PROJECT REPORT

NAME :	Preeti Nair [preetinair369@gmail.com]
TITLE :	Intelligent Customer Help Desk With Smart Document Understanding.
CATEGORY :	Artificial Intelligence

Internship at smartinternz.com@2020

CONTENT

INTRODUCTION

1.1 *Overview*

The project is focused on creating a customer care chatbot whose dialog will provide a hook that can call out to other Watson services for

additional source of information unlike the typical chatbot which rely on predefined responses. In this case, an users manual has been uploaded in Watson Discovery service. This project utilizes different IBM services like Watson Discovery,
Cloud Functions, Watson Assistant and Node-Red.

1. **Project Requirements**: Intelligent customer care chatbot with smart document understanding feature given by Watson

Discovery which will lick to appropriate help context from the user manual. Watson Assistant to store all the dialogs, Node-Red tool to connect all the above mentioned services.
Functional Requirements: Each IBM services utilized have purpose in completing the project. Watson Discovery uses data analysis to take unstructured data and enrich it so you can query it for the information needed. Watson Assistant acts as the Chatbot which gives response to the customer's query. Cloud Functions is used to integrate cloud services. Node-Red is a programming tool which will wire together all the services mentioned above.

2. **Technical Requirements**: This includes Artificial Intelligence, Machine Learning, Python, IBM Watson Services.
3. **Software Requirements**: IBM Watson services like Watson

Discovery, Watson Assistant.

4. **Project Deliverables**: This chatbot is created to improve customer experience and to show how various services are involved to get a an accurate and desired result.

5. **Project Team**: Individual project work [Preeti Nair].

6. **Project Duration**: 29 days.

1.2 ***Purpose***

Chatbots aren't just for customer. They can provide useful support throughtout a business, including service desk. From organisation to simplication to satisfcation, there are many purposes of creating chatbot with smarter service desk.

With feature like Smart Document Understanding (SDU), the customer won't have to undergo the tedious procedure of connecting with customer care representative. Instead the answer of his query will be displayed through the chatbot. The solution provided will be accurate and handy as it is straight from a owner's manual which has been pre-loaded in Watson Discovery.
It can direct the user to relevant available content. By taking in keywords and understanding syntax, bots are then able to suggest useful help rescources.
The response will be lightnening fast and the users can implement on the advice provided by the bot and get on with their day.

To summarize, the purpose of this chatbot is to enhance customer experience by integrating services provided by IBM like
Watson Discovery, Watson Assistant, Node-Red and Cloud Functions.

LITERATURE SURVEY

2.1 ***Existing Problem***

A standard chatbot will provide solution for basic queries like "location of store" or "when does the store open". But when the query is something related to the operation of a device, instead of displaying the solution it will direct the customer to a customer representative.
Due to this, the customer has to wait until the representative gets back to them.
Sometimes they don't provide useful solution to the query.
Another problem is these chatbots have limited responses for customers.
This overall deteriorates the customer experinece.

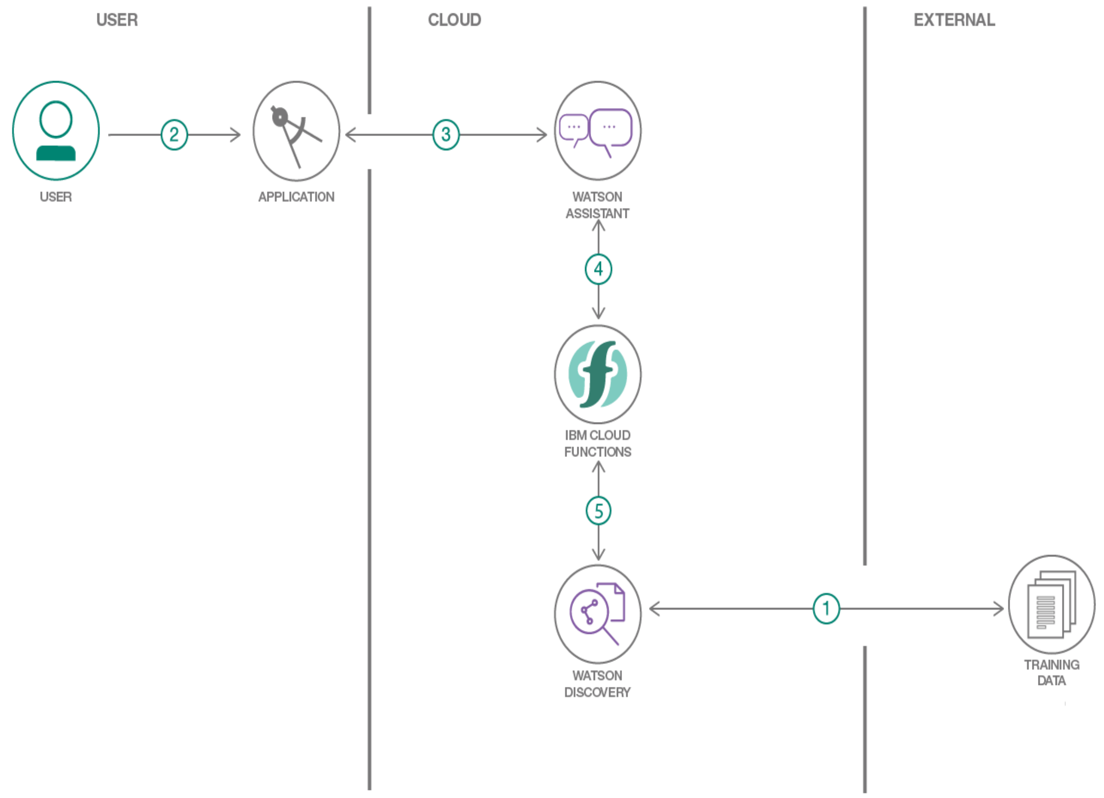
2.1 ***Proposed Solution***

The proposed solution is to create a chatbot that will display the appropriate solution instead of involving any human interaction.
If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" it will return relevant sections of the owners manual to help solve our customers' problems.

In this way the customer won't have to undergo the tedious process of connecting with the representative and the customer will get lightnening fast response. This will thereby enhance customer service experience.

THEORETICAL ANALYSIS

3.1 ***BLOCK DIAGRAM***



- **Step 1** : The document in this case the owner's manual is indexed and annotated by the Smart Document Understanding feature of Watson Discovery.
- **Step 2** : The user interacts with the app UI. The frontend server is the chatbot which engages with the customer with small talk.
- **Step 3** : The interaction is coordinated by Watson Assistant service.

- **Step 4** : When a technical query comes into picture, the assistant invokes the Cloud Functions action.
- **Step 5** : The action then queries the Discovery and returns with the result relevant to the query.

3.2 ***Hardware/Software Designing***

The softwares utilized are Watson Discovery, Watson Assistant and Node-Red.

1] Watson Discovery : Discovery makes it possible to rapidly build cognitive, cloud-based exploration applications that unlock actionable insights hidden in unstructured data. It applies latest breakthroughs in machine learning, including natural language processing capabilities, and is easily trained on the language of your domain. It breaks open the data silos and retrieves specific answers to user's questions while analysing trends and relationships buried in the uploaded data. With Smart Document Understanding (SDU) feature you can train IBM Watson Discovery to extract custom fields in your documents.

2] Watson Assistant : It is conservation AI platform that provides customer fast, straightforward and accurate answers to their question, across any application, device or channel. By addressing common customer inquiries, Watson Assistant reduces the cost of customer interactions.
It uses Watson AI machine learning (ML) and natural language understanding (NLU). The user input is recieved by the assistant and routes it to a dialog skill. The dialog skill interprets the input further, then directs the flow of the conservatio. The dialog gathers any information needs to respond on user's behalf.

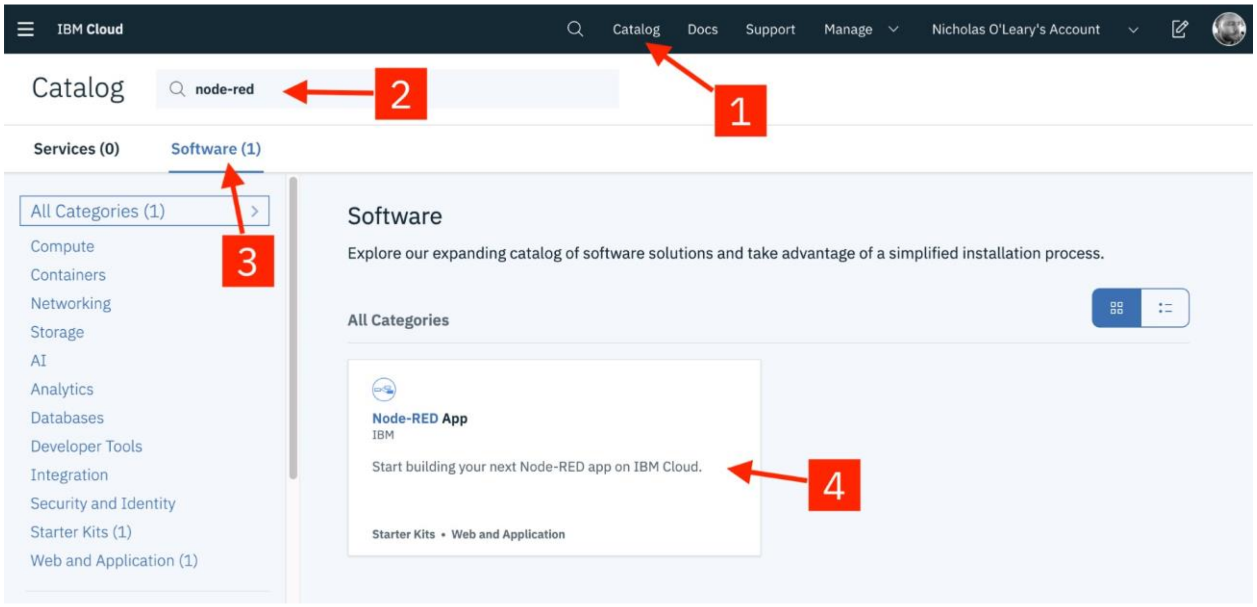
3] Node-Red : It is a prototyping tool that builds applications easily. Applications are developed by building data flows through a series of connected nodes. Intricate apllications can be built that allows Watson services to interact with range of capabilities and services exposed as Node-Red nodes.
It helps in wiring together hardware devices, APIs and online services without writing any code. It provides web based flow editor, which can be used to connect these services and also create UI.

EXPERIMENTAL INVESTIGATIONS

Following services are to be created :

4.1 *Node-Red Application*

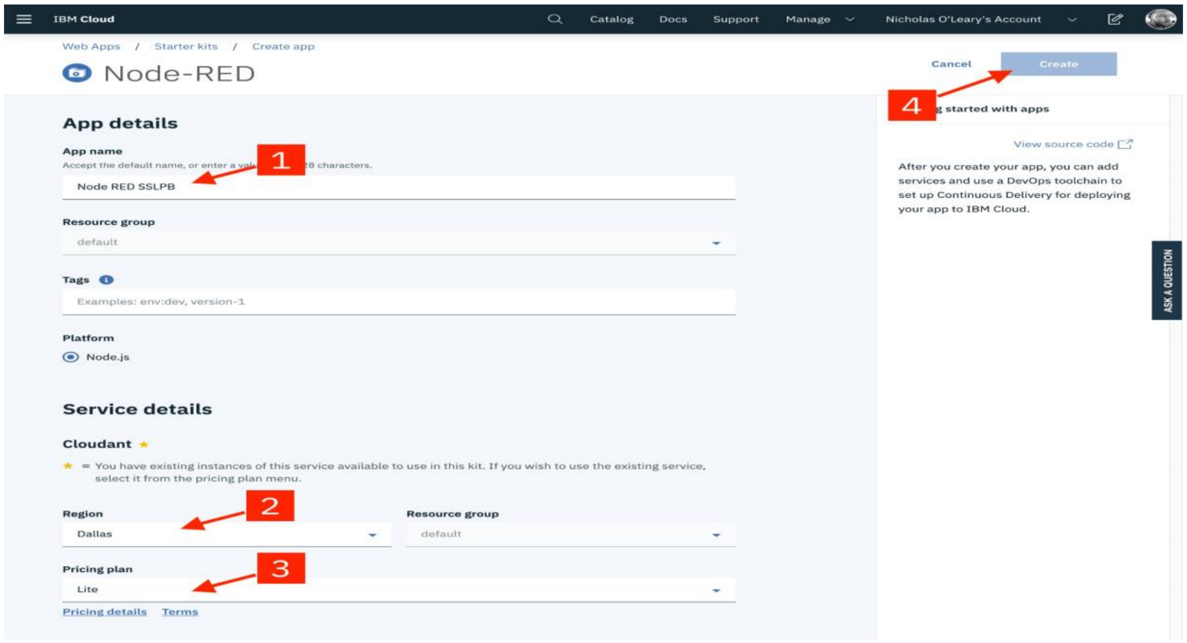
1. Search for Node-Red in catalog and create an application by filling in details like the cloudant region and a unique name.



Software

Explore our expanding catalog of software solutions and take advantage of a simplified installation process.

Start building your next Node-RED app on IBM Cloud.



App details

App name: Node RED SSLPB

Resource group: default

Tags: Examples: env:dev, version-1

Platform: Node.js

Service details

Cloudant

★ = You have existing instances of this service available to use in this kit. If you wish to use the existing service, select it from the pricing plan menu.

Region: Dallas

Resource group: default

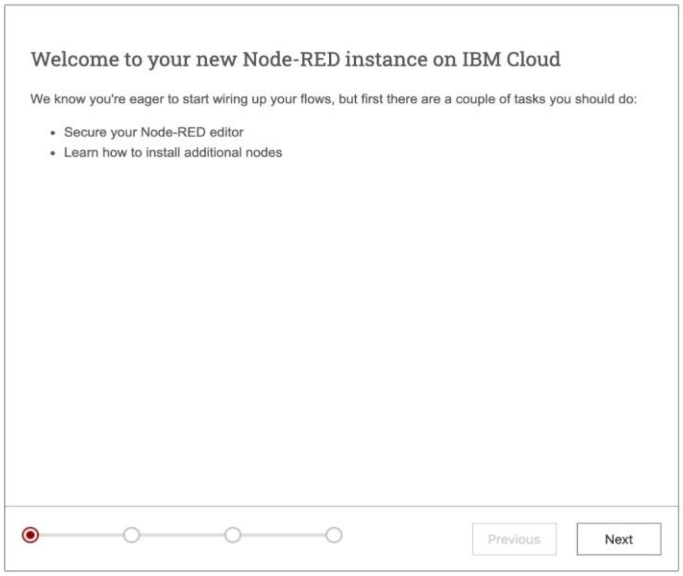
Pricing plan: Lite

After you create your app, you can add services and use a DevOps toolchain to set up Continuous Delivery for deploying your app to IBM Cloud.

