PROJECT REPORT

NAME	**SHAIK MUBEEN (mubeena6sk@gmail.com)**
TITLE	**Intelligent Customer Help Desk With Smart Document Understanding**
CATEGORY	**Machine Learning**

1.INTRODUCTION

1.1 Overview:

We will be able to design 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 + Assistant.

- Project Requirements: Python, IBM Cloud, IBM Watson, Node- RED
- Functional Requirements: IBM cloud
- Technical Requirements: AI,ML,WATSON AI
- Software Requirements: Watson assistant, Watson discovery.
- Project Deliverables: Smartinternz Intership
- Project Team: Shaik Mubeen
- Project Duration:19 days

1.2 Purpose:

The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn't valid or offer to speak to a real person. In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded w ith 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 on what text in the owner's manual is important and what is not. This will improve the answers returned from the queries.

1.2.1 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:

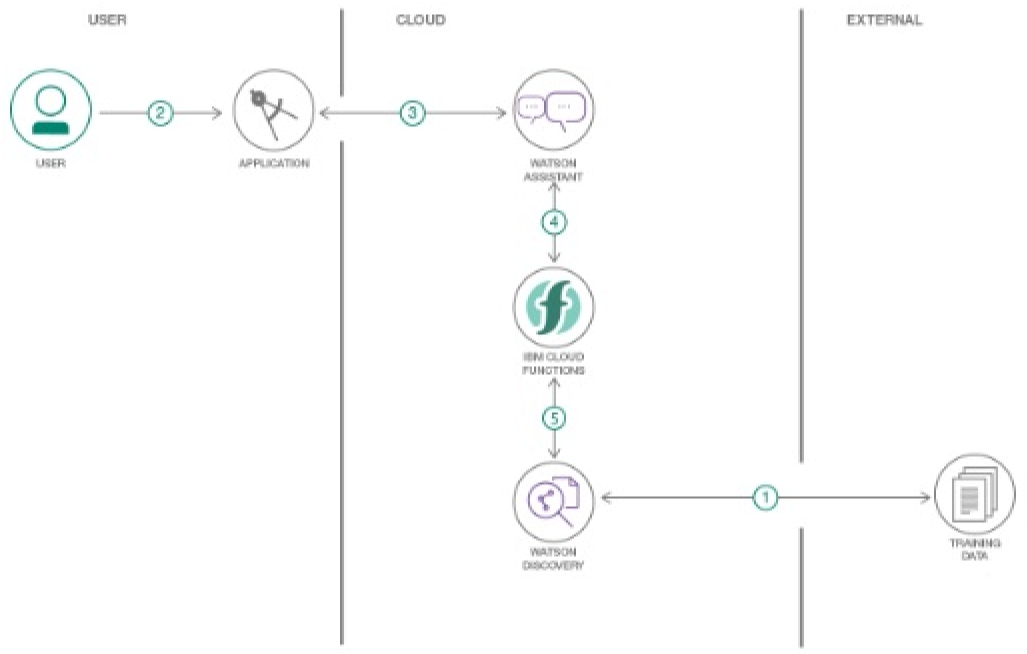
Generally Chatbots means getting input from users and getting only response questions and for some questions the output from bot will be like “ try again”, “I don’t understand”, “will you repeat again”, and so on… and directs customer to customer agent but a good customer Chatbot should minimize involvement of customer agent to chat with customer to clarify his/her doubts. So to achieve this we should include an virtual agent in c hatbot so that it will take care of real involvement of customer agent and c ustomer 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 s olution. And later including Assistant and Discovery on Node-RED.

3 .THEORITICAL ANALYSIS

3.1 Block/Flow Diagram



1 .The document is annotated using Watson Discovery SDU
2 . The user interacts with the backend server via the app UI. The frontend   a pp 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

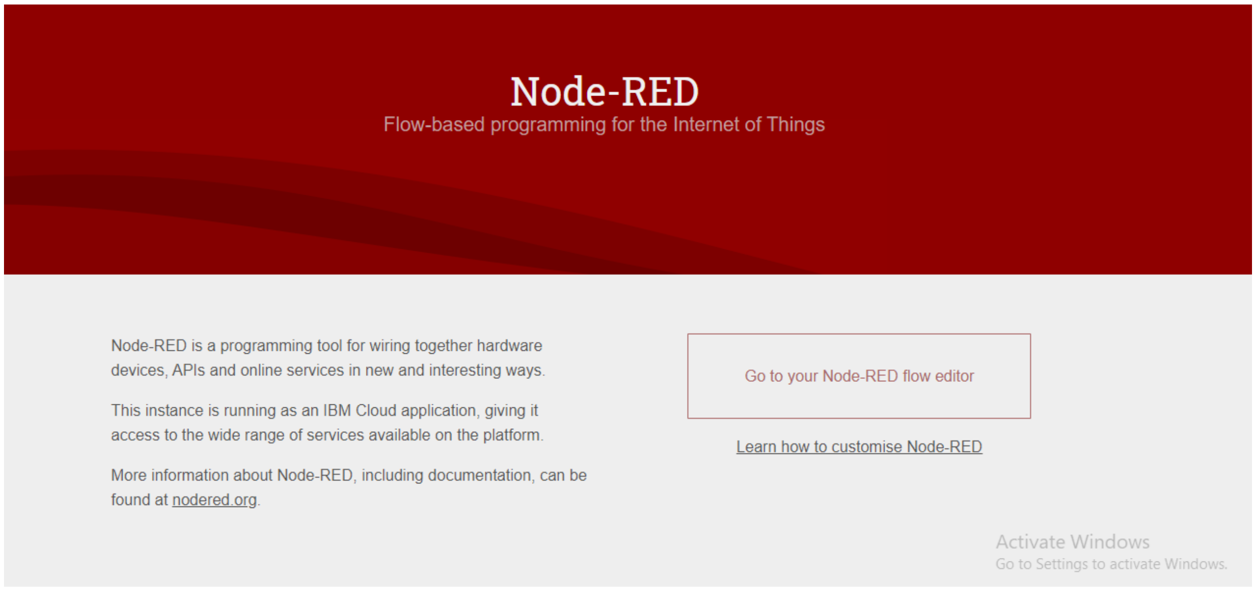
1 .Create IBM Cloud services

Create the following services:

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

Creation of 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
- Note: Your Node-red service is starting
- Step-5: We have to configure Node red for the first time. Click on next to continue
- Step-6: Secure your node red editor by giving a username and password and click on Next
- Step-7: Click to Continue
- Step-8: Click Finish



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.

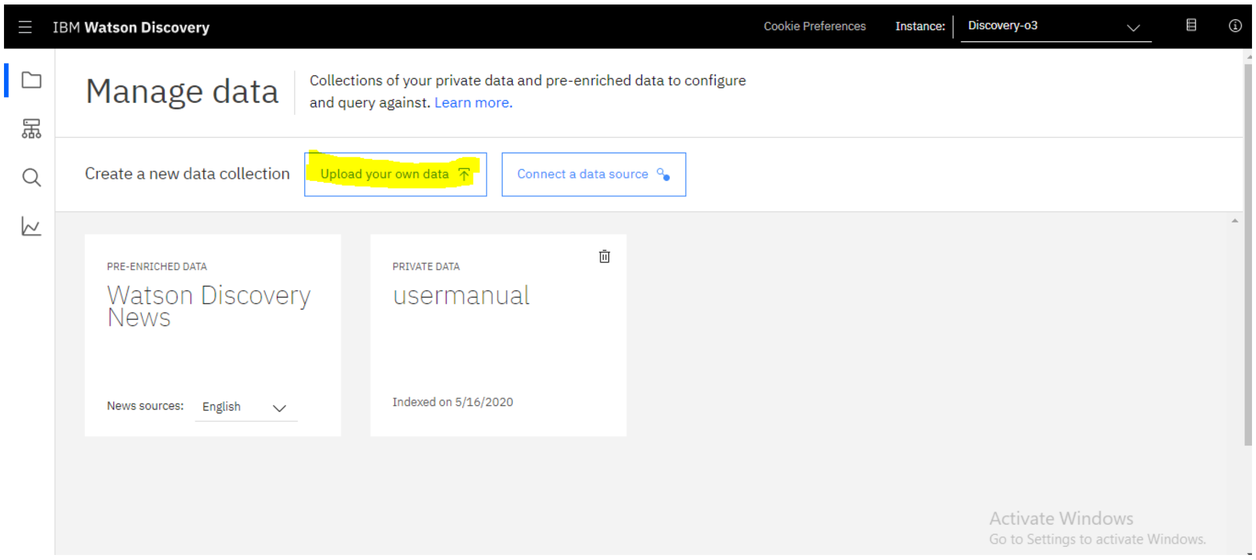
More information about Node-RED, including documentation, can be found at nodered.org.

Go to your Node-RED flow editor
Learn how to customise Node-RED
Activate Windows
Go to Settings to activate Windows.

Creation of Watson discovery instance in IBM Cloud:

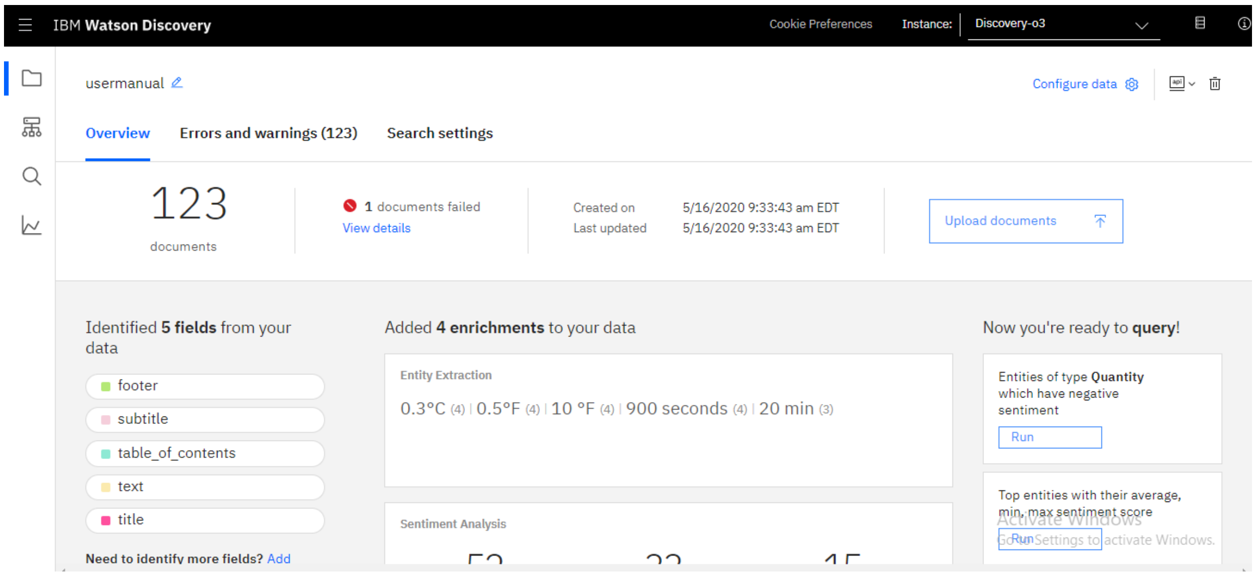
● Launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option.Give the data collection a unique name. When prompted, select and upload the ecobee3_UserGuide.pdf.

