *
  * @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);
    });
  });
}
```

## Node Red Flow : flows.json

```
[{"id":"d20d4833.fce068","type":"tab","label":"Flow
1","disabled":false,"info":""},{"id":"df1a675a.3f5548","type":"ui_form","z":"d20d4833.fce068","
name":"","label":"","group":"165ffce0.fc5bf3","order":1,"width":0,"height":0,"options":[{"label
":"Enter the
query","value":"input","type":"text","required":true,"rows":null}],"formValue":{"input":""},"pa
yload":"","submit":"submit","cancel":"cancel","topic":"","x":80,"y":340,"wires":[["4b4c20f4.58e
96"]]},{"id":"4b4c20f4.58e96","type":"function","z":"d20d4833.fce068","name":"","func":"msg.pay
load = msg.payload.input;\nreturn
msg;\n","outputs":1,"noerr":0,"x":190,"y":240,"wires":[["3f2cb79e.c23be8","88271b13.f99078"]]},
{"id":"d7c932de.d158b","type":"debug","z":"d20d4833.fce068","name":"","active":true,"tosidebar"
:true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":610,"y":120,
"wires":[]},{"id":"3f2cb79e.c23be8","type":"ui_text","z":"d20d4833.fce068","group":"165ffce0.fc
5bf3","order":2,"width":0,"height":0,"name":"","label":"Input","format":"{{msg.payload}}","layo
ut":"col-
center","x":300,"y":340,"wires":[]},{"id":"6ec32a8d.900844","type":"function","z":"d20d4833.fce
068","name":"","func":"msg.payload.text=\"\"; \nif(msg.payload.context.webhook_result_1){\n
for(var i in msg.payload.context.webhook_result_1.results){ \n
msg.payload.text=msg.payload.text+\"\\n\"+msg.payload.context.webhook_result_1.results[i].text;
\n        \n    }   \n    \n    msg.payload=msg.payload.text;\n}\nelse\nmsg.payload =
msg.payload.output.text[0];\nreturn
msg;\n","outputs":1,"noerr":0,"x":540,"y":240,"wires":[["1b4fe696.410fc9"]]},{"id":"88271b13.f9
9078","type":"watson-conversation-
v1","z":"d20d4833.fce068","name":"Assistant","workspaceid":"8b8589ba-67c5-4f53-ba2d-
f04fea5ff952","multiuser":false,"context":false,"empty-payload":false,"service-
endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/6dc4a147-1de6-4449-94b7-
c78e417f5481","timeout":"","optout-
learning":false,"x":380,"y":160,"wires":[["d7c932de.d158b","6ec32a8d.900844"]]},{"id":"1b4fe696
.410fc9","type":"ui_text","z":"d20d4833.fce068","group":"7ed5f621.985e38","order":1,"width":10,
"height":12,"name":"","label":"Results","format":"{{msg.payload}}","layout":"col-
center","x":720,"y":300,"wires":[]},{"id":"165ffce0.fc5bf3","type":"ui_group","z":"","name":"IN
PUT","tab":"46915180.2a5e8","order":1,"disp":true,"width":8,"collapse":false},{"id":"7ed5f621.9
85e38","type":"ui_group","z":"","name":"OUTPUT","tab":"46915180.2a5e8","order":2,"disp":true,"w
idth":10,"collapse":false},{"id":"46915180.2a5e8","type":"ui_tab","z":"","name":"WELCOME","icon
":"dashboard","disabled":false,"hidden":false}]
```

## B. References

- https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery
- https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started
- https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/
- http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red
- https://github.com/IBM/watson-discovery-sdu-with-assistant
- https://www.youtube.com/watch?v=Jpr3wVH3FVA