2. Setup Continous Delivery feature to deploy the application created into the Cloud foundary apps in IBM Cloud.

After deploying stage is completed, the application is now running.

3. In the details page, click on Visit App URL to configure it and set up security.

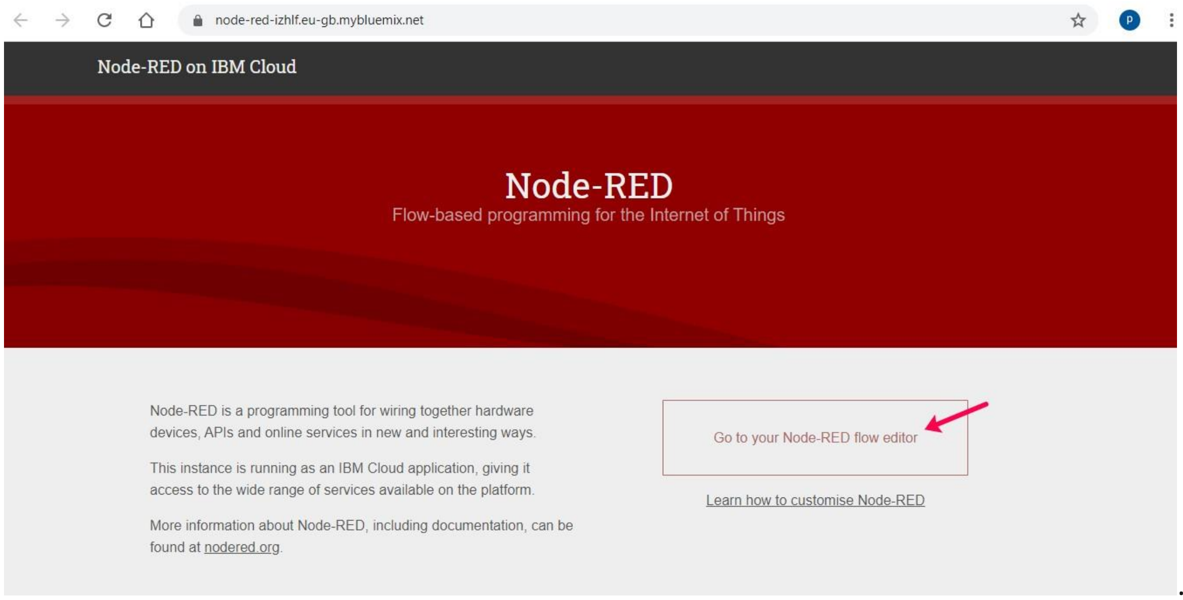


Welcome to your new Node-RED instance on IBM Cloud

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
- Learn how to install additional nodes

4. Click on Next. Now it will ask for username and password for security reasons. Enter them.
5. The final screen summarizes the options you've made and highlights the environment variables you can use to change the options in the future. Click Finish to proceed.
6. Node-RED will save your changes and then load the main application. From here you can click the Go to your Node-RED flow editor button to open the editor



node-red-izhlf.eu-gb.mybluemix.net

Node-RED on IBM Cloud

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.

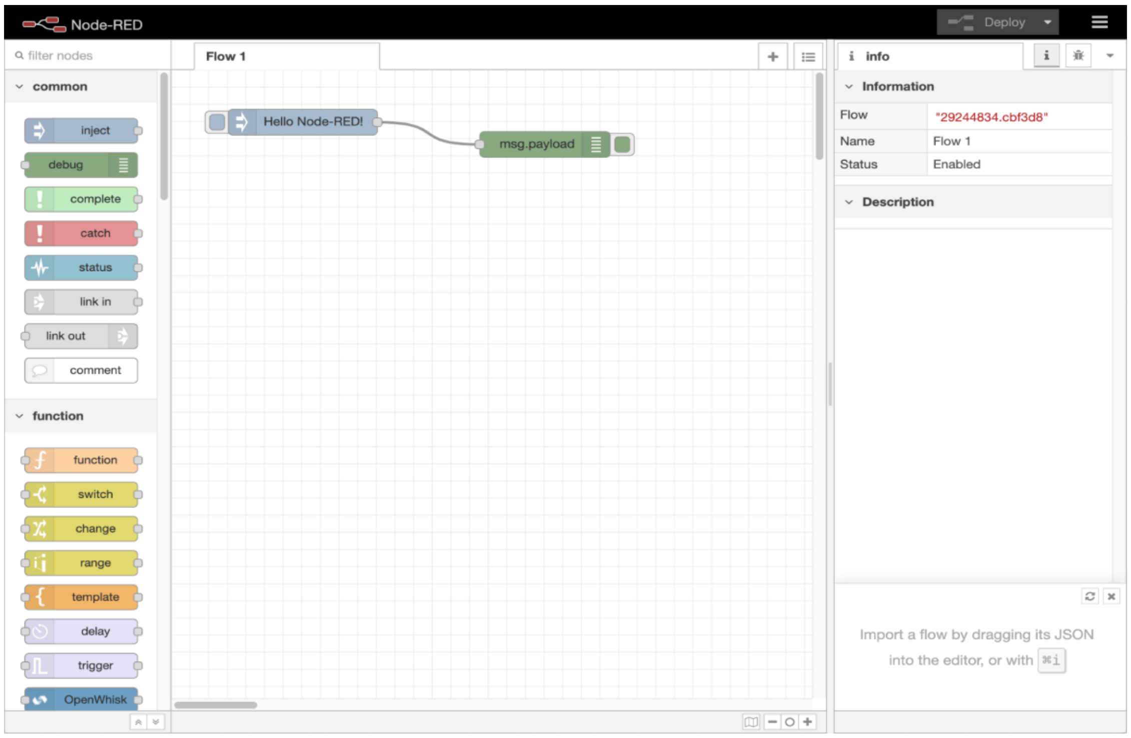
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

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

.

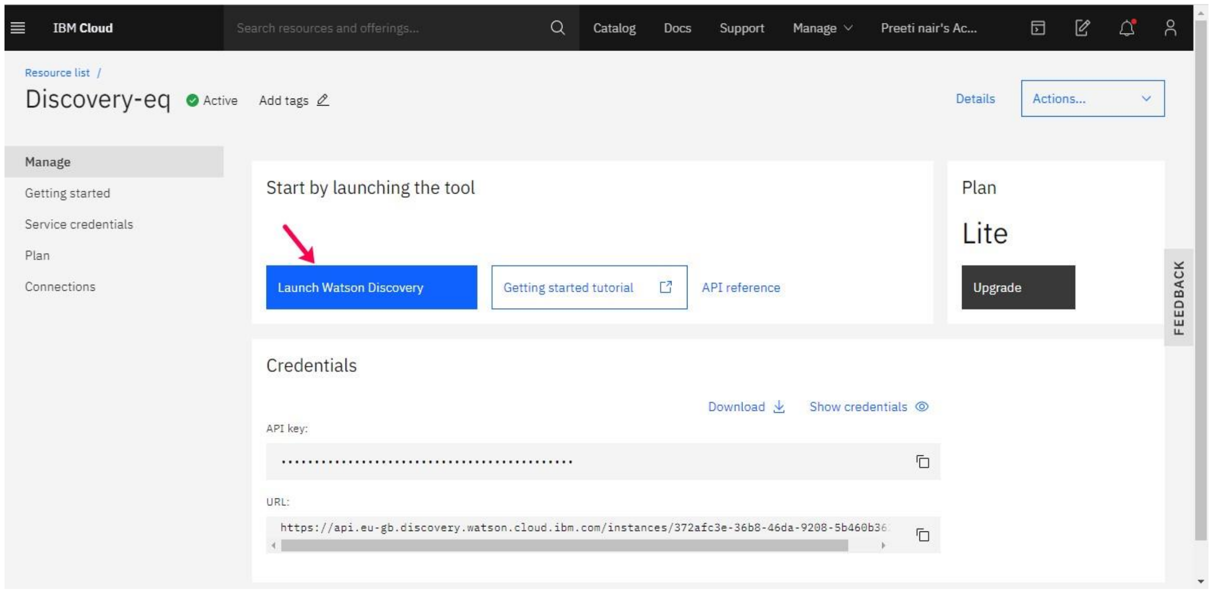
7. This is the place where you will create flow to wire all the services utilized to function the chatbot.



Flow	"29244834.cbf3d8"
Name	Flow 1
Status	Enabled

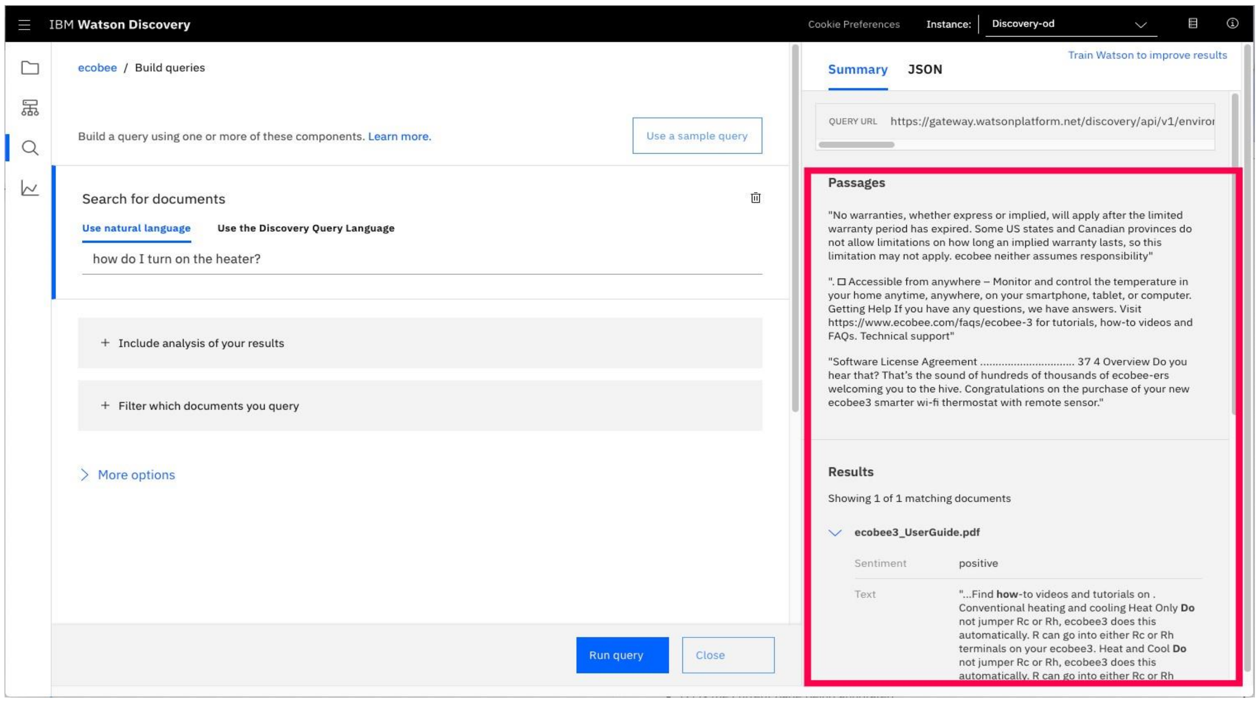
4.2 ***Watson Discovery***

1. Search for Discovery from Catalog and create an instance of the service.
2. After creating, launch the service by clicking on Launch Watson Discovery.



API key:
```
..............................................
```
URL:
```
https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/372afc3e-36b8-46da-9208-5b460b36
```

3. Upload the ecobee3 owner's manual by clicking on <u>Upload your own data</u>.
4. Now before setting up the SDU feature, try few queries. Click on the <u>Build your own query.</u>
5. When ask a query, it will display you many sections from the manual. To improve query result, setup SDU feature.



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

Search for documents

how do I turn on the heater?

+ Include analysis of your results

+ Filter which documents you query

> More options

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

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."