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.



Manage data

Collections of your private data and pre-enriched data to configure and query against. Learn more.


IBM Watson Discovery

usermanual

Overview | Errors and warnings (123) | Search settings

123 documents

1 documents failed
View details

Created on 5/16/2020 9:33:43 am EDT
Last updated 5/16/2020 9:33:43 am EDT

Upload documents

Identified **5 fields** from your data
- footer
- subtitle
- table_of_contents
- text
- title

Need to identify more fields? Add

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)

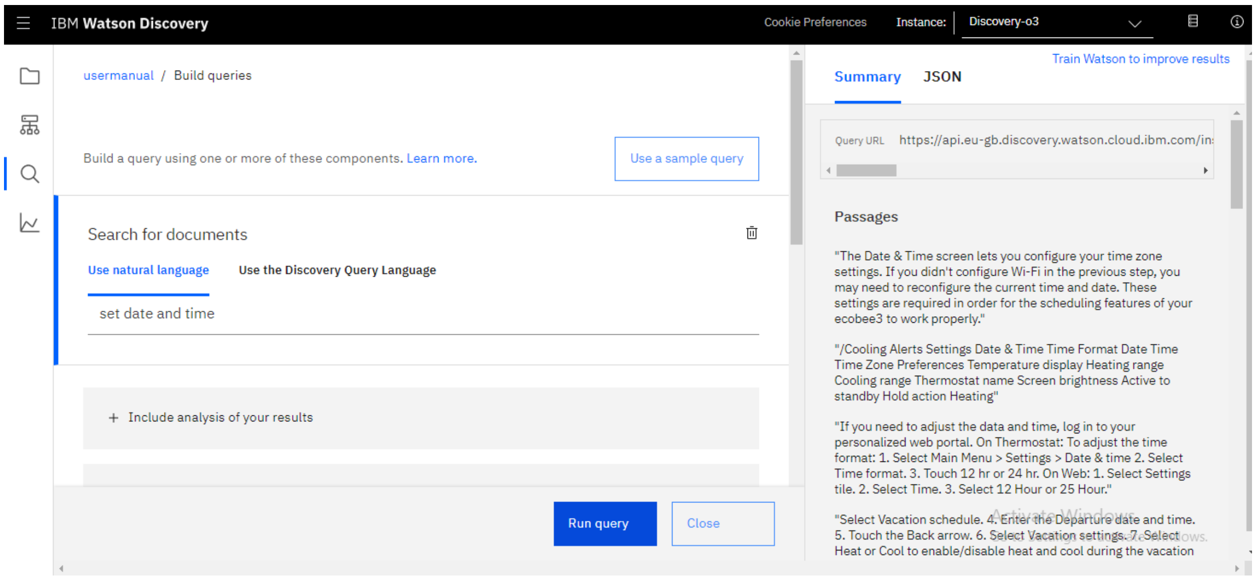
Sentiment Analysis

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

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

Top entities with their average, min, max sentiment score
Run

● Click the Build your own query button.


IBM Watson Discovery

usermanual / 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

set date and time

+ Include analysis of your results

Run query | Close

Summary | JSON

Train Watson to improve results

Query URL https://api.eu-gb.discovery.watson.cloud.ibm.com/in

Passages

"The Date & Time screen lets you configure your time zone settings. If you didn't configure Wi-Fi in the previous step, you may need to reconfigure the current time and date. These settings are required in order for the scheduling features of your ecobee3 to work properly."

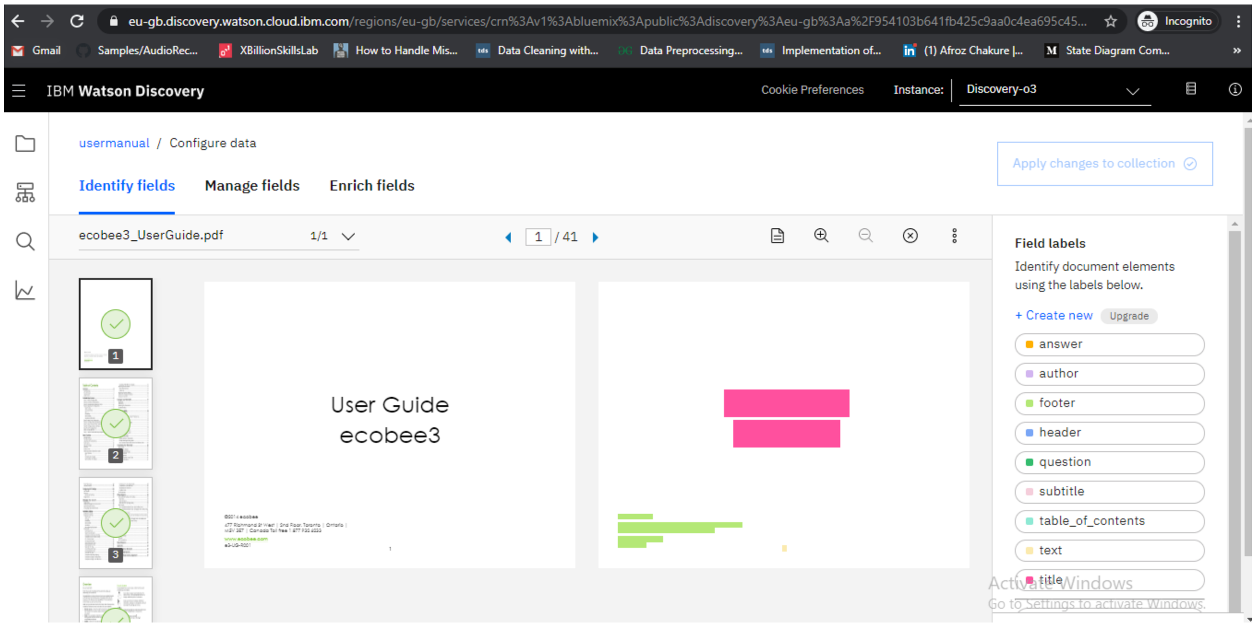
"/Cooling Alerts Settings Date & Time Time Format Date Time Time Zone Preferences Temperature display Heating range Cooling range Thermostat name Screen brightness Active to standby Hold action Heating"

"If you need to adjust the data and time, log in to your personalized web portal. On Thermostat: To adjust the time format: 1. Select Main Menu > Settings > Date & time 2. Select Time format. 3. Touch 12 hr or 24 hr. On Web: 1. Select Settings tile. 2. Select Time. 3. Select 12 Hour or 25 Hour."

"Select Vacation schedule. 4. Enter the Departure date and time. 5. Touch the Back arrow. 6. Select Vacation settings. 7. Select Heat or Cool to enable/disable heat and cool during the vacation

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:



eu-gb.discovery.watson.cloud.ibm.com/regions/eu-gb/services/crn%3Av1%3Abluemix%3Apublic%3Adiscovery%3Aeu-gb%3Aa%2F954103b641fb425c9aa0c4ea695c45...
IBM Watson Discovery
Cookie Preferences
Instance: Discovery-o3
usermanual / Configure data
Identify fields
Manage fields
Enrich fields
Apply changes to collection
ecobee3_UserGuide.pdf 1/1
1 / 41
User Guide
ecobee3
Field labels
Identify document elements using the labels below.
+ Create new Upgrade
answer
author
footer
header
question
subtitle
table_of_contents
text
title

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.

● Those are the list of pages in the manual. As each is processed, a green check mark will appear on the page.

● You need tois where you select text and assign it a label.

● There is a list of labels you can assign to the page text.

● Click to submit the page to Discovery.

● Click "Apply to collection" 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 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

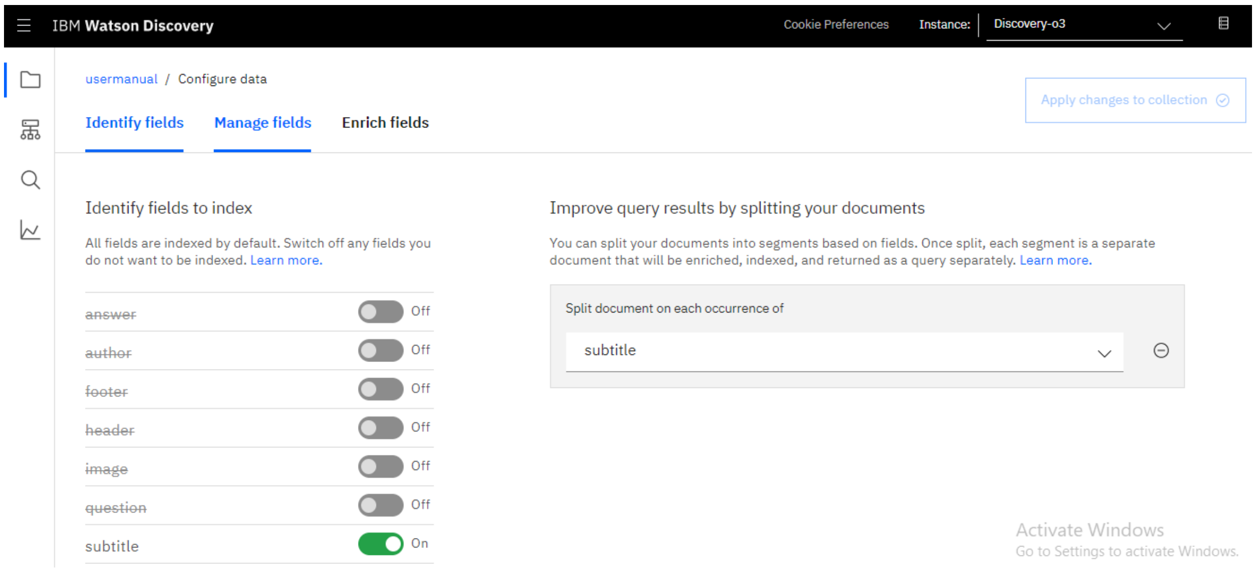
● All headers and sub-headers (typed in light green text) as a subtitle

● 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 tab.