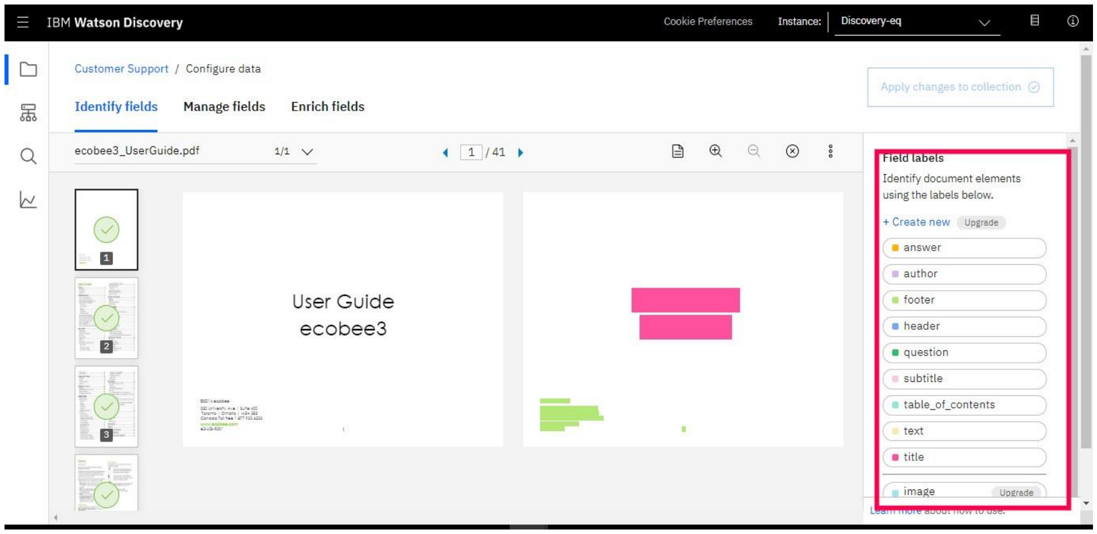
Results

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

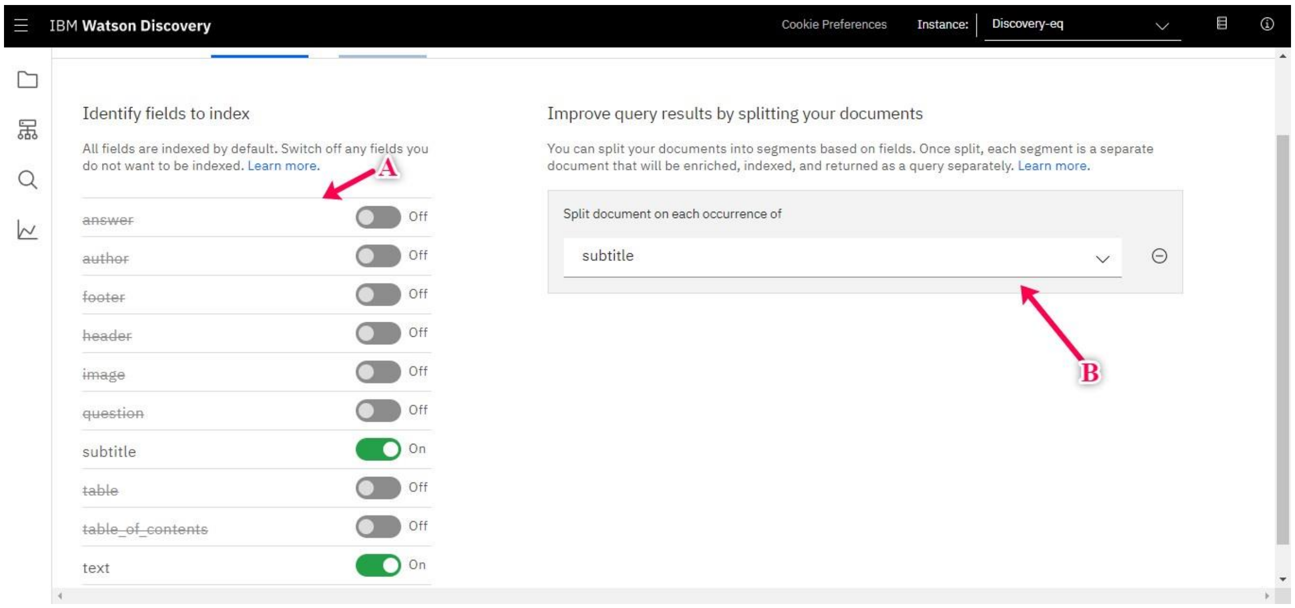
6. Go to Configure data. You will see SDU layout. There are three fields : **Identify fields, Manage fields** and **Enrich fields.**
7. In Identify field, here you will identify the elements in the document by labelling them as **title , subtitle** or **text** which you will see on the right side of the layout page.



Field labels

Identify document elements using the labels below.

8. Assign relevant labels for 10-11 pages and click on Submit Page accordingly. After few pages, Discovery starts to recognize the element automatically. This means the service is trained.
9. Now go to Manage Fields tab, select the items you want to be indexed (A). After that, split the document by subtitle (B). Click the **Apply to changes button**.



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

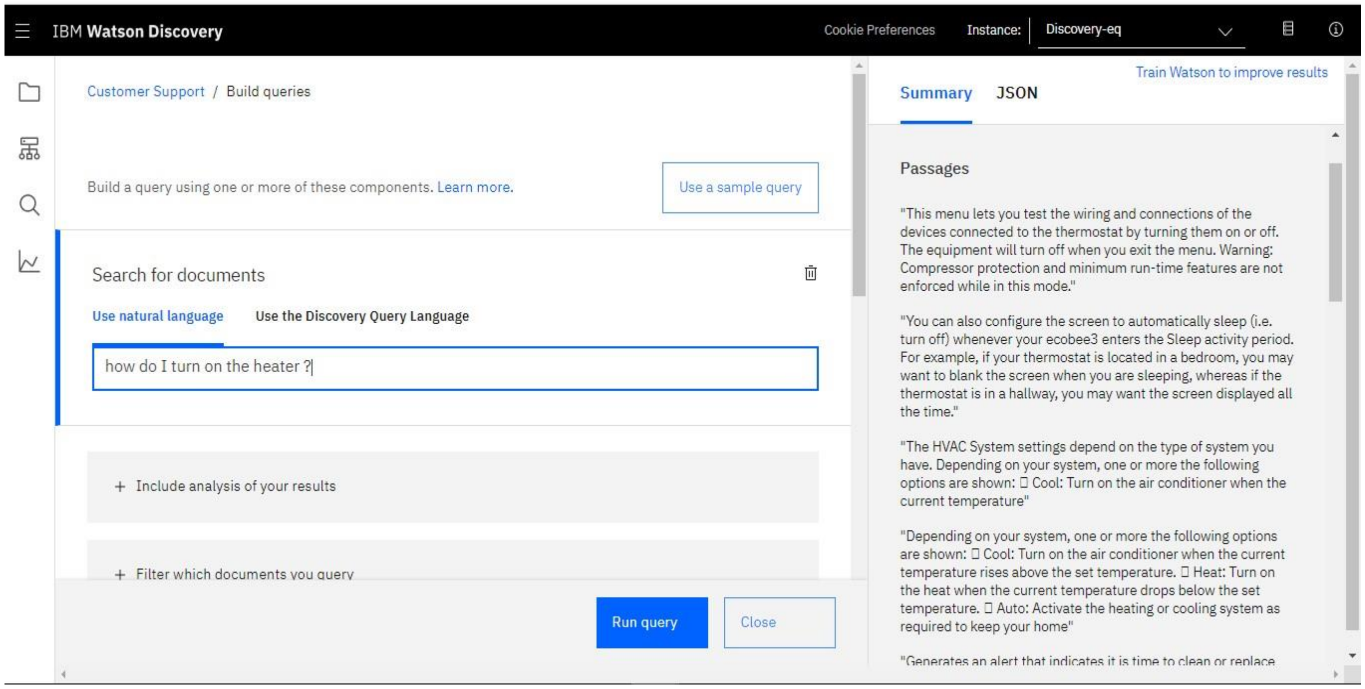
Improve query results by splitting your documents

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

9. It will ask you to upload the document, upload it. You will see all the annotations applied to your index.

10. Again click on **Build your own query.** Ask the same query. This time you will see the most relevant answer from the manual being displayed.


IBM Watson Discovery
Cookie Preferences
Instance: Discovery-eq
Customer Support / Build queries
Use a sample query
Use natural language
Use the Discovery Query Language
+ Include analysis of your results
+ Filter which documents you query
Run query
Close
Train Watson to improve results
Summary
JSON

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

Search for documents

how do I turn on the heater ?

Passages

"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."

"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."

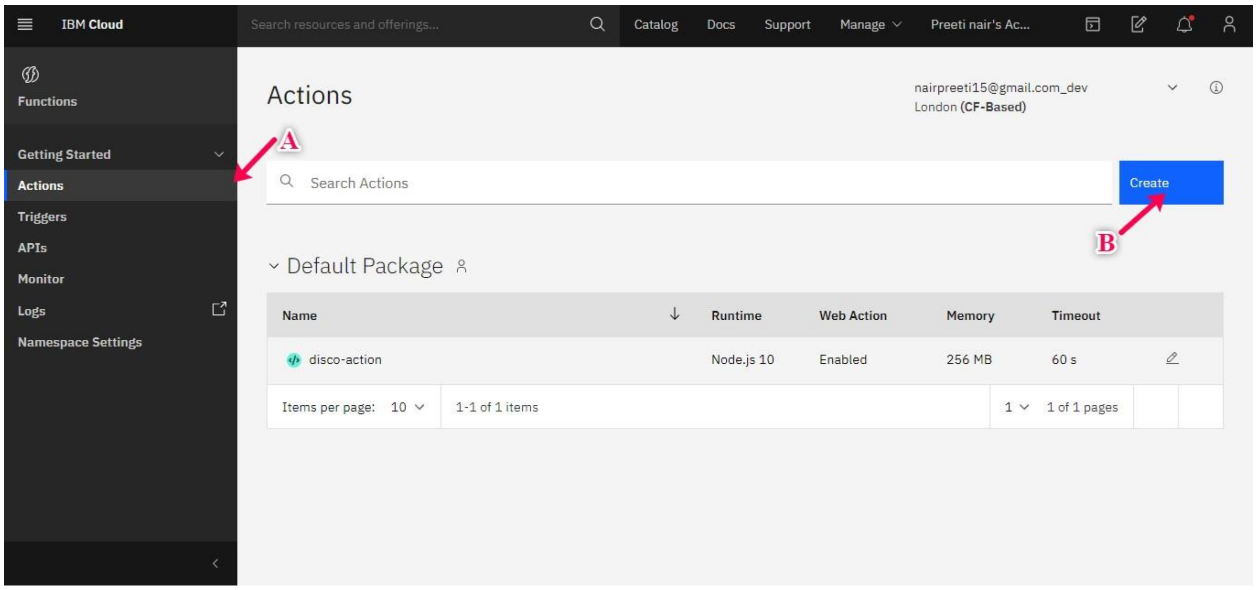
"The HVAC System settings depend on the type of system you have. Depending on your system, one or more the following options are shown: Cool: Turn on the air conditioner when the current temperature"

"Depending on your system, one or more the following options are shown: Cool: Turn on the air conditioner when the current temperature rises above the set temperature. Heat: Turn on the heat when the current temperature drops below the set temperature. Auto: Activate the heating or cooling system as required to keep your home"

"Generates an alert that indicates it is time to clean or replace

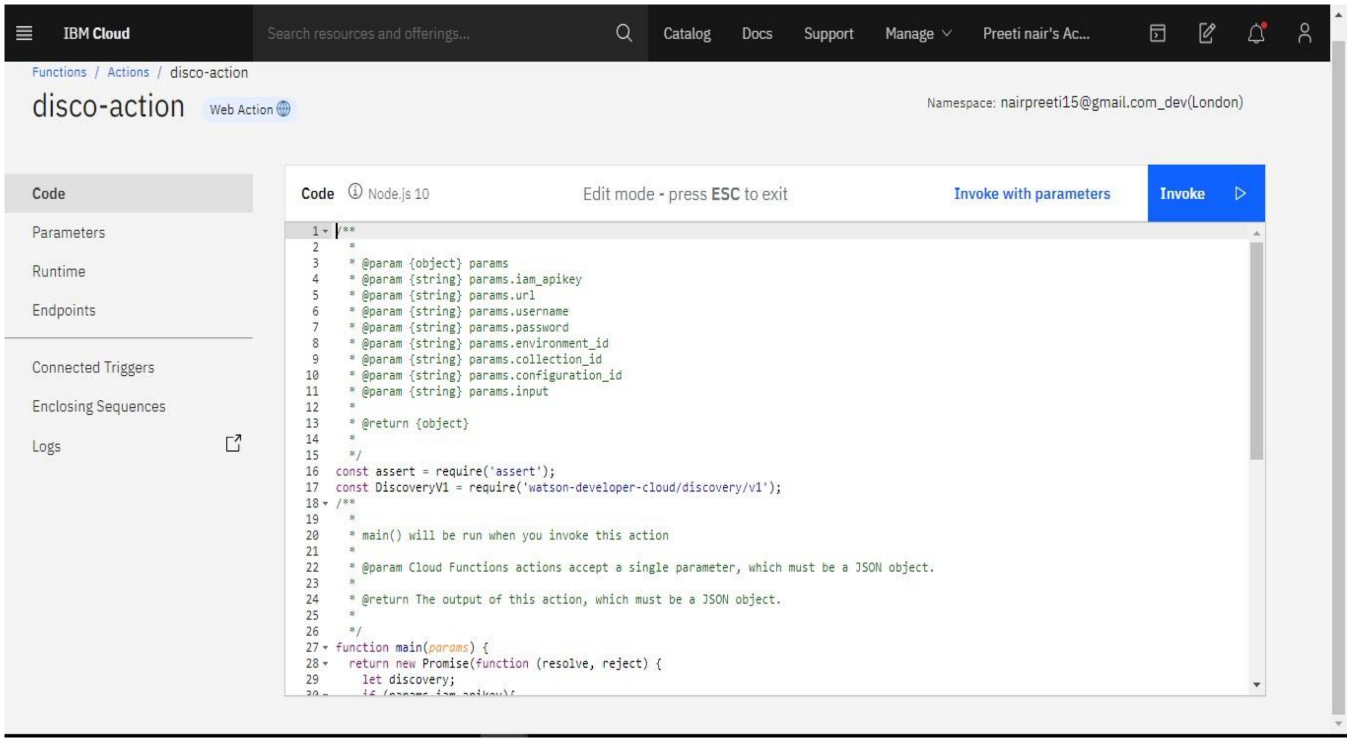
4.3 *Cloud Functions*

1. Search for Functions in catalog and click on Functions.
2. Opening Functions, in the left tab click on **Actions** (A) and select create (B).



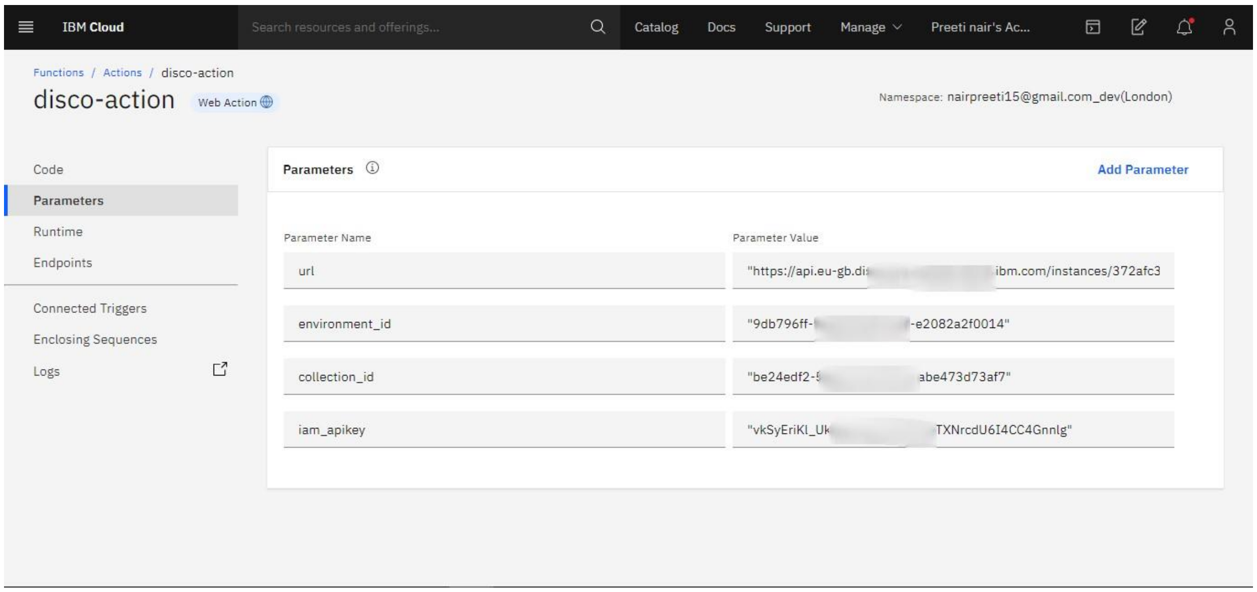
Name	Runtime	Web Action	Memory	Timeout
disco-action	Node.js 10	Enabled	256 MB	60 s

3. Once the action is being created, go to code tab. Paste the code which will revoke the Discovery service.



```
/**
  *
  * @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;
```

4. Before clicking on **Invoke** button, click on Parameters tab and add all necessary parameters of Discovery service which you will find on following pages.

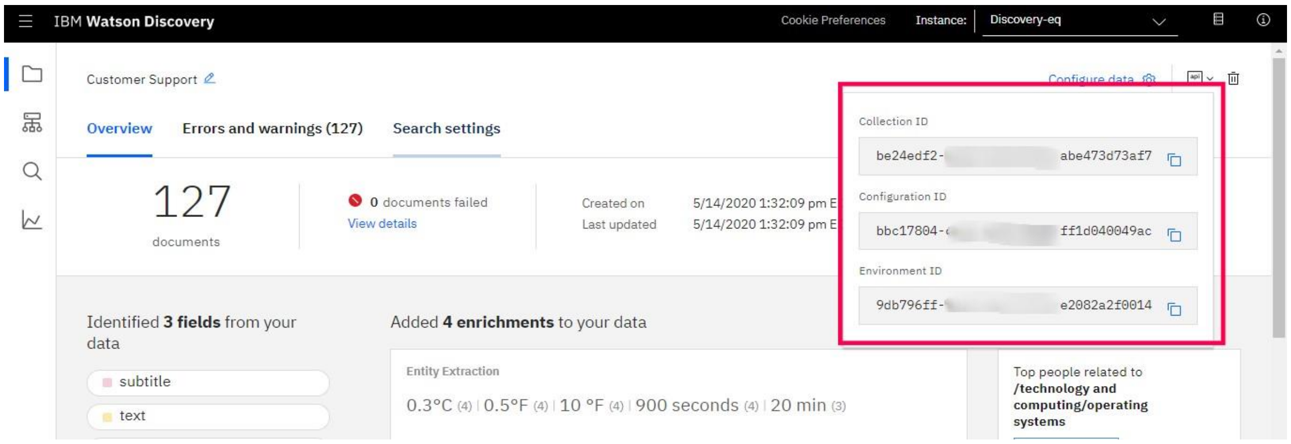


disco-action

Namespace: nairpreeti15@gmail.com_dev(London)

Parameters

Parameter Name	Parameter Value
url	"https://api.eu-gb.dis[redacted]ibm.com/instances/372afc3
environment_id	"9db796ff-[redacted]-e2082a2f0014"
collection_id	"be24edf2-[redacted]abe473d73af7"
iam_apikey	"vkSyEriKl_Uk[redacted]TXNrcdU6I4CC4Gnnlg"



127 documents

0 documents failed

Created on 5/14/2020 1:32:09 pm E

Last updated 5/14/2020 1:32:09 pm E

Collection ID: be24edf2-[redacted]abe473d73af7

Configuration ID: bbc17804-[redacted]ff1d040049ac

Environment ID: 9db796ff-[redacted]e2082a2f0014

Identified **3 fields** from your data

subtitle

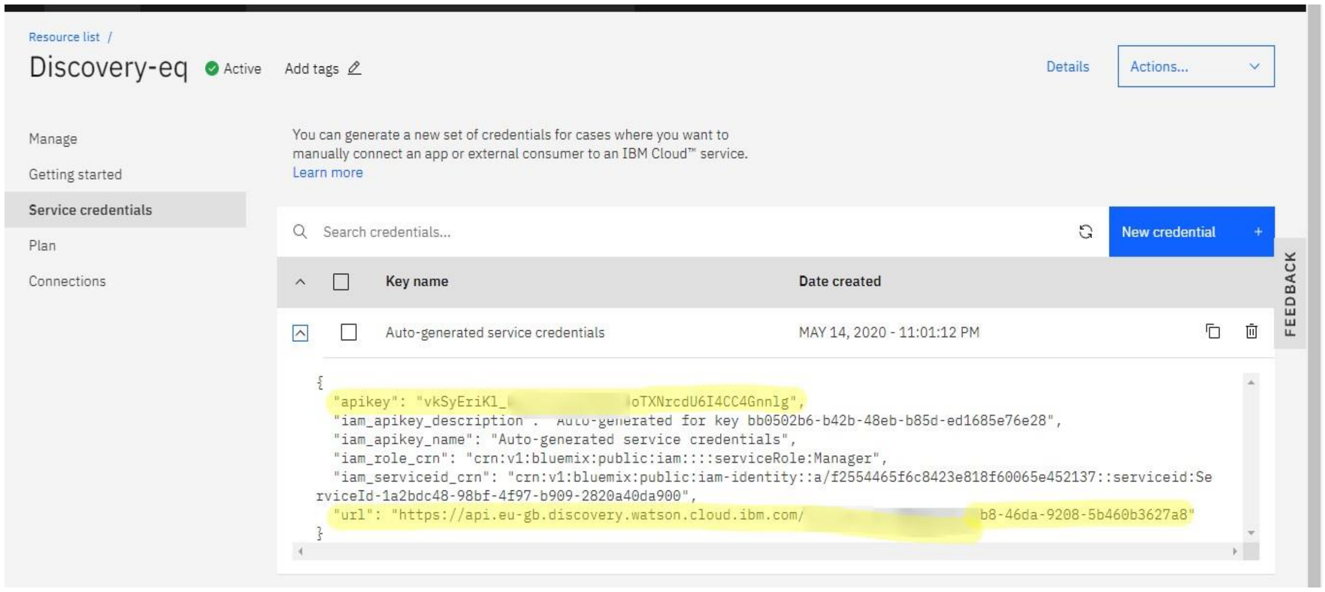
text

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)

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



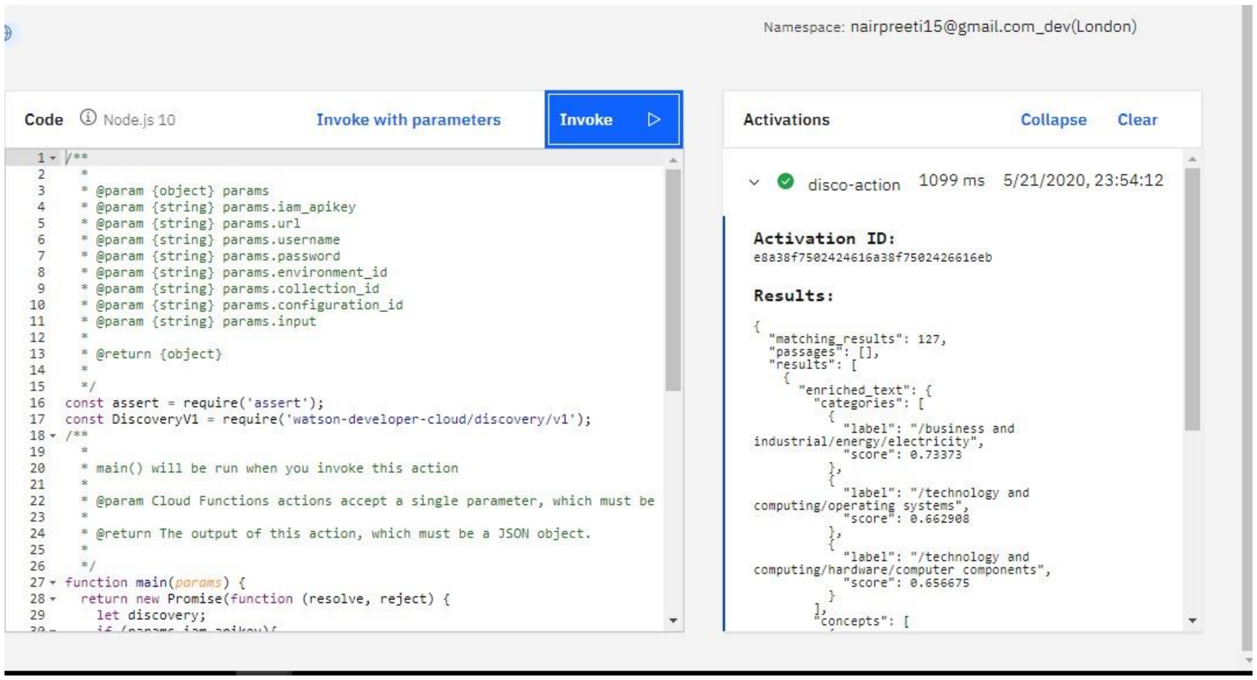
Discovery-eq

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud™ service.
Learn more

Key name	Date created
Auto-generated service credentials	MAY 14, 2020 - 11:01:12 PM

```
{
  "apikey": "vkSyEriKl_[redacted]oTXNrcdU6I4CC4Gnnlg",
  "iam_apikey_description": "Auto-generated for key bb0502b6-b42b-48eb-b85d-ed1685e76e28",
  "iam_apikey_name": "Auto-generated service credentials",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/f2554465f6c8423e818f60065e452137::serviceid:Se
rviceId-1a2bdc48-98bf-4f97-b909-2820a40da900",
  "url": "https://api.eu-gb.discovery.watson.cloud.ibm.com/[redacted]b8-46da-9208-5b460b3627a8"
}
```

5. After adding parameters, go back to code page and click on **Invoke**. You will see the actual results returned from the Discovery service.



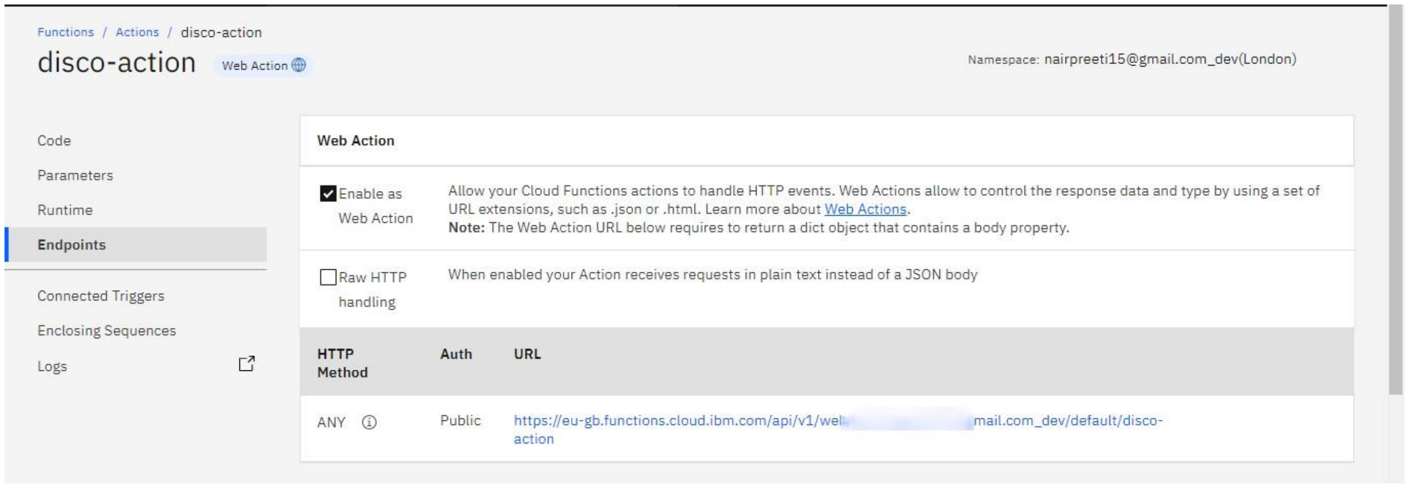
```
/**
 *
 * @param {object} params
 * @param {string} params.iam_apikey
 * @param {string} params.url
 * @param {string} params.username
 * @param {string} params.password
 * @param {string} params.environment_id
 * @param {string} params.collection_id
 * @param {string} params.configuration_id
 * @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
 *
 * @return The output of this action, which must be a JSON object.
 *
 */
function main(params) {
  return new Promise(function (resolve, reject) {
    let discovery;
```

```
disco-action   1099 ms   5/21/2020, 23:54:12

Activation ID:
e8a38f7502424616a38f7502426616eb

Results:
{
  "matching_results": 127,
  "passages": [],
  "results": [
    {
      "enriched_text": {
        "categories": [
          {
            "label": "/business and
industrial/energy/electricity",
            "score": 0.73373
          },
          {
            "label": "/technology and
computing/operating systems",
            "score": 0.662908
          },
          {
            "label": "/technology and
computing/hardware/computer components",
            "score": 0.656675
          }
        ],
        "concepts": [
```

6. In Cloud Functions, click on Endpoints tab, and select **Enable as Web Action**. This will create a url which will be used to setup Webhook in Watson Assistant.



Web Action

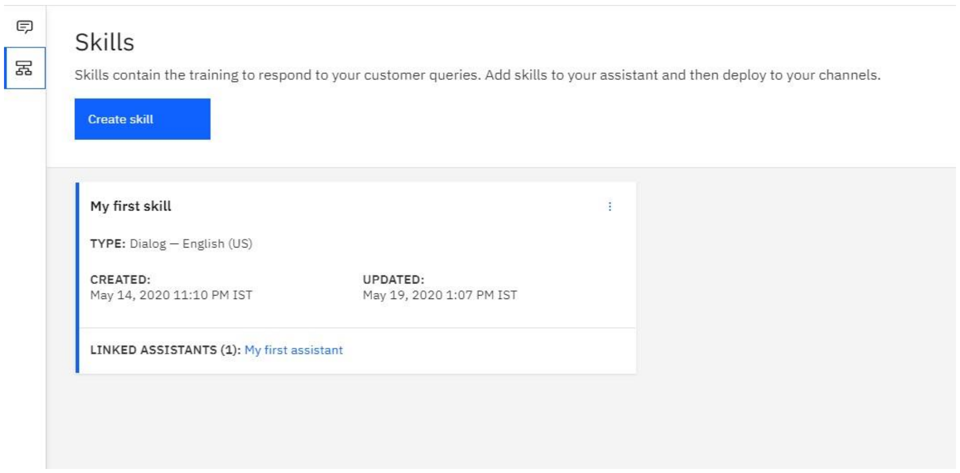
☑ Enable as Web Action — Allow your Cloud Functions actions to handle HTTP events. Web Actions allow to control the response data and type by using a set of URL extensions, such as .json or .html. Learn more about Web Actions.
Note: The Web Action URL below requires to return a dict object that contains 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/wel[redacted]mail.com_dev/default/disco-action

4.4 ***Watson Assistant***

1. Create Watson Assistant service in similar fashion as Watson Discovery. Launch the service.
2. When you open the service, you will see a dialog skill named Customer Care skill automatically provided by the service. Either choose that and remodify it or create new skill.