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

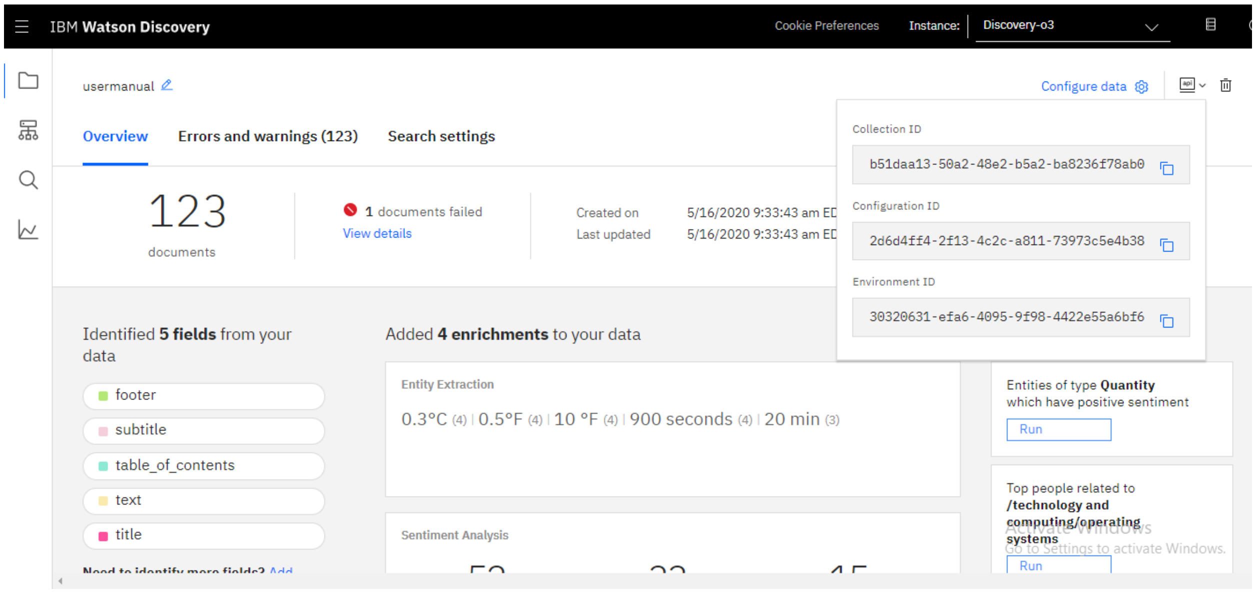
Improve query results by splitting your documents

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

● Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except subtitles and text.
● Split document is telling Discovery to split the document apart, based on subtitle.
● Click 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

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 located at the top right side of your collection panel:


IBM Watson Discovery   Cookie Preferences   Instance: Discovery-o3

usermanual

Overview   Errors and warnings (123)   Search settings

Configure data

Collection ID
b51daa13-50a2-48e2-b5a2-ba8236f78ab0

Configuration ID
2d6d4ff4-2f13-4c2c-a811-73973c5e4b38

Environment ID
30320631-efa6-4095-9f98-4422e55a6bf6

123
documents

1 documents failed
View details

Created on   5/16/2020 9:33:43 am ED
Last updated   5/16/2020 9:33:43 am ED

Identified **5 fields** from your data
- footer
- subtitle
- table_of_contents
- text
- title

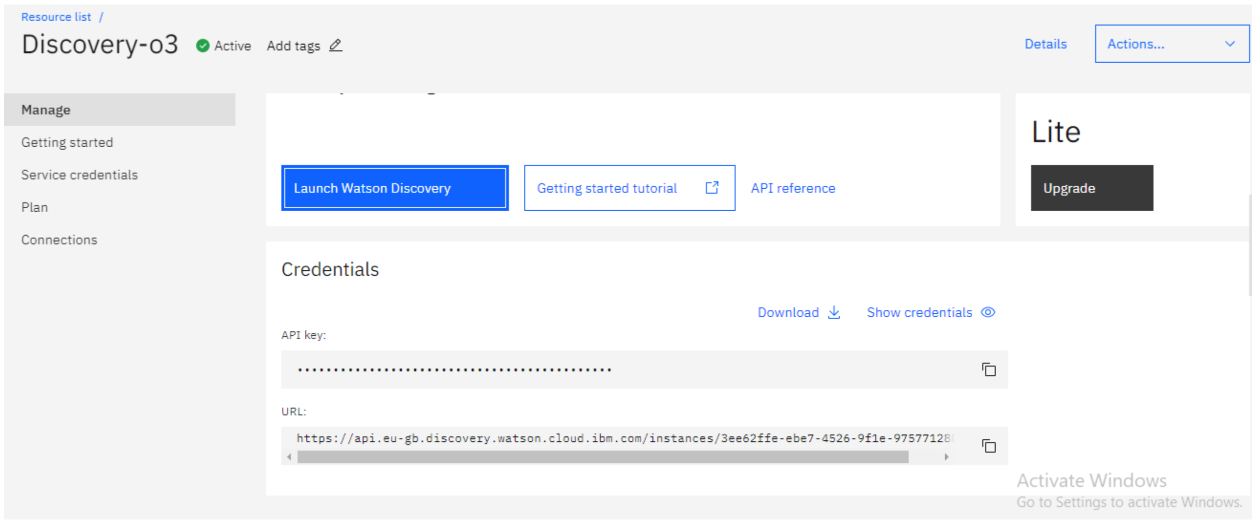
Added **4 enrichments** to your data

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

Sentiment Analysis

Entities of type **Quantity** which have positive sentiment
Run

Top people related to **/technology and computing/operating systems**
Run


Resource list /

Discovery-o3   Active   Add tags   Details   Actions...

Manage
Getting started
Service credentials
Plan
Connections

Launch Watson Discovery   Getting started tutorial   API reference

Lite
Upgrade

Credentials

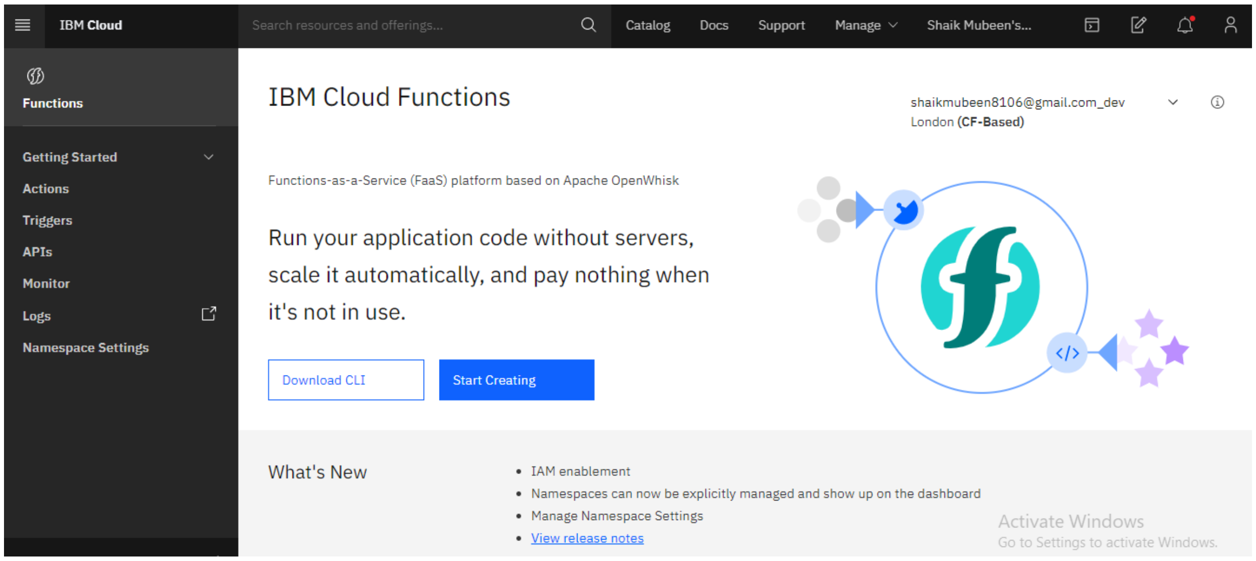
Download   Show credentials

API key:
```
............................................
```
URL:
```
https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/3ee62ffe-ebe7-4526-9f1e-97577128
```

Also store URL and api key of watson discovery.

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 , then select the Functions card:



IBM Cloud Functions

shaikmubeen8106@gmail.com_dev
London (CF-Based)

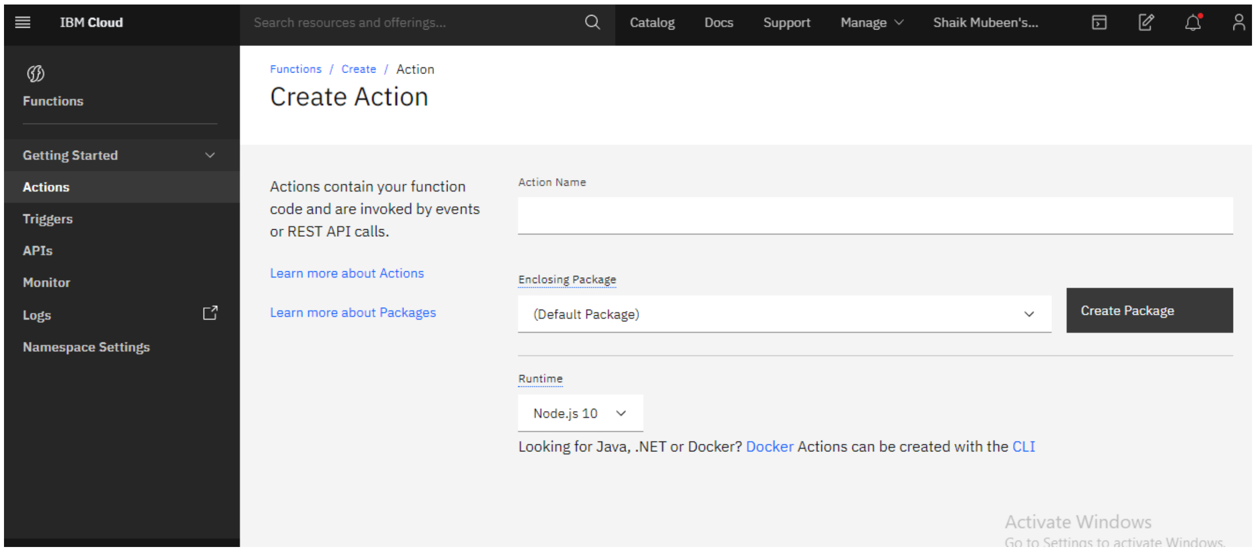
Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk

Run your application code without servers, scale it automatically, and pay nothing when it's not in use.