Skills

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

Create skill

My first skill

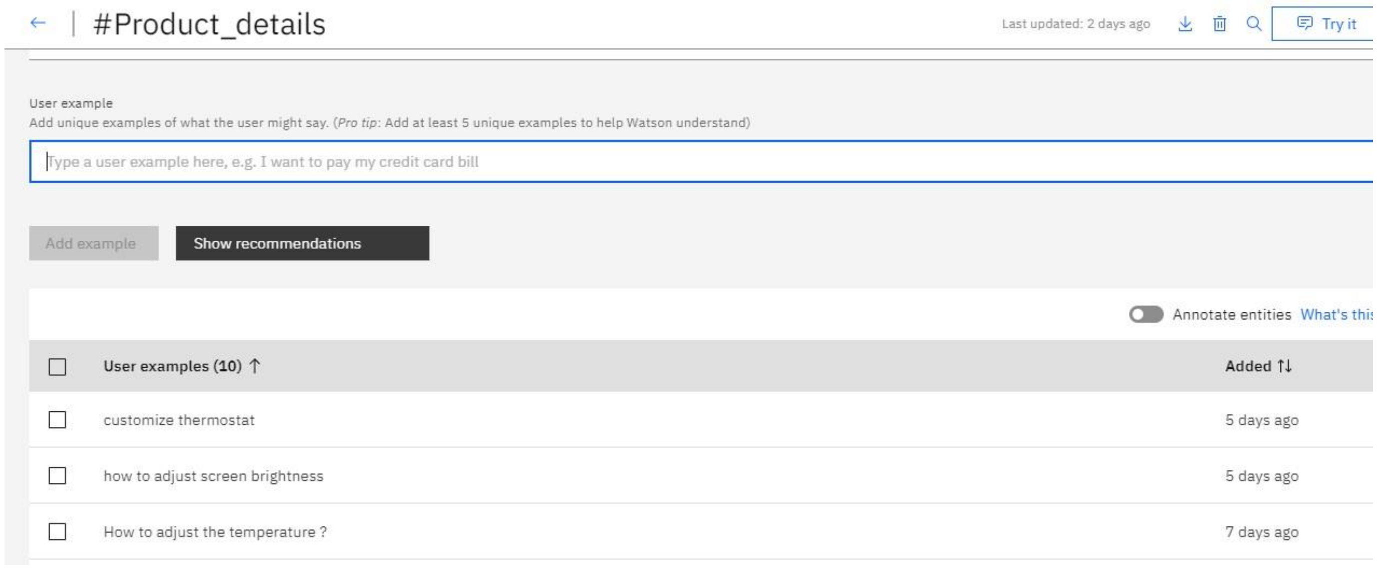
TYPE: Dialog — English (US)

CREATED: May 14, 2020 11:10 PM IST

UPDATED: May 19, 2020 1:07 PM IST

LINKED ASSISTANTS (1): My first assistant

3. Firstly create intent which we will detect if the customer is asking about the operation of the product. Given down below is an intent about Product_details. Provide few sample examples of queries for the assistant to detect.


#Product_details

Last updated: 2 days ago

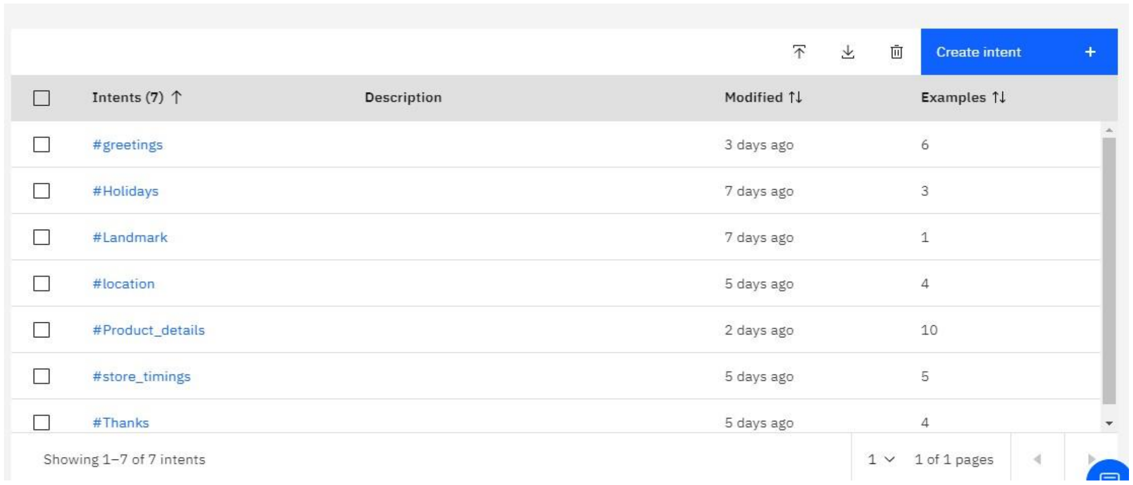
User example

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

User examples (10)	Added
customize thermostat	5 days ago
how to adjust screen brightness	5 days ago
How to adjust the temperature ?	7 days ago

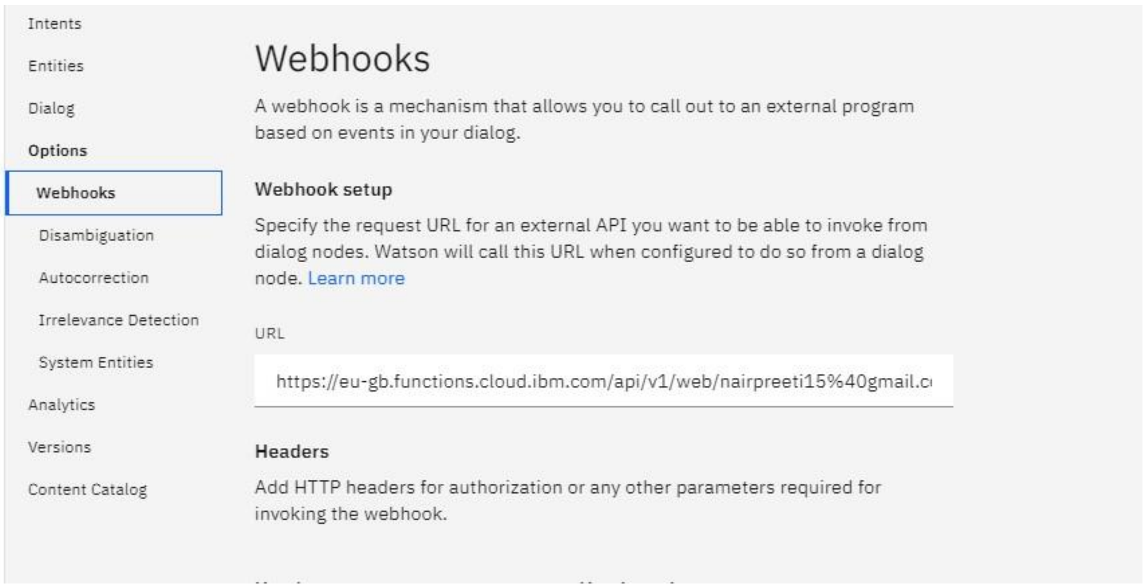
Similarly you can give intents for location, landmarks or timings.



Intents (7)	Description	Modified	Examples
#greetings		3 days ago	6
#Holidays		7 days ago	3
#Landmark		7 days ago	1
#location		5 days ago	4
#Product_details		2 days ago	10
#store_timings		5 days ago	5
#Thanks		5 days ago	4

Showing 1–7 of 7 intents   1   1 of 1 pages

4. Click on Options. Here you will setup Webhook. Use the url from the Web action of Cloud functions in Endpoints 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

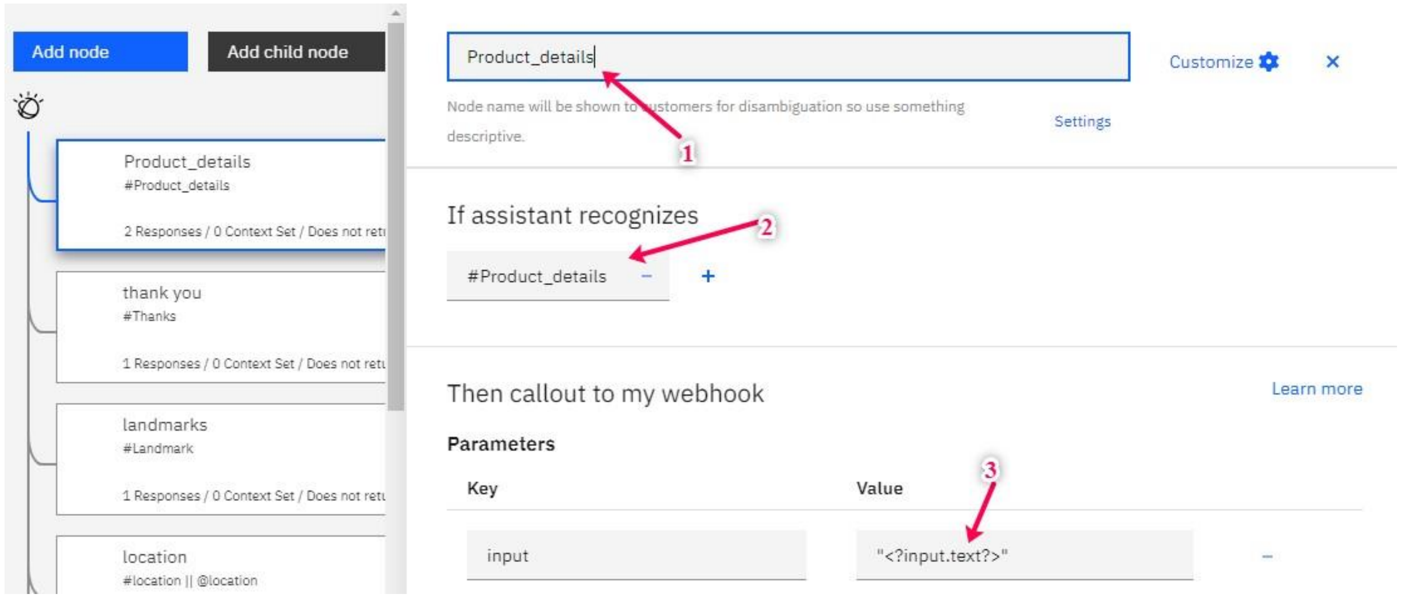
URL

https://eu-gb.functions.cloud.ibm.com/api/v1/web/nairpreeti15%40gmail.c

Headers

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

5. Move onto Dialog tab. Here, add nodes to handle intents.
6. Create a node for Product_details (1) and accordingly assign its intent (2).
7. Enter syntax (3) which will pass on the query via **input** parameter to discovery.



Product_details

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

If assistant recognizes

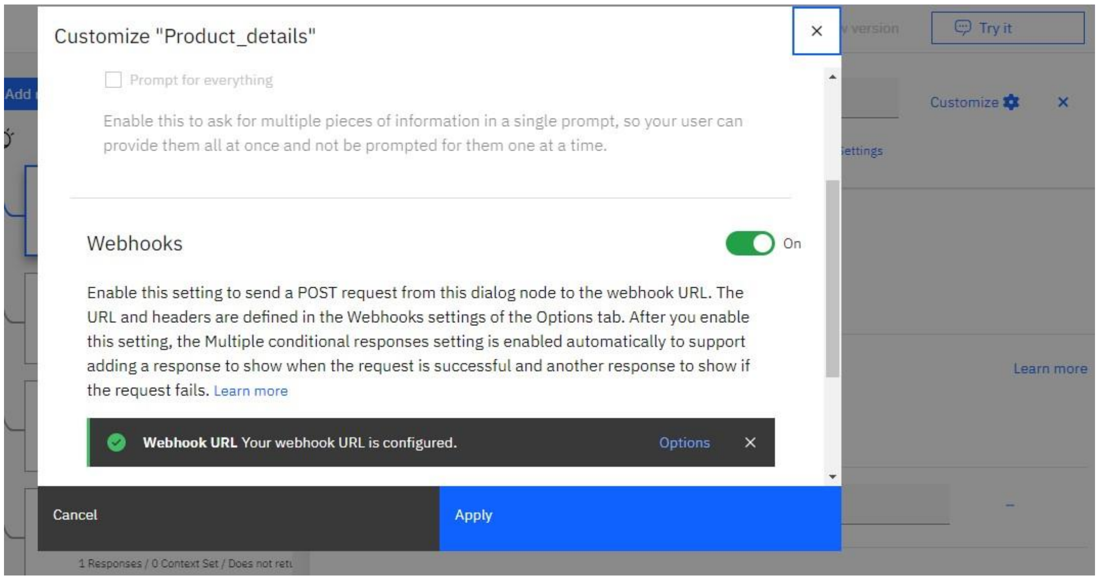
#Product_details

Then callout to my webhook

Parameters

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

8. Click on Customize and enable the Webhook.


Customize "Product_details"

Prompt for everything

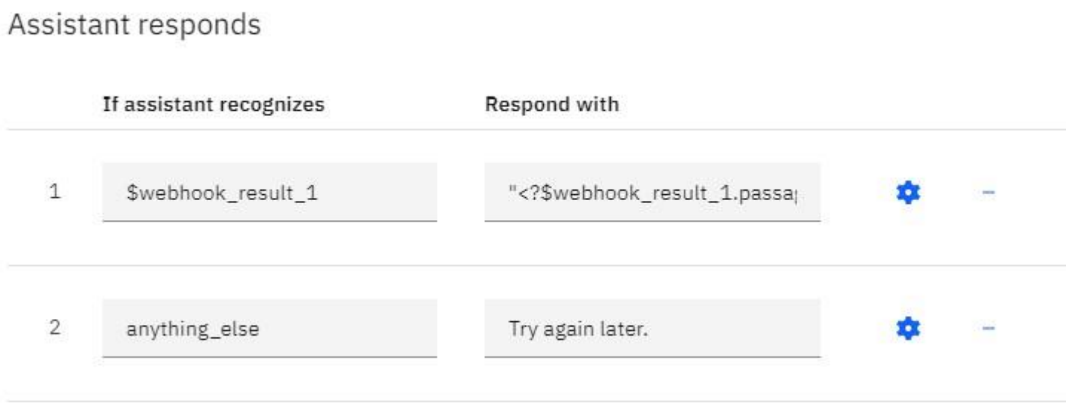
Enable this to ask for multiple pieces of information in a single prompt, so your user can provide them all at once and not be prompted for them one at a time.