What's New

- IAM enablement
- Namespaces can now be explicitly managed and show up on the dashboard
- Manage Namespace Settings
- View release notes

From the Functions main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option. On the Create Action panel, provide a unique Action Name, keep the default package, and select the Node.js 10 runtime. Click the Create button to create the action



Create Action

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

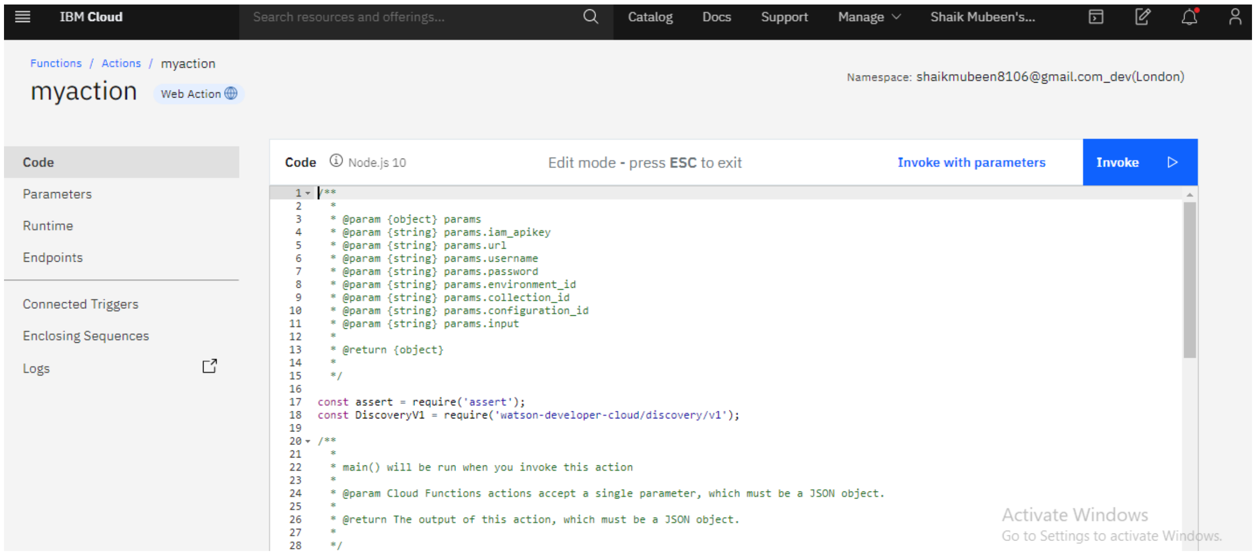
Action Name:

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,
In the code editor window, cut and paste in the code from the myaction.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, it will fail due to credentials not being defined yet. We'll do this next. Select the Parameters tab:

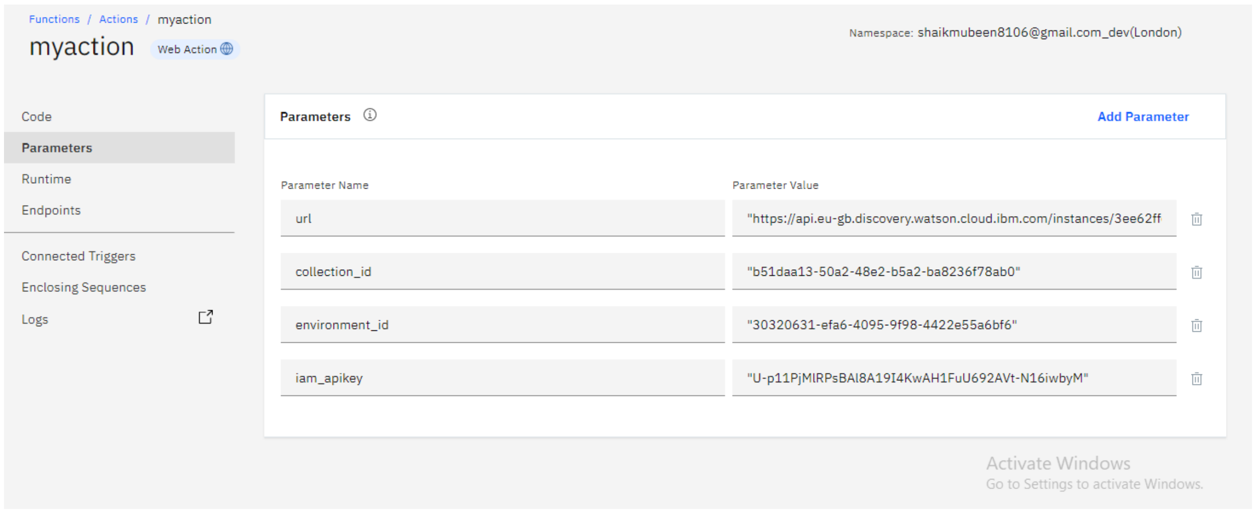


```
/**
  *
  * @param {object} params
  * @param {string} params.iam_apikey
  * @param {string} params.url
  * @param {string} params.username
  * @param {string} params.password
  * @param {string} params.environment_id
  * @param {string} params.collection_id
  * @param {string} params.configuration_id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

Add the following keys:

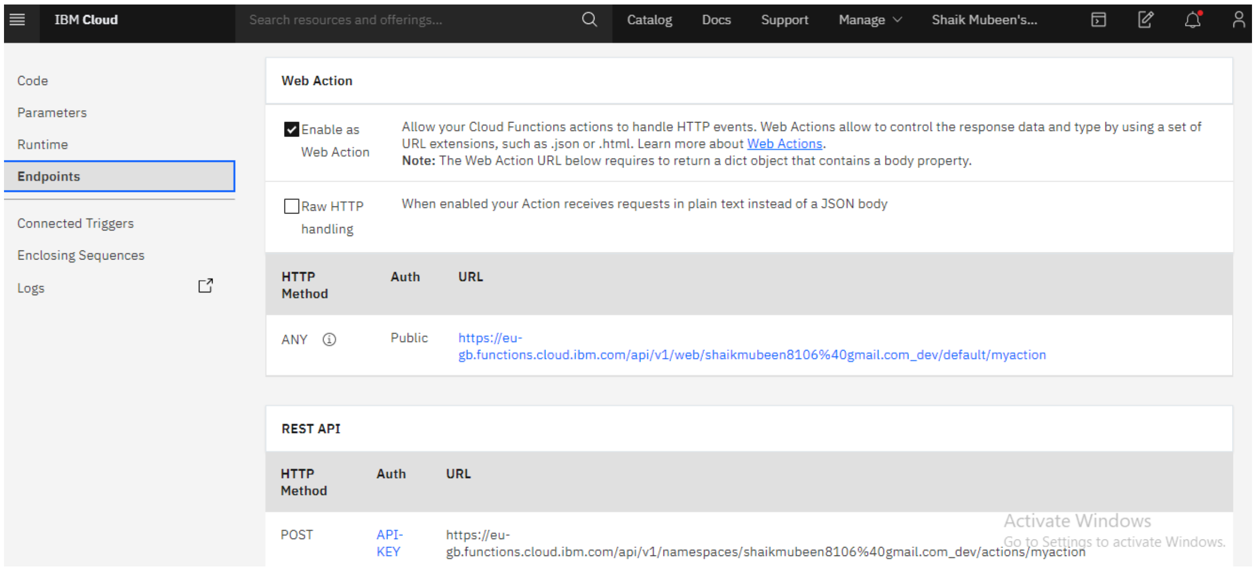
- url
- environment_id
- collection_id
- iam_apikey

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:



Parameter Name	Parameter Value
url	"https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/3ee62ff
collection_id	"b51daa13-50a2-48e2-b5a2-ba8236f78ab0"
environment_id	"30320631-efa6-4095-9f98-4422e55a6bf6"
iam_apikey	"U-p11PjMlRPsBAl8A19I4KwAH1FuU692AVt-N16iwbyM"

Next, go to the Endpoints panel:



Web Action

[✓] Enable as Web Action — Allow your Cloud Functions actions to handle HTTP events. Web Actions allow to control the response data and type by using a set of URL extensions, such as .json or .html. Learn more about Web Actions.
Note: The Web Action URL below requires to return a dict object that contains a body property.

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

HTTP Method	Auth	URL
ANY	Public	https://eu-gb.functions.cloud.ibm.com/api/v1/web/shaikmubeen8106%40gmail.com_dev/default/myaction

REST API

HTTP Method	Auth	URL
POST	API-KEY	https://eu-gb.functions.cloud.ibm.com/api/v1/namespaces/shaikmubeen8106%40gmail.com_dev/actions/myaction

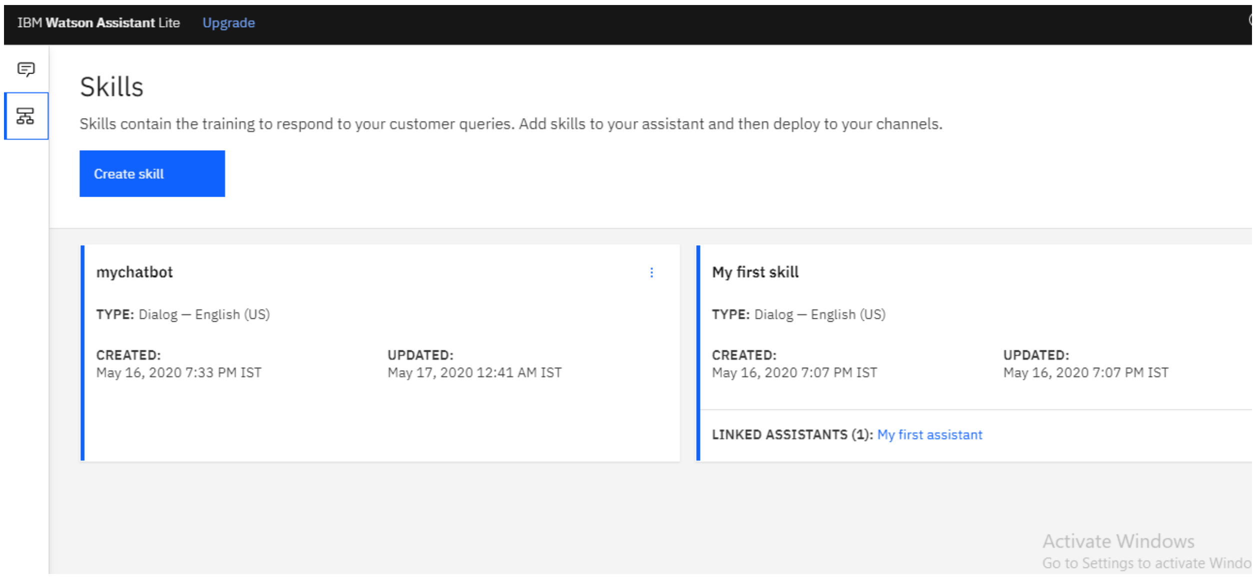
Activate Windows
Go to Settings to activate Windows.

Click the checkbox for Enable as Web Action . This will generate a public endpoint URL . Take note of the URL value, 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. 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 Watson 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 skill.


IBM Watson Assistant Lite Upgrade

Skills

Skills contain the training to respond to your customer queries. Add skills to your assistant and then deploy to your channels.

Create skill

mychatbot

TYPE: Dialog — English (US)

CREATED: May 16, 2020 7:33 PM IST

UPDATED: May 17, 2020 12:41 AM IST

My first skill

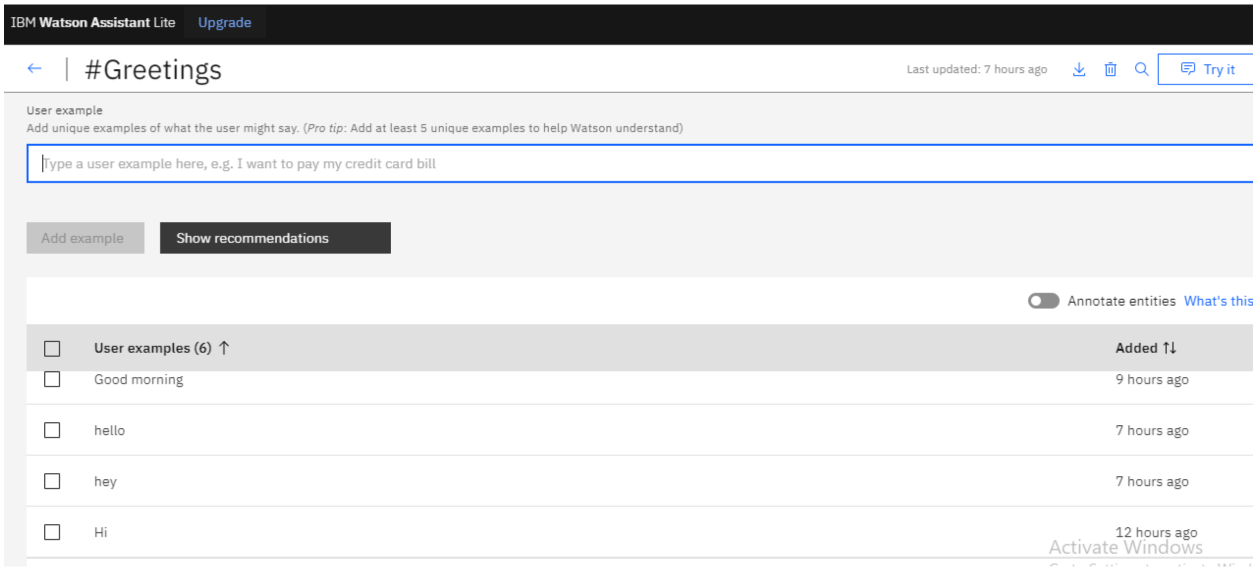
TYPE: Dialog — English (US)

CREATED: May 16, 2020 7:07 PM IST

UPDATED: May 16, 2020 7:07 PM IST

LINKED ASSISTANTS (1): My first assistant

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


IBM Watson Assistant Lite Upgrade

#Greetings

Last updated: 7 hours ago

Try it

User example
Add unique examples of what the user might say. (*Pro tip:* Add at least 5 unique examples to help Watson understand)

Type a user example here, e.g. I want to pay my credit card bill

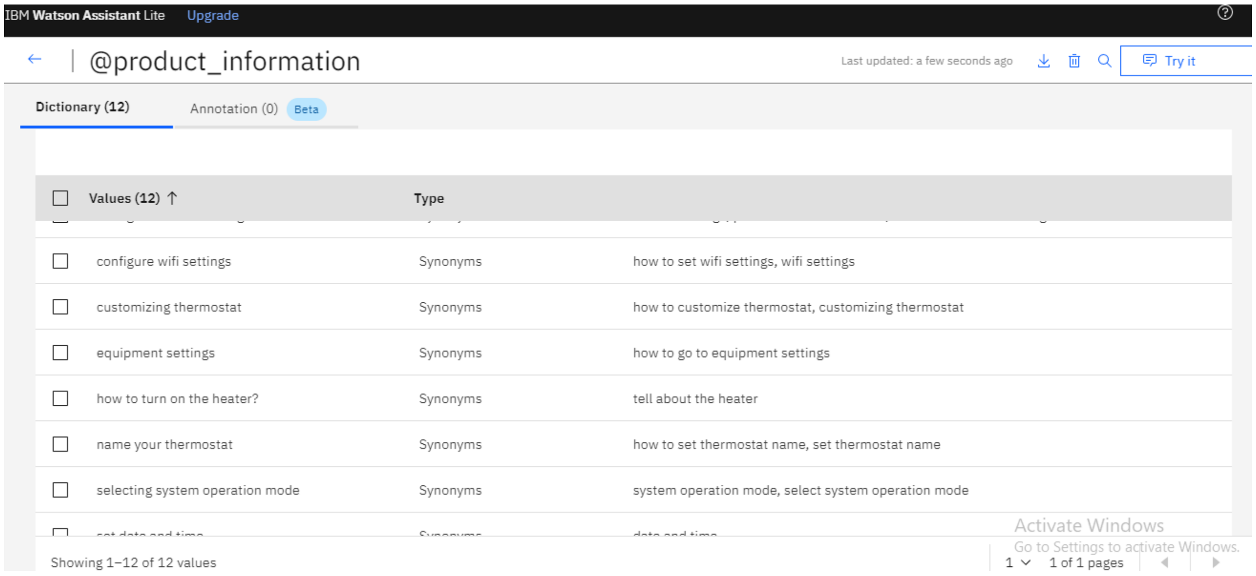
Add example   Show recommendations

Annotate entities What's this

User examples (6)	Added
Good morning	9 hours ago
hello	7 hours ago
hey	7 hours ago
Hi	12 hours ago

Add new entity

Along with all possible cases of addressing the query.

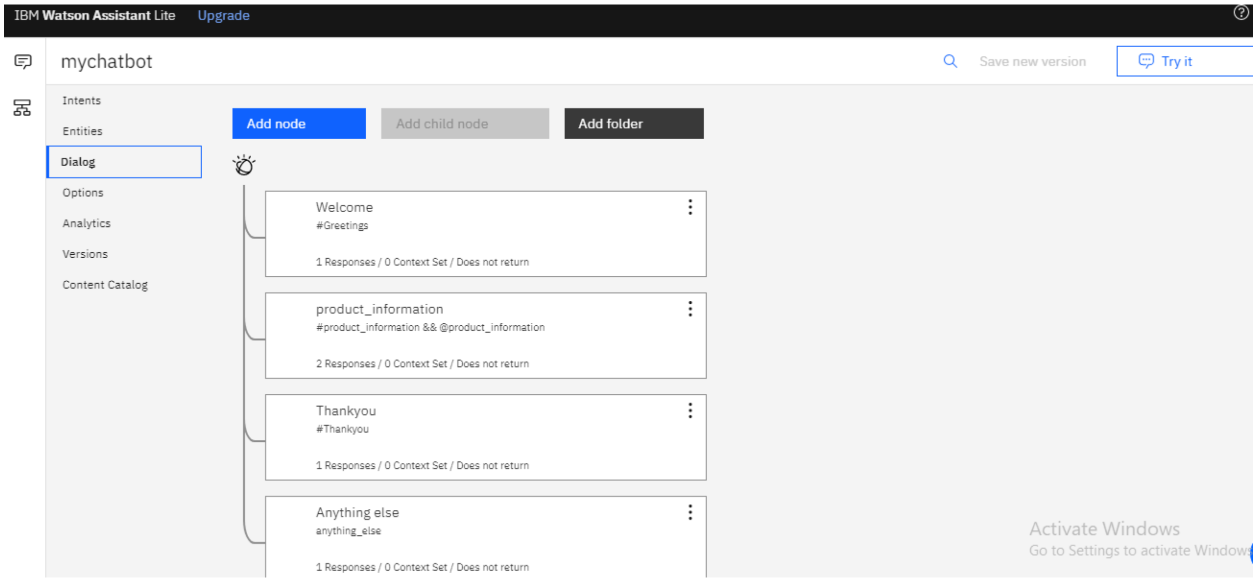


Values (12) ↑	Type	
configure wifi settings	Synonyms	how to set wifi settings, wifi settings
customizing thermostat	Synonyms	how to customize thermostat, customizing thermostat
equipment settings	Synonyms	how to go to equipment settings
how to turn on the heater?	Synonyms	tell about the heater
name your thermostat	Synonyms	how to set thermostat name, set thermostat name
selecting system operation mode	Synonyms	system operation mode, select system operation mode
set date and time	Synonyms	date and time

Showing 1–12 of 12 values   1   1 of 1 pages

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.



Welcome
#Greetings
1 Responses / 0 Context Set / Does not return

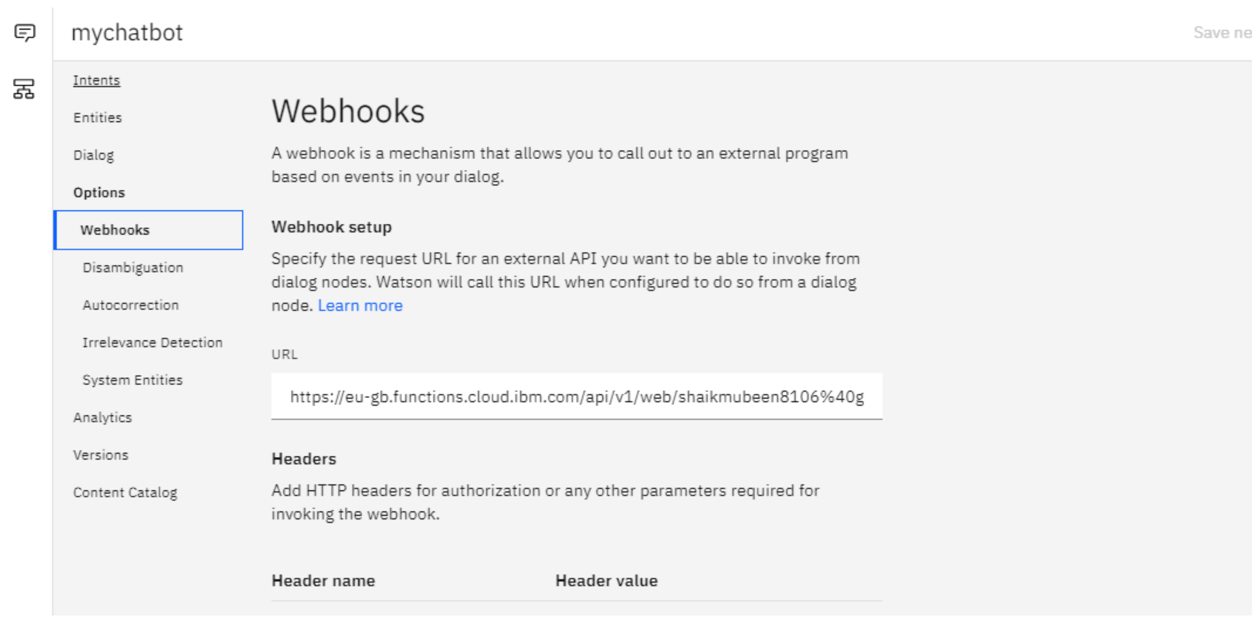
product_information
#product_information && @product_information
2 Responses / 0 Context Set / Does not return

Thankyou
#Thankyou
1 Responses / 0 Context Set / Does not return

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

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



Webhooks

A webhook is a mechanism that allows you to call out to an external program based on events in your dialog.

Webhook setup

Specify the request URL for an external API you want to be able to invoke from dialog nodes. Watson will call this URL when configured to do so from a dialog node. Learn more

URL

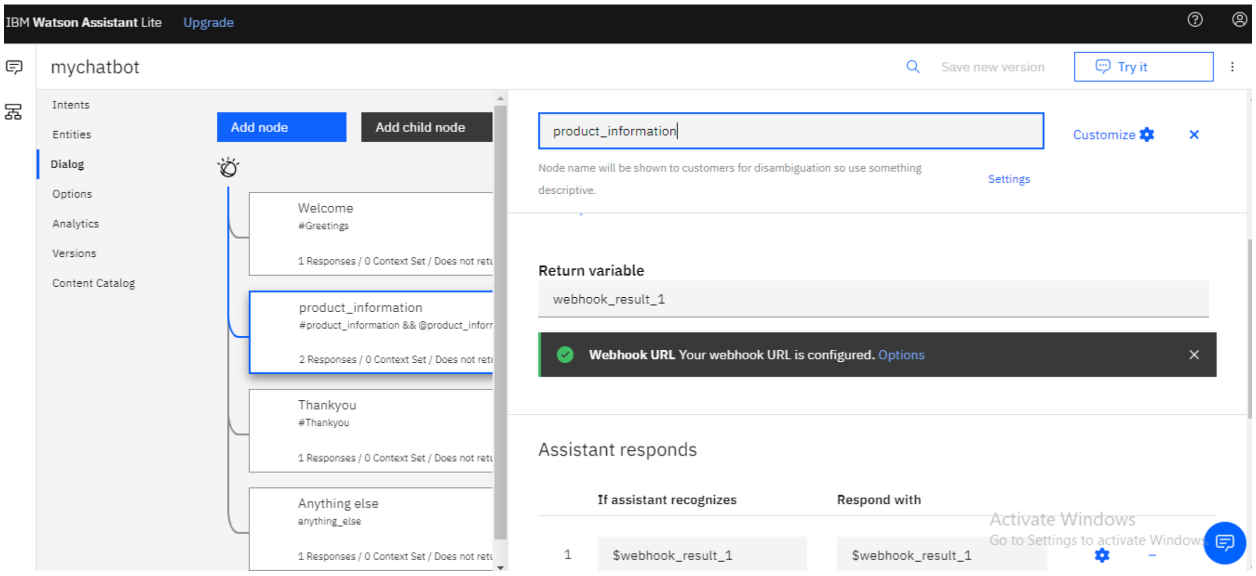
https://eu-gb.functions.cloud.ibm.com/api/v1/web/shaikmubeen8106%40g

Headers

Add HTTP headers for authorization or any other parameters required for invoking the webhook.

Header name	Header value

Enter the public URL endpoint for your action.
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


IBM Watson Assistant Lite
Upgrade
mychatbot
Save new version
Try it
Intents
Entities
Dialog
Options
Analytics
Versions
Content Catalog
Add node
Add child node
Welcome
#Greetings
1 Responses / 0 Context Set / Does not ret
product_information
#product_information && @product_infor
2 Responses / 0 Context Set / Does not ret
Thankyou
#Thankyou
1 Responses / 0 Context Set / Does not ret
Anything else
anything_else
1 Responses / 0 Context Set / Does not ret
Customize
Settings
Activate Windows
Go to Settings to activate Windows

product_information

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

Return variable

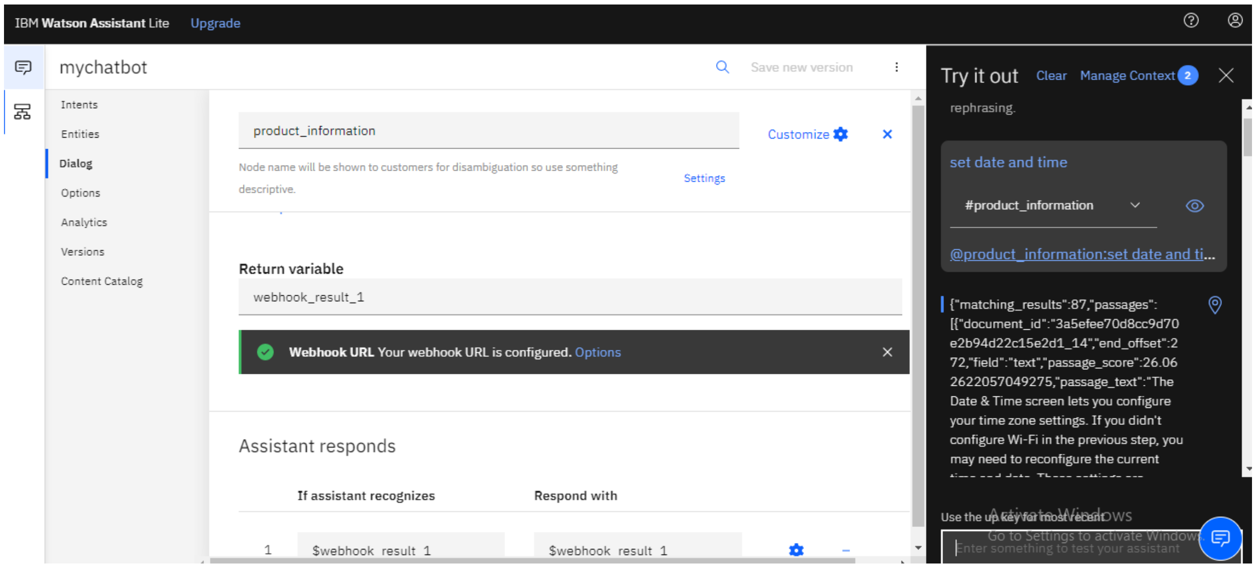
webhook_result_1

Webhook URL Your webhook URL is configured. Options

Assistant responds

	If assistant recognizes	Respond with
1	$webhook_result_1	$webhook_result_1

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” in respond with in Assistant responds block as shown below. This gives json format output.



product_information

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

Return variable

webhook_result_1

Webhook URL Your webhook URL is configured. Options

Assistant responds

	If assistant recognizes	Respond with
1	$webhook_result_1	$webhook_result_1

rephrasing.

set date and time

#product_information

@product_information:set date and ti...

{"matching_results":87,"passages":[{"document_id":"3a5efee70d8cc9d70e2b94d22c15e2d1_14","end_offset":272,"field":"text","passage_score":26.062622057049275,"passage_text":"The Date & Time screen lets you configure your time zone settings. If you didn't configure Wi-Fi in the previous step, you may need to reconfigure the current

Use the up key for most recent

Enter something to test your assistant

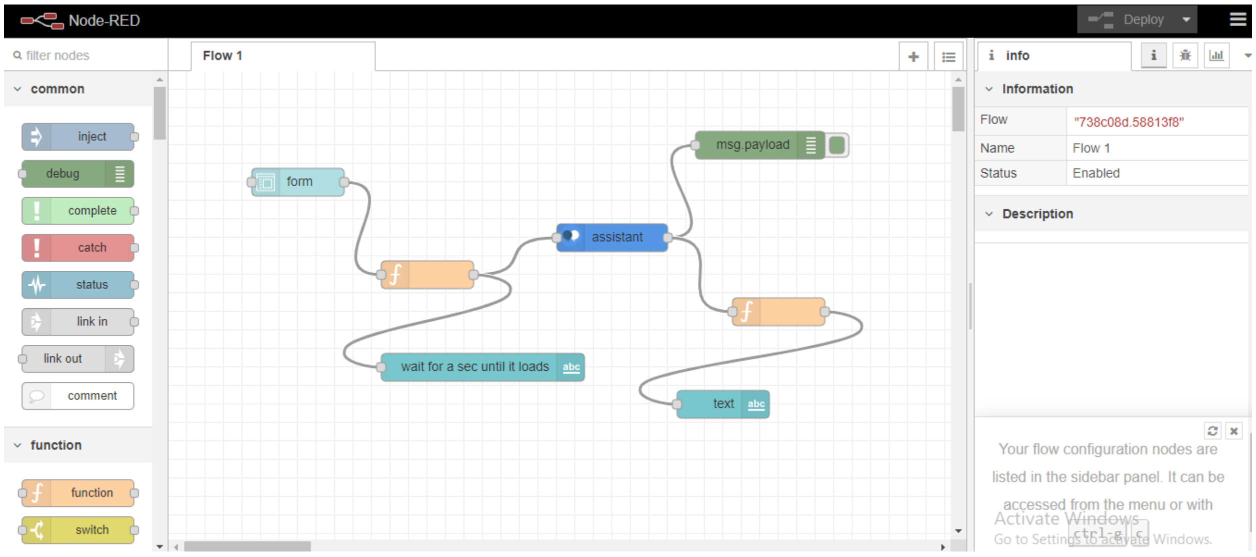
Integration of watson assistant in Node-RED

● 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
● Select the payload as a string
● Enter a sample input to be sent to the assistant service and click on done
● 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
● The Bot output is located inside “output.text"
● Drag the function node to parse the JSON data and get the bot response
● 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

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.

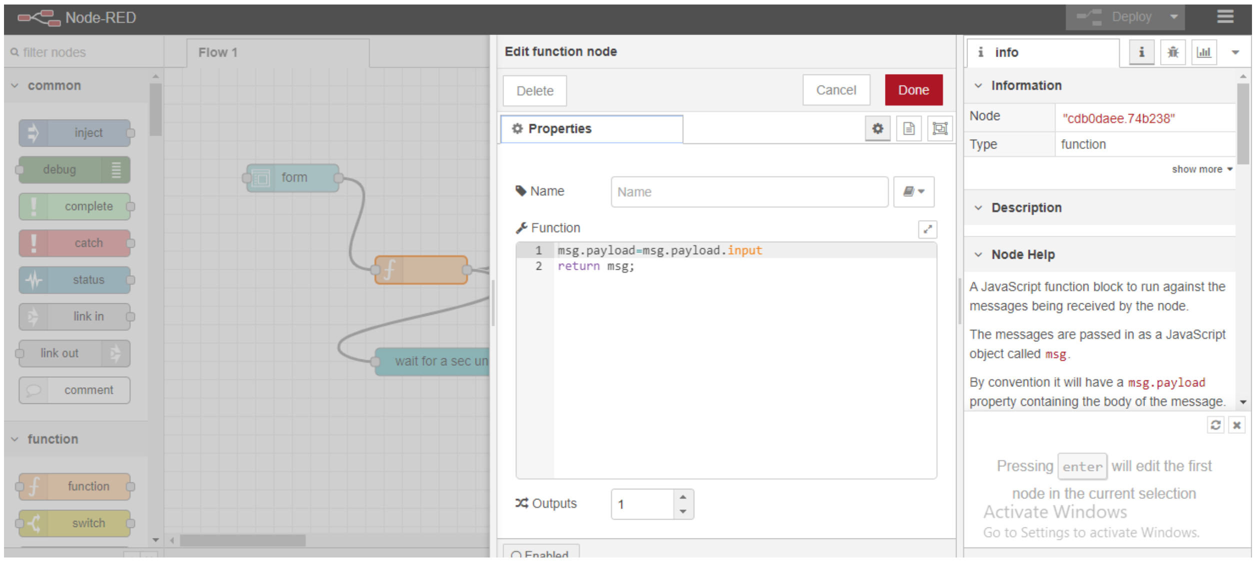
● Go to navigation pane and click on manage palette
● Click on install
● Search for “node-red-dashboard” and click on install and again click on install on the prompt

- 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 as shown below



Flow	"738c08d.58813f8"
Name	Flow 1
Status	Enabled

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with ctrl-g c