Webhooks On

Enable this setting to send a POST request from this dialog node to the webhook URL. The URL and headers are defined in the Webhooks settings of the Options tab. After you enable this setting, the Multiple conditional responses setting is enabled automatically to support adding a response to show when the request is successful and another response to show if the request fails. Learn more

Webhook URL Your webhook URL is configured. Options

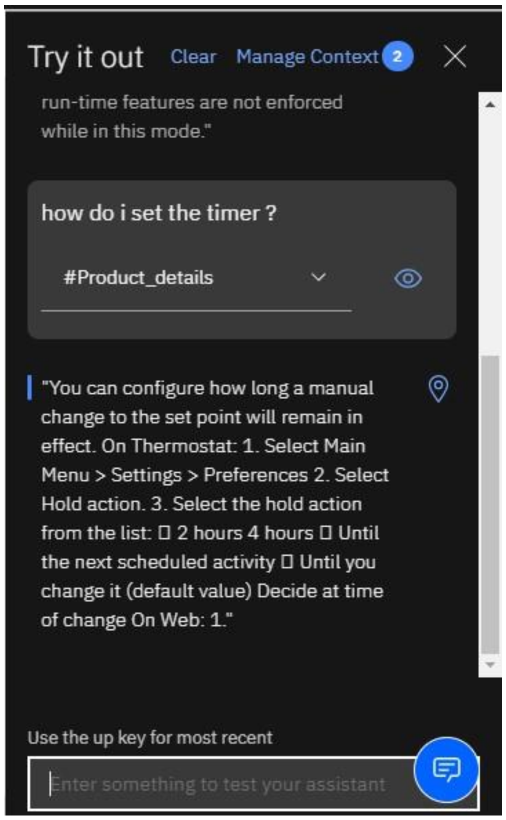
9. Now add the reponses to aid in debugging.



Assistant responds

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

10. Try out the assistant.



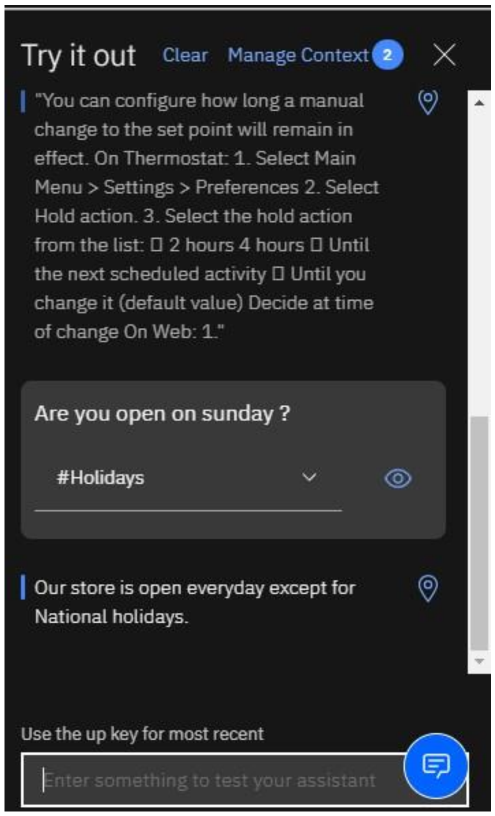
run-time features are not enforced while in this mode."

how do i set the timer ?

#Product_details

"You can configure how long a manual change to the set point will remain in effect. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Hold action. 3. Select the hold action from the list: 2 hours 4 hours Until the next scheduled activity Until you change it (default value) Decide at time of change On Web: 1."

Use the up key for most recent
Enter something to test your assistant


Try it out
Clear
Manage Context 2

"You can configure how long a manual change to the set point will remain in effect. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Hold action. 3. Select the hold action from the list: 2 hours 4 hours Until the next scheduled activity Until you change it (default value) Decide at time of change On Web: 1."

Are you open on sunday ?

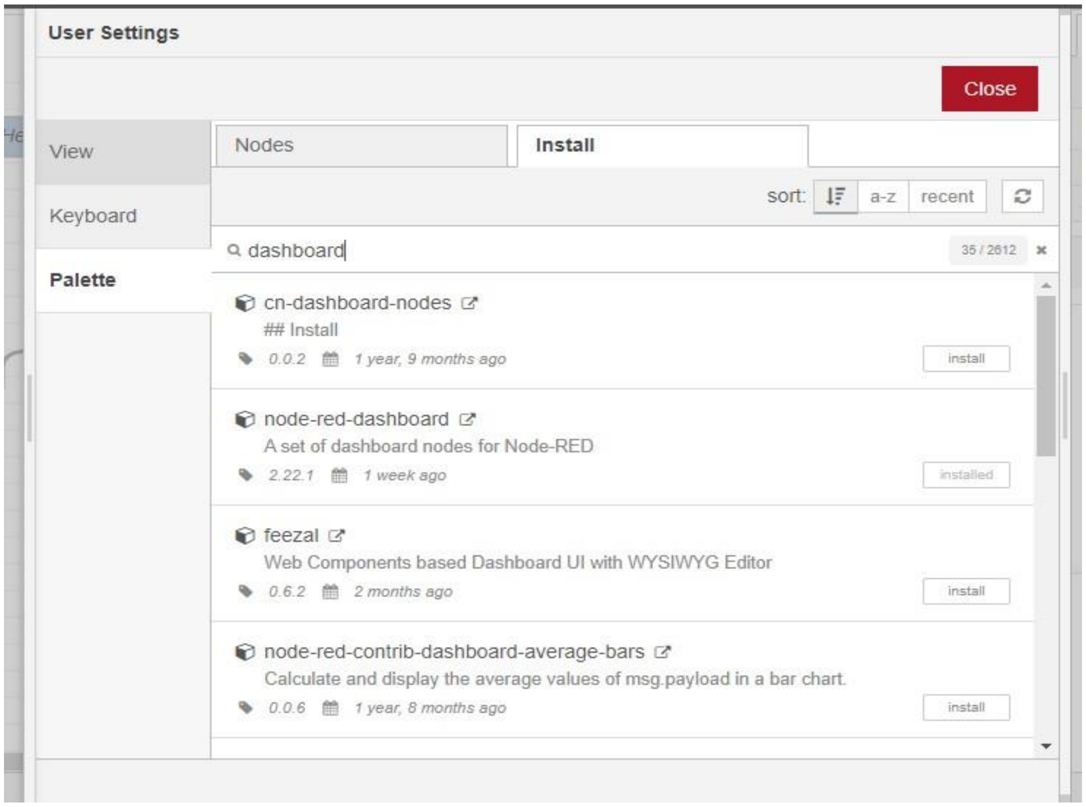
#Holidays

Our store is open everyday except for National holidays.

Use the up key for most recent
Enter something to test your assistant

<u>Integration of all the above services in NODE-RED</u>

1. Open the Node-Red Flow editor. Before beginning, download dashboard nodes used for creating chatbot UI. Click on **Manage Palette.** Go to install tab and search for node-red-dashboard. Then click on install.


User Settings

Close

View

Keyboard

Palette

Nodes | Install

sort: a-z recent

dashboard 35 / 2612

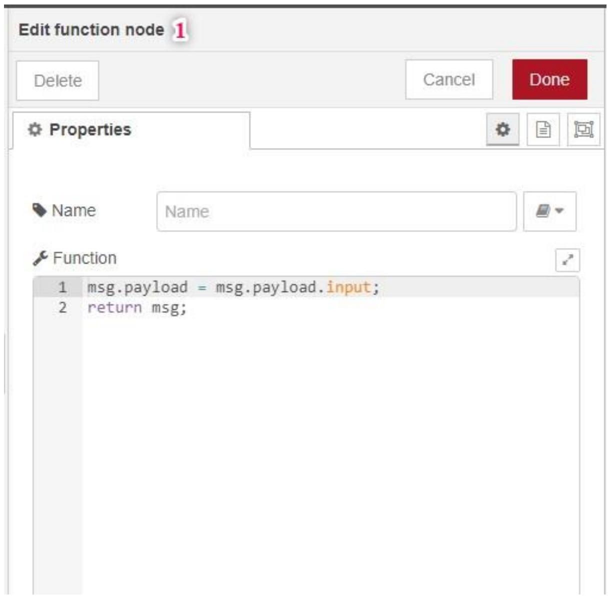
cn-dashboard-nodes
Install
0.0.2 1 year, 9 months ago install

node-red-dashboard
A set of dashboard nodes for Node-RED
2.22.1 1 week ago installed

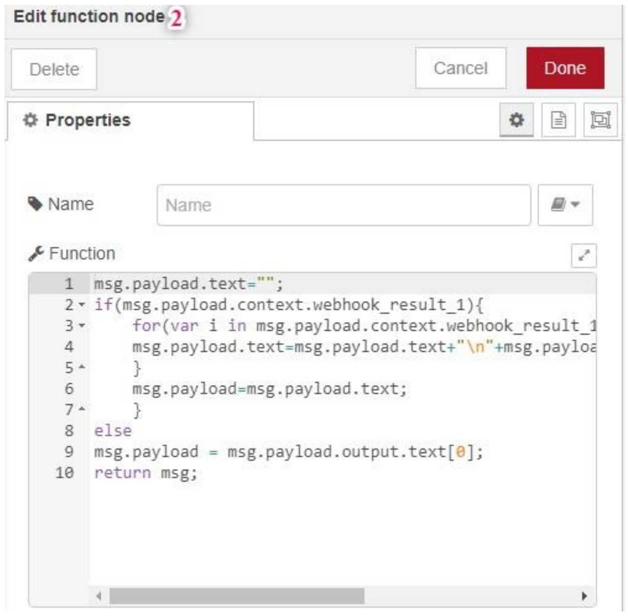
feezal
Web Components based Dashboard UI with WYSIWYG Editor
0.6.2 2 months ago install

node-red-contrib-dashboard-average-bars
Calculate and display the average values of msg.payload in a bar chart.
0.0.6 1 year, 8 months ago install

2. Add **Form** node. Double click on it and add the Group name and Tab name. Also select the size of form.
3. Add **Inject** node. Double click it and name it as Hello.
4. Add two **Text** nodes. Double click the first one and name it as Query and the second one as the Response.
5. Drag two **Debug** nodes.
6. Drag two **Functions** nodes. Add in the codes for both the nodes.

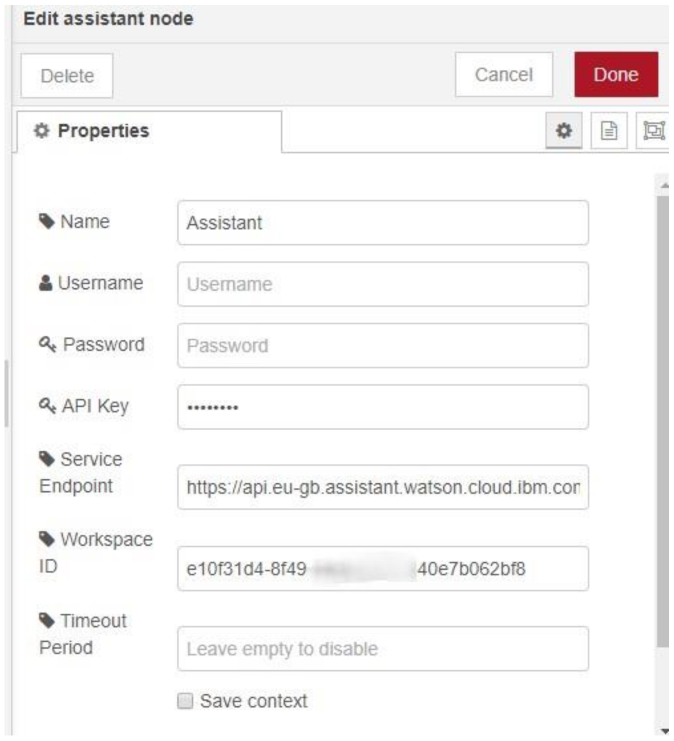


```
msg.payload = msg.payload.input;
return msg;
```

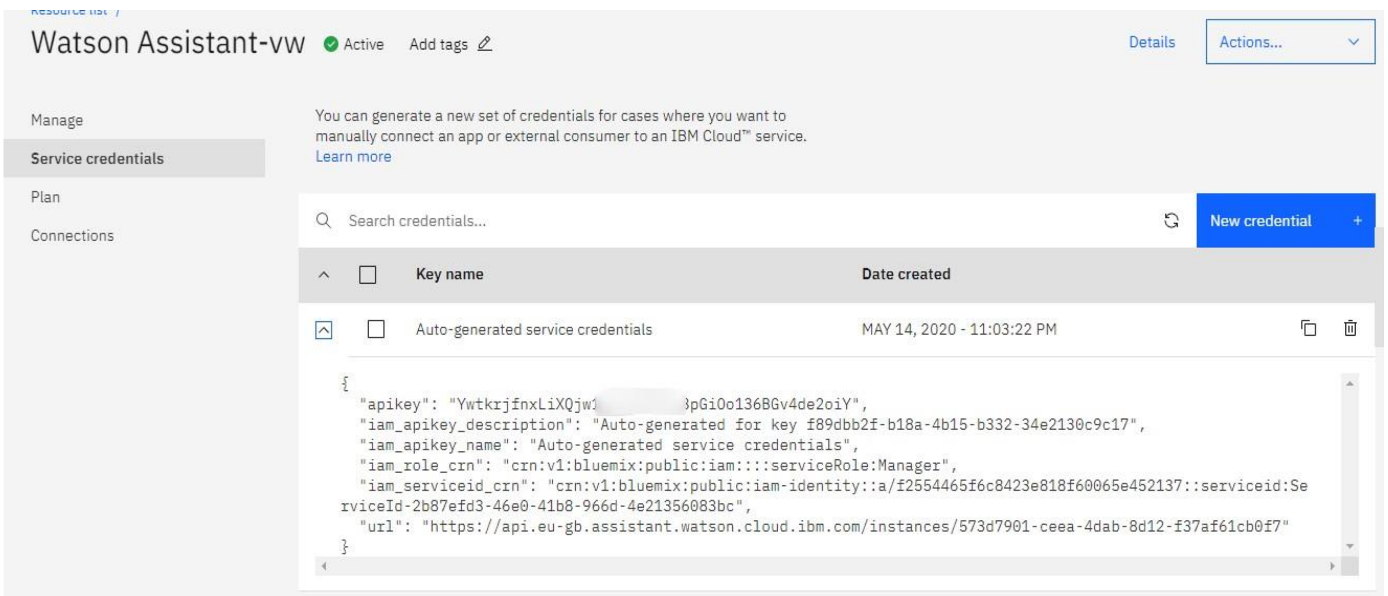


```
msg.payload.text="";
if(msg.payload.context.webhook_result_1){
    for(var i in msg.payload.context.webhook_result_1
    msg.payload.text=msg.payload.text+"\n"+msg.payloa
    }
    msg.payload=msg.payload.text;
    }
else
msg.payload = msg.payload.output.text[0];
return msg;
```

7. Drag **Assistant** node. Double click on it and it will ask credentials like API key, Workspace ID and Service Endpoint which you will get from Watson Assistant service page. Get the details and enter them and click on Done.