```
msg.payload=msg.payload.input
return msg;
```

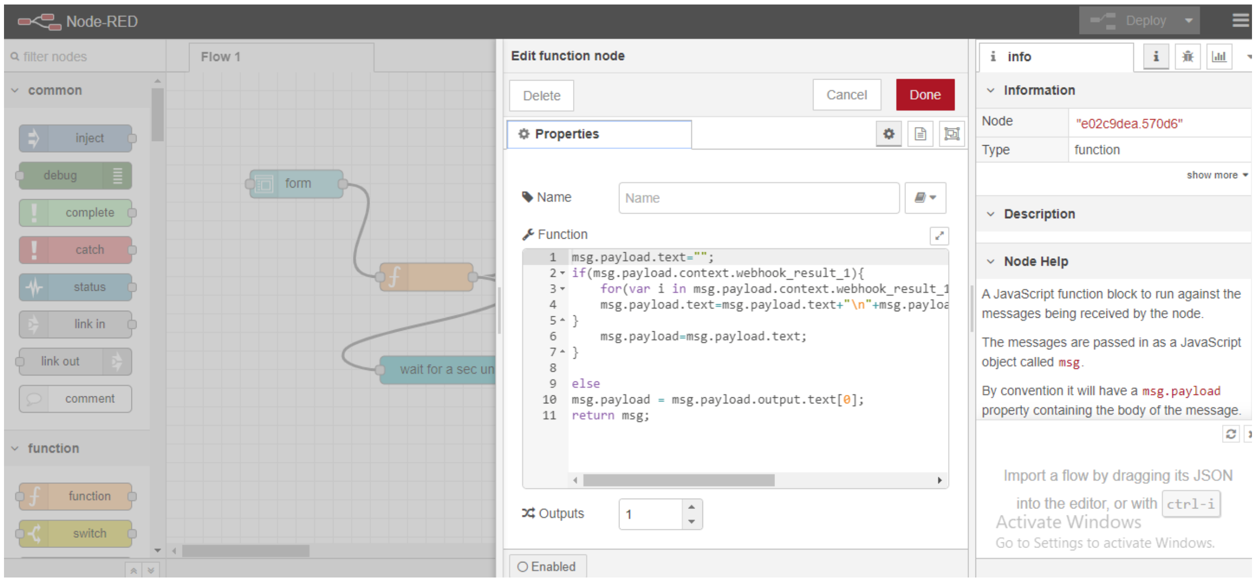
Node	"cdb0daee.74b238"
Type	function

A JavaScript function block to run against the messages being received by the node.

The messages are passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

Pressing enter will edit the first node in the current selection



Node	"e02c9dea.570d6"
Type	function

```
msg.payload.text="";
if(msg.payload.context.webhook_result_1){
    for(var i in msg.payload.context.webhook_result_1
    msg.payload.text=msg.payload.text+"\n"+msg.payloa
}
    msg.payload=msg.payload.text;
}

else
msg.payload = msg.payload.output.text[0];
return msg;
```

A JavaScript function block to run against the messages being received by the node.

The messages are passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

Import a flow by dragging its JSON into the editor, or with `ctrl-i`

Activate Windows
Go to Settings to activate Windows.

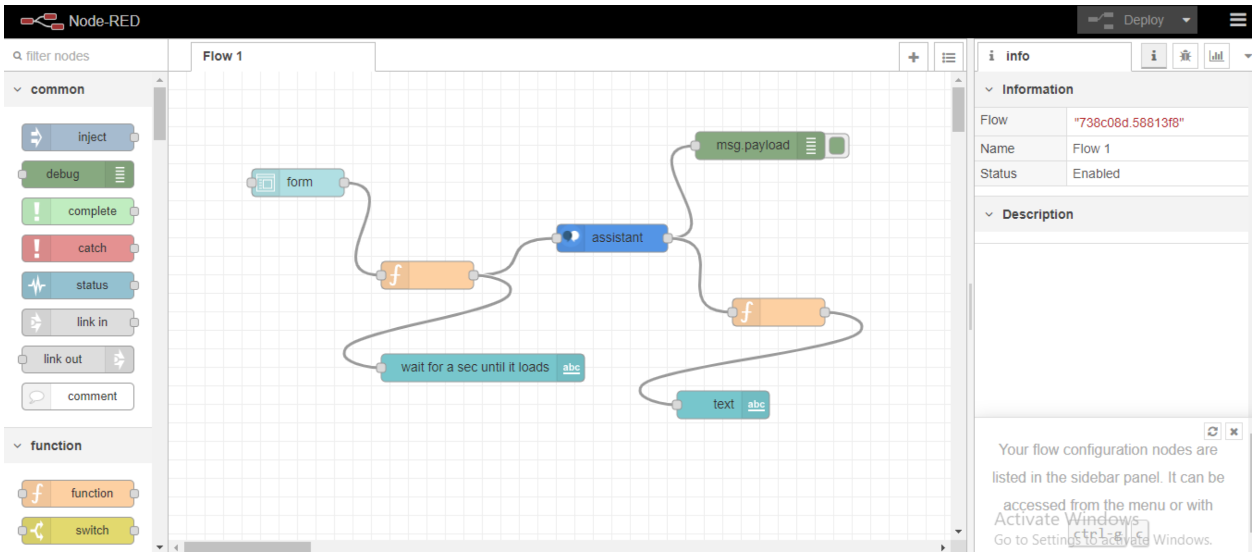
- Click on done
- Connect the form output to the input of the function node and output of the function to input of assistant node
- Search for “text” node from the “dashboard” section
- Drag two “text” nodes on to the flow
- Double click on the first text node, change the label as “wait fo r a sec to load” and click on Done
- Click on Deploy

5.FLOWCHART

1.Create flow and configurenode:

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

- Ui_Form
- Function(input parsing)
- Ui_text
- Function
- Assistant
- Debug
- Function (json to text)
- Text (output)



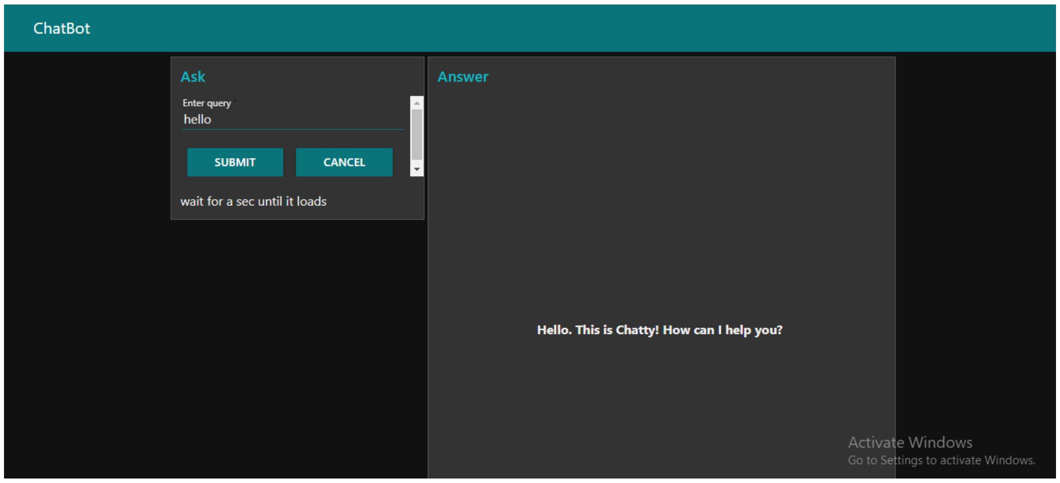
Flow	"738c08d.58813f8"
Name	Flow 1
Status	Enabled

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with ctrl-g c

Activate Windows
Go to Settings to activate Windows.

6.RESULTS

Finally our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing URL-https://node-red-qpyap.eu-gb.mybluemix.net/ui in browser


ChatBot

Ask

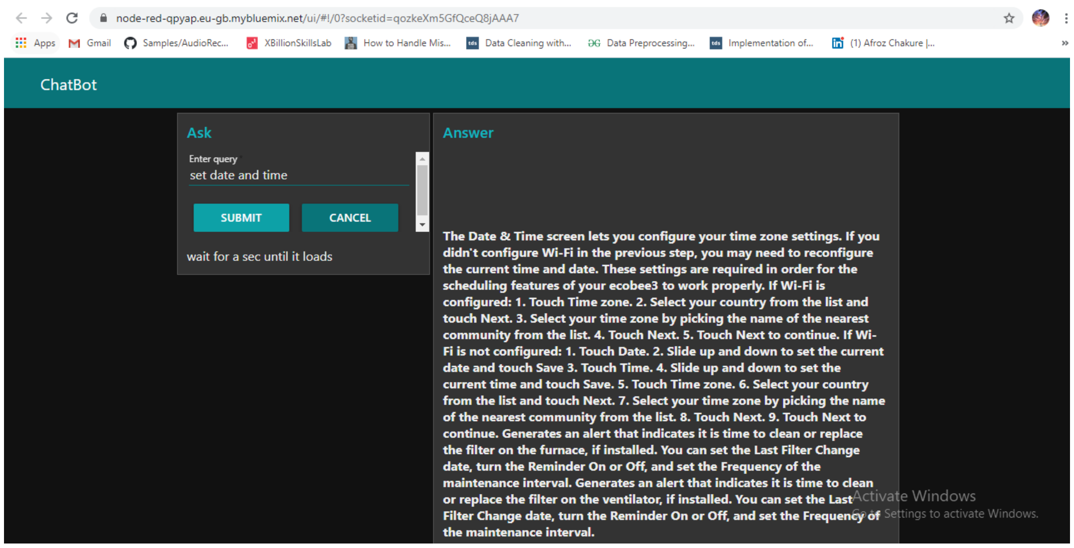
Enter query
hello

SUBMIT CANCEL

wait for a sec until it loads

Answer

Hello. This is Chatty! How can I help you?


node-red-qpyap.eu-gb.mybluemix.net/ui/#!/0?socketid=qozkeXm5GfQceQ8jAAA7

ChatBot

Ask

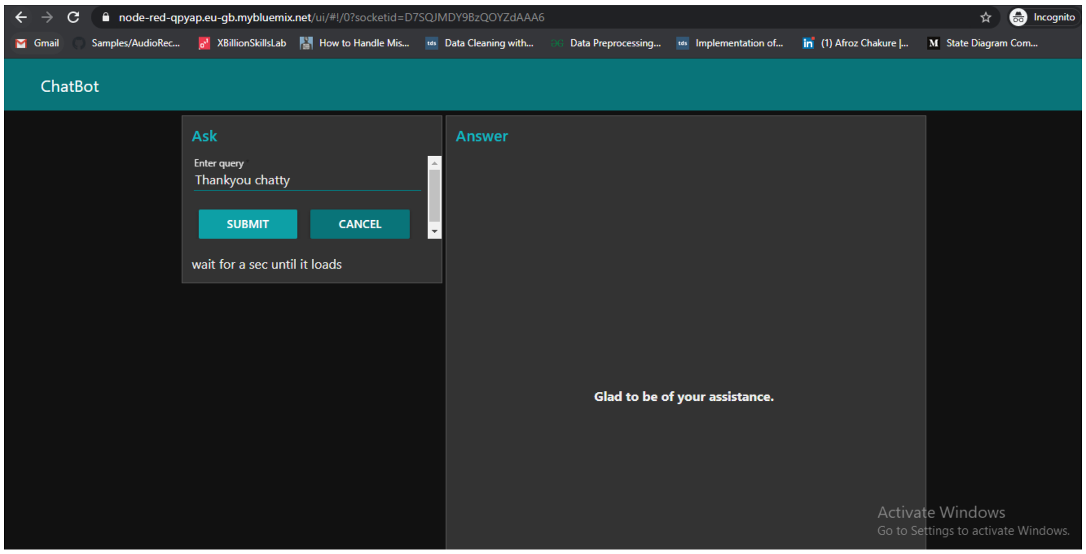
Enter query
set date and time

SUBMIT CANCEL

wait for a sec until it loads

Answer

The Date & Time screen lets you configure your time zone settings. If you didn't configure Wi-Fi in the previous step, you may need to reconfigure the current time and date. These settings are required in order for the scheduling features of your ecobee3 to work properly. If Wi-Fi is configured: 1. Touch Time zone. 2. Select your country from the list and touch Next. 3. Select your time zone by picking the name of the nearest community from the list. 4. Touch Next. 5. Touch Next to continue. If Wi-Fi is not configured: 1. Touch Date. 2. Slide up and down to set the current date and touch Save 3. Touch Time. 4. Slide up and down to set the current time and touch Save. 5. Touch Time zone. 6. Select your country from the list and touch Next. 7. Select your time zone by picking the name of the nearest community from the list. 8. Touch Next. 9. Touch Next to continue. Generates an alert that indicates it is time to clean or replace the filter on the furnace, if installed. You can set the Last Filter Change date, turn the Reminder On or Off, and set the Frequency of the maintenance interval. Generates an alert that indicates it is time to clean or replace the filter on the ventilator, if installed. You can set the Last Filter Change date, turn the Reminder On or Off, and set the Frequency of the maintenance interval.


node-red-qpyap.eu-gb.mybluemix.net/ui/#!/0?socketid=D7SQJMDY9BzQOYZdAAA6

ChatBot

Ask

Enter query
Thankyou chatty

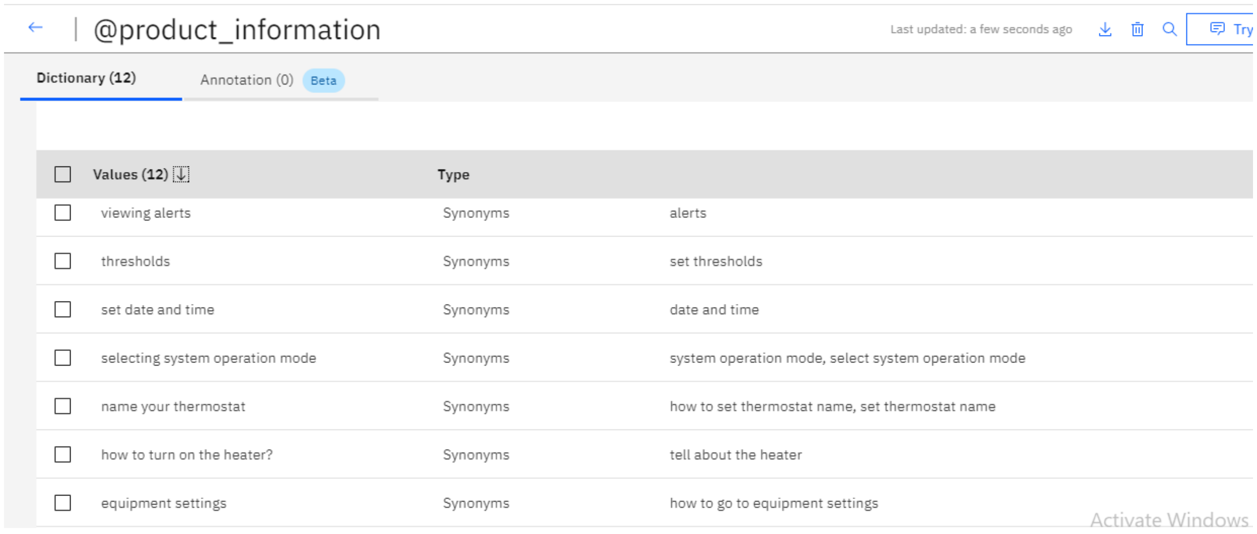
SUBMIT CANCEL

wait for a sec until it loads

Answer

Glad to be of your assistance.

Few queries of ecobee3 manual:



@product_information

Last updated: a few seconds ago

Dictionary (12) Annotation (0) Beta

Values (12)	Type	
viewing alerts	Synonyms	alerts
thresholds	Synonyms	set thresholds
set date and time	Synonyms	date and time
selecting system operation mode	Synonyms	system operation mode, select system operation mode
name your thermostat	Synonyms	how to set thermostat name, set thermostat name
how to turn on the heater?	Synonyms	tell about the heater
equipment settings	Synonyms	how to go to equipment settings

Activate Windows

7. ADVANTAGES & DISADVANTAGES

Advantages:

- Campanies can deploy chatbots to rectifiy simple and general human queries .
- Reduces man power
- Cost efficient
- No need to divert calls to customer agent and customer agent can look on other works.

Disadvantages:

- Some times chatbot can mislead customers
- Giving same answer for different sentiments.
- Some times cannot connect to customer sentiments and intentions.

8. APPLICATIONS

- It can deploy in popular social media applications like facebook,slack,telegram.
- Chatbot can deploy any website to clarify basic doubts of viewers.

9.CONCLUSION

By doing the abo ve procedure and all we successfully created Intelligent helpdesk smart char tbot using Watson assistant, Watson discovery, Node-RED and cloud-functions.

10.FUTURE SCOPE

We can include watson studio text to speech and speech to text services to access the chatbot handsfree. This is one of the future scope of this project.

11. BIBILOGRAPHY
APPENDIX

Source code:

"flows.json"

[{"id":"738c08d.58813f8","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"4dcf8fb0.38b58","type":"ui_form","z":"738c08d.58813f8","name":"","label":"","group":"c5142f35.7b43d","order":1,"width":0,"height":0,"options":[{"label":"Enter query","value":"input","type":"text","required":true,"rows":null}],"formValue":{"input":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":140,"y":120,"wires":[["cdb0daee.74b238"]]},{"id":"cdb0daee.74b238","type":"function","z":"738c08d.58813f8","name":"","func":"msg.payload=msg.payload.input\nreturn msg;","outputs":1,"noerr":0,"x":280,"y":220,"wires":[["70395f0b.7776e","989711db.c6535"]]},{"id":"70395f0b.7776e","type":"watson-conversation-v1","z":"738c08d.58813f8","name":"","workspaceid":"598cf10e-b40b-4081-9120-32101f929114","multiuser":false,"context":false,"empty-payload":false,"service-endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/a2b7744b-a0c2-447a-8c68-3e0d4e25e67e","timeout":"","optout-learning":false,"x":480,"y":180,"wires":[["55836ccc.3963a4","e02c9dea.570d6"]]},{"id":"55836ccc.3963a4","type":"debug","z":"738c08d.58813f8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":640,"y":80,"wires":[]},{"id":"e02c9dea.570d6","type":"function","z":"738c08d.58813f8","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.re sults[i].text;\n}\n   msg.payload=msg.payload.text;\n}\n\nelse\nmsg.payload = msg.payload.output.text[0];\nreturn msg;","outputs":1,"noerr":0,"x":660,"y":260,"wires":[["43572cb.312a2d4"]]},{"id":"43572cb.312a2d4","type":"ui_text","z":"738c08d.58813f8","group":"b4509f10.60649","order":1,"width":"11","height":"11","name":"","label":"","format":"{{msg.payload}}","layout":"col-center","x":600,"y":360,"wires":[]},{"id":"989711db.c6535","type":"ui_text","z":"738c08d.58813f8","group":"c5142f35.7b43d","order":2,"width":0,"height":0,"name":"","label":"wait for a sec until it loads","format":"","layout":"row-spread","x":340,"y":320,"wires":[]},{"id":"c5142f35.7b43d","type":"ui_group","z":"","name":"Ask","tab":"ea930e15.0c2b2","order":1,"disp":true,"width":"6","collapse":false},{"id":"b4509f10.60649","type":"ui_group","z":"","name":"Answer","tab":"ea930e15.0c2b2","order":2,"disp":true,"width":"11","collapse":false},{"id":"ea930e15.0c2b2","type":"ui_tab","z":"","name":"ChatBot","icon":"dashboard","disabled":false,"hidden":false}]

***Cloud function Node.js 10 code for discovery integration webhook generation:

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

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

    let discovery;

    if (params.iam_apikey){
      discovery = new DiscoveryV1({
        'iam_apikey': params.iam_apikey,
        'url': params.url,
        'version': '2019-03-25'
      });
    }
    else {
      discovery = new DiscoveryV1({
        'username': params.username,
        'password': params.password,
        'url': params.url,
        'version': '2019-03-25'
      });
    }

    discovery.query({
      'environment_id': params.environment_id,
      'collection_id': params.collection_id,
      'natural_language_query': params.input,
      'passages': true,
      'count': 3,
      'passages_count': 3
    }, function(err, data) {
      if (err) {
        return reject(err);
      }
      return resolve(data);
    });
  });
}
```