Name: Assistant
Username: Username
Password: Password
API Key: ••••••••
Service Endpoint: https://api.eu-gb.assistant.watson.cloud.ibm.con
Workspace ID: e10f31d4-8f49 [redacted] 40e7b062bf8
Timeout Period: Leave empty to disable
☐ Save context


Resource list /

Watson Assistant-vw Active Add tags

Details Actions...

Manage
Service credentials
Plan
Connections

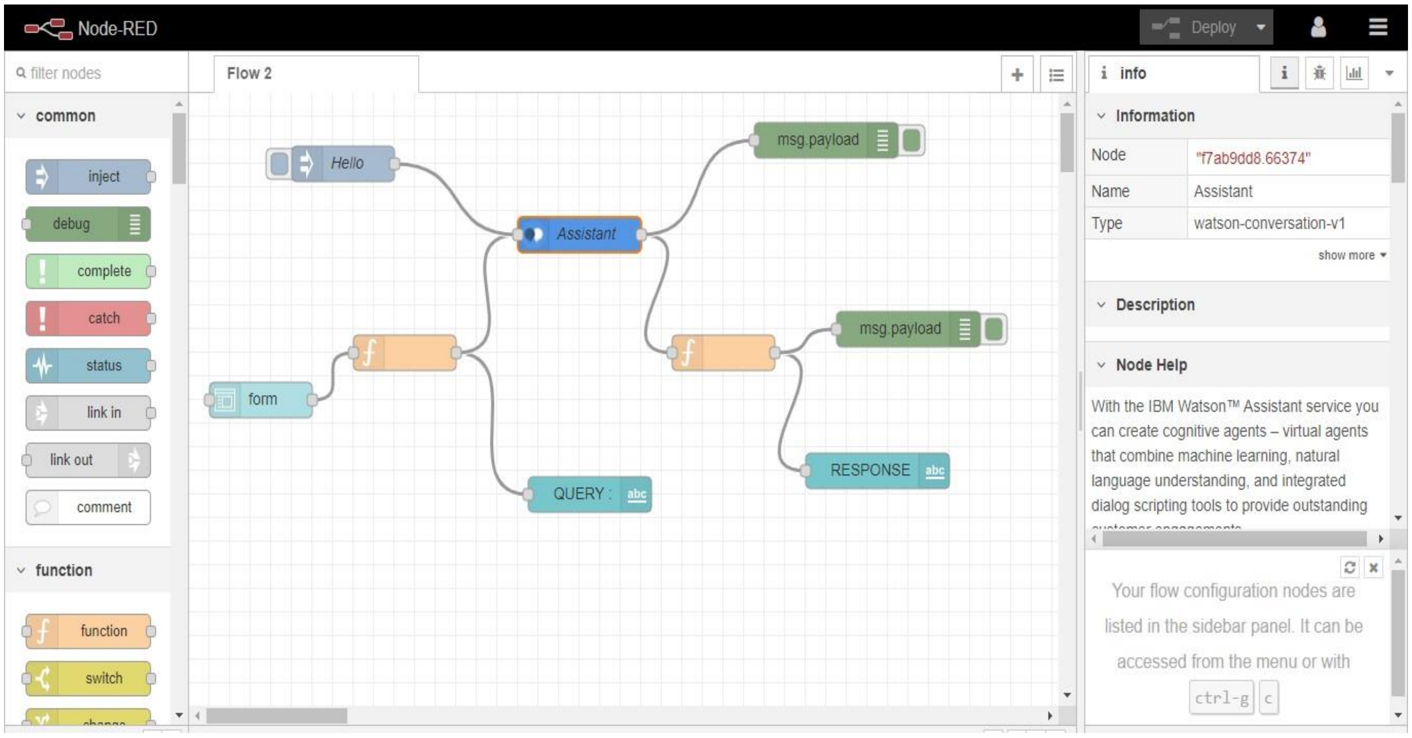
You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud™ service.
Learn more

Search credentials... New credential +

Key name	Date created
Auto-generated service credentials	MAY 14, 2020 - 11:03:22 PM

```
{
  "apikey": "YwtkrjfnxLiXQjw1[redacted]3pGiOo136BGv4de2oiY",
  "iam_apikey_description": "Auto-generated for key f89dbb2f-b18a-4b15-b332-34e2130c9c17",
  "iam_apikey_name": "Auto-generated service credentials",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/f2554465f6c8423e818f60065e452137::serviceid:Se
rviceId-2b87efd3-46e0-41b8-966d-4e21356083bc",
  "url": "https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/573d7901-ceea-4dab-8d12-f37af61cb0f7"
}
```

8. Connect the form node to input of function node 1 and its output to input of Assistant node. Connect output of function node 1 to "Query" text node. Connect the output of Assistant node to input of function node 2 and its output to "Response" text node.
9. Click on Deploy.

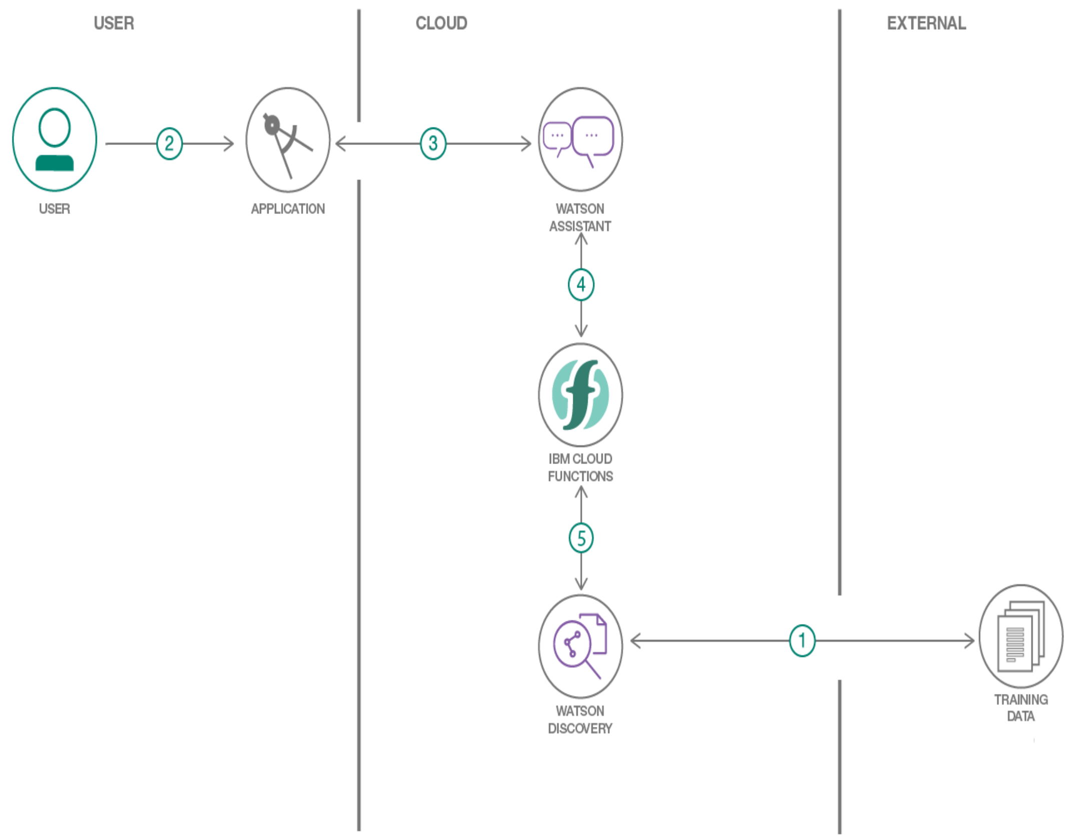


Node	"f7ab9dd8.66374"
Name	Assistant
Type	watson-conversation-v1

With the IBM Watson™ Assistant service you can create cognitive agents – virtual agents that combine machine learning, natural language understanding, and integrated dialog scripting tools to provide outstanding customer engagements

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

FLOWCHART



The flow summarizes the functionality of the Chatbot.

The document preloaded in Discovery is being annotated by Smart Document Understanding feature. This trains the Discovery to improve the query result. The user inetracts with frontend ui of the chatbot and it keeps them engaged in conversation with small talk. This interaction between the user and the UI is being coordinated by Watson Assistant.

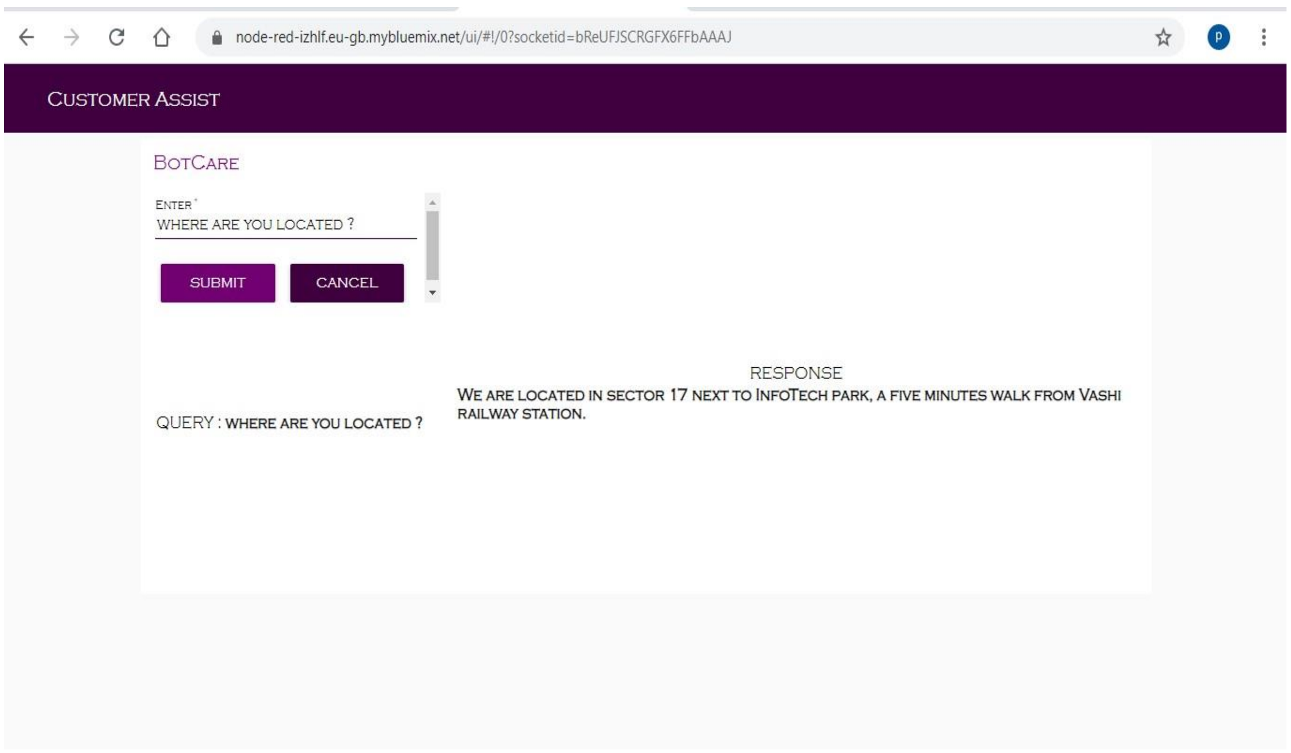
When user asks a technical query, the Assistant invokes the Cloud Function action. This action then queries Watson Discovery and returns with relevant result from the pre-loaded owner's manual.

Watson Assistant, Discovery and Cloud Functions are IBM services. The Training Data is the owner's manual which is external section. The user and applications comes under

User section because the application is accessed by the user.

RESULT

After deploying the flow in Node-Red, open the UI by typing https://node-red-izhlf.eu-gb.mybluemix.net/ui in the browser. You will see the Help Desk created. The layout of the page can also be customized.


node-red-izhlf.eu-gb.mybluemix.net/ui/#!/0?socketid=bReUFJSCRGFX6FFbAAAJ

Customer Assist

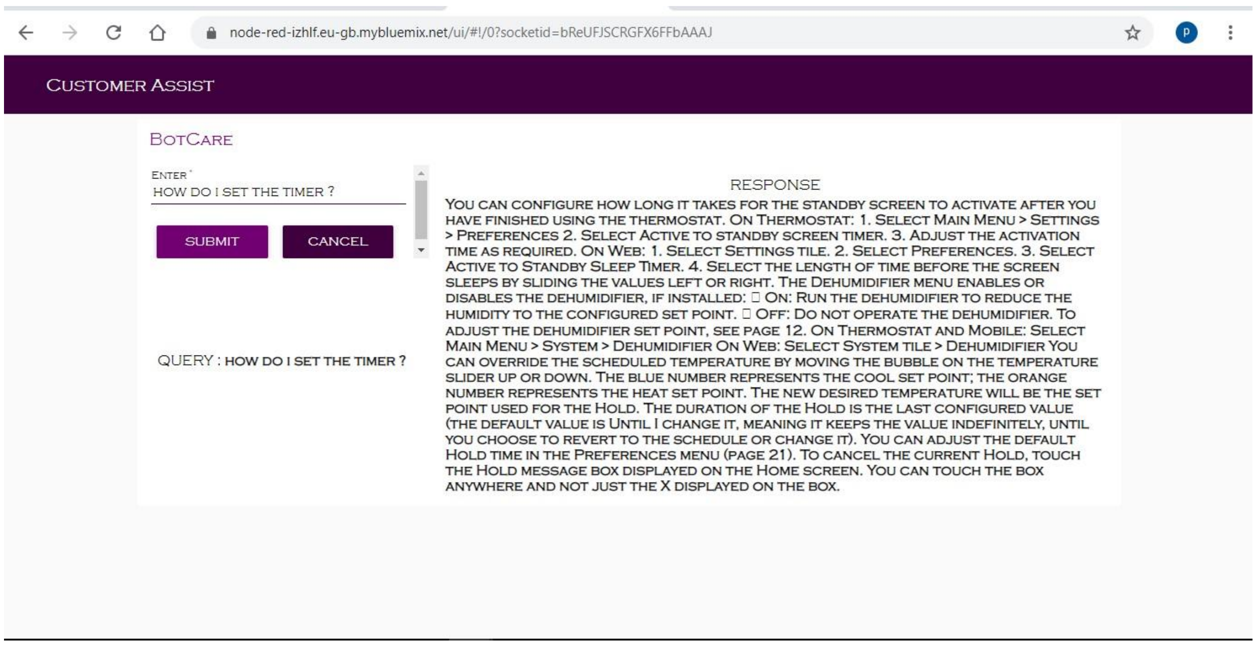
BotCare

Enter *
WHERE ARE YOU LOCATED ?

SUBMIT CANCEL

QUERY : WHERE ARE YOU LOCATED ?

RESPONSE
We are located in sector 17 next to InfoTech park, a five minutes walk from Vashi railway station.


node-red-izhlf.eu-gb.mybluemix.net/ui/#!/0?socketid=bReUFJSCRGFX6FFbAAAJ

Customer Assist

BotCare

Enter *
HOW DO I SET THE TIMER ?

SUBMIT CANCEL

QUERY : HOW DO I SET THE TIMER ?

RESPONSE
You can configure how long it takes for the standby screen to activate after you have finished using the thermostat. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Active to standby screen timer. 3. Adjust the activation time as required. On Web: 1. Select Settings tile. 2. Select Preferences. 3. Select Active to Standby Sleep Timer. 4. Select the length of time before the screen sleeps by sliding the values left or right. The Dehumidifier menu enables or disables the dehumidifier, if installed: On: Run the dehumidifier to reduce the humidity to the configured set point. Off: Do not operate the dehumidifier. To adjust the dehumidifier set point, see page 12. On Thermostat and Mobile: Select Main Menu > System > Dehumidifier On Web: Select System tile > Dehumidifier You can override the scheduled temperature by moving the bubble on the temperature slider up or down. The blue number represents the cool set point; the orange number represents the heat set point. The new desired temperature will be the set point used for the Hold. The duration of the Hold is the last configured value (the default value is Until I change it, meaning it keeps the value indefinitely, until you choose to revert to the schedule or change it). You can adjust the default Hold time in the Preferences menu (page 21). To cancel the current Hold, touch the Hold message box displayed on the Home screen. You can touch the box anywhere and not just the X displayed on the box.

ADVANTAGES / DISADVANTAGES

Advantages

- They are available 24/7 which maintain continous communication between the customer and the seller.

- An operator can concentrate on one customer at a time. A chatbot can however, answer thousands of questions at the same time.
- This reduces the cost of manpower.
- Actions like changing or querying records are almost instantaneous for bots which can significantly improve customer experience.
- Since bots are on digital platforms where people spend majority of their time, bots can be used to automate common tasks such as providing adavanced search functionality.

Disadvantages

- Chatbots are installed in the aim of getting fast response and improving customer interaction. However, due to limited availability of data and the time needed for self updation, the process can be slow and costly.
- For better outcome, more complex designing is required and need of skilled developers.
- At times they can mislead customer if the query doesn't match with the pre loaded dialog. Bots get confused and respond with same answer for different questions. This leads to frustrated customers.
- Unlike humans, each bot needs to be programmed differently for each business, which increases the initial installation cost.

APPLICATIONS

Integration of chatbots with social media platforms like Facebook Messenger, LINE, WeChat, and WhatsApp make it easier for businesses to provide 24/7 customer service to the user.

- For travel industry, such customer service chatbot proved to be a boon. Apart from benefit of booking and scheduling flights, they help to integrate additional services via social media platforms. Services like Airbnb, Uber can be integrated for enhancement of customer experience.
- One of the most useful application of these bots is in healthcare department. An average patient spends 30 minutes trying to get to the right

service in a local hospital. But deploying conversational chatbots in the healthcare sector can significantly reduce long waits. From registration to coverage and claims, chatbots are in popular demand in this industry.

- The use of chatbots in any appliance website is a very common thing today. Earlier, chatbots would tranfer any function related query onto an agent. But with growing technology, that is also eliminated. The bot will the give accurate result without any human interaction.

CONCLUSION

In conclusion, we have created a chatbot which not only engage the customer with small talk but will also provide user with accurate and smart solution. The response will be fast and customer will be satisfied with the service.
By integrating all the above mentioned services, a smarter and efficiently working Customer Help Desk has been developed.

FUTURE SCOPE

While most businesses will have chatbot for customer services, many will develop bots for their internal processing.
One such thing will be Bots handling first stage interviews to find promsing candidates. The bots can process resumes to find highly qualified candidates and fasten the otherwise slow recruiting process and managing cost.
Customer service is increasingly an automated one. Many companies are adopting AI tools to build a better way of handling and expanding the service.

Chatbots have a long way to go to realize their full potential. Still, the chatbots will ultimately generate significant future value in both corporate and customer settings.

BIBLIOGRAPHY

1. https://developer.ibm.com/tutorials/how-to-create-a-node-re d-starter-application/
2. https://cloud.ibm.com/docs/services/discovery?topic=disco very-getting-started
3. https://eu-gb.discovery.watson.cloud.ibm.com/regions/eu-g b/services/crn%3Av1%3Abluemix%3Apublic%3Adiscovery%3 Aeu-gb%3Aa%2Ff2554465f6c8423e818f60065e452137%3A 372afc3e-36b8-46da-9208-5b460b3627a8%3A%3A
4. https://eu-gb.assistant.watson.cloud.ibm.com/eu-gb/crn:v1: bluemix:public:conversation:eu-gb:a~2Ff2554465f6c8423e8 18f60065e452137:573d7901-ceea-4dab-8d12-f37af61cb0f7: :/home
5. https://node-red-izhlf.eu-gb.mybluemix.net/red/#flow/20dda a74.f6e756

APPENDIX

A. ***Source Code***

Code to invoke action :

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

Code for Watson Assistant :

https://github.com/SmartPracticeschool/llSPS-INT-619-Intelligent-Customer-Help-Deskwith-Smart-Document-Understanding/blob/master/WATSON-ASSISTANT/skill-My-first-s kill.json

NODE-RED FLOW:

```
[
        {
            "id": "20ddaa74.f6e756",
            "type": "tab",
            "label": "Flow 2",
            "disabled": false,
            "info": ""
        },
        {
            "id": "88effef0.1d155",
            "type": "ui_base",
            "theme": {
                "name": "theme-light",
                "lightTheme": {
                    "default": "#0094CE",
                    "baseColor": "#400040",
                    "baseFont": "Copperplate,Copperplate Gothic Light,fantasy",
                    "edited": true,
                    "reset": false
                },
                "darkTheme": {
                    "default": "#097479",
                    "baseColor": "#097479",
                    "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                    "edited": false
                },
                "customTheme": {
                    "name": "Untitled Theme 1",
                    "default": "#4B7930",
                    "baseColor": "#4B7930",
                    "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                    "reset": false
                },
                "themeState": {
                    "base-color": {
                        "default": "#0094CE",
                        "value": "#0094CE",
                        "edited": true
                    },
                    "page-titlebar-backgroundColor": {
                        "value": "#400040",
                        "edited": false
            },
            "page-backgroundColor": {
                "value": "#fafafa",
                "edited": false
            },
            "page-sidebar-backgroundColor": {
                "value": "#000000",
                "edited": false
            },
            "group-textColor": {
                "value": "#8c008c",
                "edited": false
            },
            "group-borderColor": {
                "value": "#ffffff",
                "edited": false
            },
            "group-backgroundColor": {
                "value": "#ffffff",
                "edited": false
            },
            "widget-textColor": {
                "value": "#111111",
                "edited": true
            },
            "widget-backgroundColor": {
                "value": "#400040",
                "edited": false
            },
            "widget-borderColor": {
                "value": "#ffffff",
                "edited": false
            },
            "base-font": {
                "value": "Copperplate,Copperplate Gothic Light,fantasy"
            }
        },
        "angularTheme": {
            "primary": "indigo",
            "accents": "blue",
            "warn": "red",
            "background": "grey"
        }
    },
    "site": {
            "name": "Node-RED Dashboard",
            "hideToolbar": "false",
            "allowSwipe": "false",
            "lockMenu": "false",
            "allowTempTheme": "true",
            "dateFormat": "DD/MM/YYYY",
            "sizes": {
                "sx": 48,
                "sy": 48,
                "gx": 6,
                "gy": 6,
                "cx": 6,
                "cy": 6,
                "px": 0,
                "py": 0
            }
        }
    },
    {
        "id": "3052ee13.5d4892",
        "type": "ui_tab",
        "z": "",
        "name": "Customer Assist",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    },
    {
        "id": "f485bfba.2c8e9",
        "type": "ui_group",
        "z": "",
        "name": "BotCare",
        "tab": "3052ee13.5d4892",
        "order": 1,
        "disp": true,
        "width": 20,
        "collapse": false
    },
    {
        "id": "ca61f16c.74c6c",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 3,
        "width": 1,
        "height": 1
    },
    {
        "id": "7a4fd691.3bd1b8",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 4,
        "width": 1,
        "height": 1
    },
    {
        "id": "cdeabcfb.bb9b5",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 6,
        "width": 1,
        "height": 1
    },
    {
        "id": "7ae1761.0e9fb88",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 7,
        "width": 1,
        "height": 1
    },
    {
        "id": "101dc7b5.a1f448",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 8,
        "width": 6,
        "height": 1
    },
    {
        "id": "32bf8e4a.e5eaf2",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 9,
        "width": 1,
        "height": 1
    },
    {
        "id": "d54d9df0.0a3cf",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 10,
        "width": 6,
        "height": 1
    },
    {
        "id": "3e8399b3.a57016",
        "type": "ui_spacer",
        "name": "spacer",
        "group": "f485bfba.2c8e9",
        "order": 11,
        "width": 1,
        "height": 1
    },
    {
        "id": "b2d43cd2.d98b7",
        "type": "function",
        "z": "20ddaa74.f6e756",
        "name": "",
        "func": "msg.payload = msg.payload.input;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 210,
        "y": 220,
        "wires": [
            [
                "f79c73f.62bda9",
                "f7ab9dd8.66374"
            ]
        ]
    },
    {
        "id": "eec436f9.ad0ea8",
        "type": "debug",
        "z": "20ddaa74.f6e756",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "x": 700,
        "y": 200,
        "wires": []
    },
    {
        "id": "db23a440.ffec28",
        "type": "function",
        "z": "20ddaa74.f6e756",
        "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    msg.payload=msg.payload.text;\n    }\nelse\nmsg.payload =
msg.payload.output.text[0];\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 520,
        "y": 220,
        "wires": [
            [
                "f437990b.a15178",
                "eec436f9.ad0ea8"
            ]
        ]
    },
    {
        "id": "53fde0fa.7f963",
        "type": "debug",
        "z": "20ddaa74.f6e756",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "x": 620,
        "y": 40,
        "wires": []
    },
    {
        "id": "f7ab9dd8.66374",
        "type": "watson-conversation-v1",
        "z": "20ddaa74.f6e756",
        "name": "Assistant",
        "workspaceid": "e10f31d4-8f49-44cb-8021-540e7b062bf8",
        "multiuser": false,
        "context": false,
        "empty-payload": false,
        "service-endpoint": "https://api.eu-
gb.assistant.watson.cloud.ibm.com/instances/573d7901-ceea-4dab-8d12-f37af61cb0f7",
        "timeout": "",
        "optout-learning": false,
        "x": 380,
        "y": 120,
        "wires": [
            [
                "db23a440.ffec28",
                "53fde0fa.7f963"
            ]
        ]
    },
    {
        "id": "7e84a245.aa880c",
        "type": "ui_form",
        "z": "20ddaa74.f6e756",
        "name": "",
        "label": "",
        "group": "f485bfba.2c8e9",
        "order": 1,
        "width": 6,
        "height": 2,
        "options": [
            {
                "label": "Enter",
                "value": "input",
                "type": "text",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "input": ""
        },
        "payload": "",
        "submit": "submit",
        "cancel": "cancel",
        "topic": "",
        "x": 70,
        "y": 260,
        "wires": [
            [
                "b2d43cd2.d98b7"
            ]
        ]
    },
    {
        "id": "f79c73f.62bda9",
        "type": "ui_text",
        "z": "20ddaa74.f6e756",
        "group": "f485bfba.2c8e9",
        "order": 5,
        "width": 6,
        "height": 2,
        "name": "",
        "label": "QUERY :",
        "format": "{{msg.payload}}",
        "layout": "row-center",
        "x": 390,
        "y": 340,
        "wires": []
    },
    {
        "id": "f437990b.a15178",
        "type": "ui_text",
        "z": "20ddaa74.f6e756",
        "group": "f485bfba.2c8e9",
        "order": 2,
        "width": "14",
        "height": "7",
        "name": "",
        "label": "RESPONSE",
        "format": "{{msg.payload}}",
        "layout": "col-center",
        "x": 670,
        "y": 320,
        "wires": []
    },
    {
        "id": "57e5c959.3b6718",
        "type": "inject",
        "z": "20ddaa74.f6e756",
        "name": "Hello",
        "topic": "",
        "payload": "Hello",
        "payloadType": "str",
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "x": 150,
        "y": 60,
        "wires": [
            [
                "f7ab9dd8.66374"
            ]
        ]
    }
]
```