

# **Project Report**

**Name : PRASHANT SINGH VERMA**

**Email: prashant.verma54@gmail.com**

**Title : Intelligent Customer Help Desk With Smart Document Understanding**

**Category: Artificial Intelligence**

**Internship at [smartinternz.com](https://smartinternz.com)@2020**

<b>1</b>	<b>INTRODUCTION</b>
	<b>1.1 Overview</b>
	<b>1.2 Purpose</b>
<b>2</b>	<b>LITERATURE SURVEY</b>
	<b>2.1 Existing problem</b>
	<b>2.2 Proposed solution</b>
<b>3</b>	<b>THEORITICAL ANALYSIS</b>
	<b>3.1 Block diagram</b>
	<b>3.2 Hardware / Software designing</b>
<b>4</b>	<b>EXPERIMENTAL INVESTIGATIONS</b>
<b>5</b>	<b>FLOWCHART</b>
<b>6</b>	<b>RESULT</b>
<b>7</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>
<b>8</b>	<b>APPLICATIONS</b>
<b>9</b>	<b>CONCLUSION</b>
<b>10</b>	<b>FUTURE SCOPE</b>
<b>11</b>	<b>BIBILOGRAPHY</b>
	<b>APPENDIX</b>
	<b>A. Reference</b>

## **1.INTRODUCTION**

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

**Project Requirements:** Python, IBM Cloud, IBM Watson

**Functional Requirements:** IBM cloud

**Technical Requirements:** AI,ML,WATSON AI,PYTHON

**Software Requirements:** Watson assistant, Watson discovery.

**Project Deliverables:** Smartinternz Internship

**Project Team:** Prashant Singh Verma

**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 with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners manual to help solve our customers' problems.

To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owners manual is important and what is not. This will improve the answers returned from the queries.

#### **1.2.1 Scope of Work**

**Create a customer care dialog skill in Watson Assistant**

**Use Smart Document Understanding to build an enhanced Watson  
Discovery collection**

**Create an IBM Cloud Functions web action that allows Watson Assistant to  
post queries to Watson Discovery**

**Build a web application with integration to all these services & deploy the  
same on IBM Cloud Platform**

## **2.LITERATURE SURVEY**

### **2.1 Existing problem:**

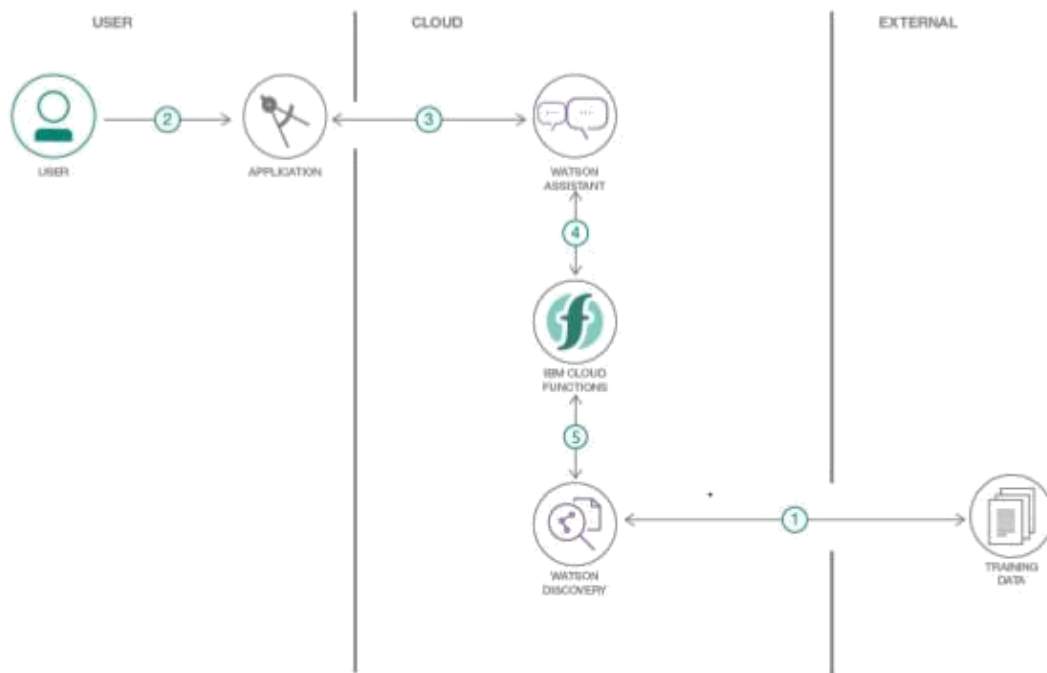
Generally Chatbots means getting input from users and getting only response questions and for some questions the output from bot will be like “try again”, “I don’t understand”, “will you repeat again”, and so on... and directs customer to customer agent but a good customer Chatbot should minimize involvement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

### **2.2 Proposed solution:**

For the above problem to get solved we have to put an virtual agent in chatbot so it can understand the queries that are posted by customers. The virtual agent should trained from some insight records based company background so it can answer queries based on the product or related to company. In this project I used Watson Discovery to achieve the above solution. And later including Assistant and Discovery on Node-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 app UI is a chatbot that engages the user in a conversation.
3. Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
4. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
5. The Cloud Functions action will query the Watson Discovery service and return the results.

### 3.2 Hardware / Software designing:

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

## **4.EXPERIMENTAL INVESTIGATIONS**

### **1.Create IBM Cloud services**

**Create the following services:**

**Watson Discovery**

**Watson Assistant**

**Node Red**

### **2. Configure Watson Discovery**

**Import the document**

**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 electrical\_power\_transmission\_systems.pdf file located in the data directory of your local repo.**

**The electrical power transmission lines are essential parameters in transmission of power from one place to another.**

**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.**

The screenshot shows the IBM Watson Discovery interface. The top navigation bar includes 'Overview', 'Errors and warnings (1)', and 'Search settings'. The main content area displays '1 document' and '0 documents failed'. It lists 'Identified 1 field from your data' as 'text'. Under 'Added 4 enrichments to your data', it shows 'Entity Extraction' with results like '10\$', '100%', '160 km', and '250 Kms'; 'Sentiment Analysis' with '100% positive', '0% neutral', and '0% negative'; 'Concept Tagging' with 'Direct current', 'Electric power transmission', and 'Electrical engineering'; and 'Category Classification' with 'business and industr... energy'. A 'Build your own query' button is visible at the bottom right of the main content area.

Click the Build your own query [1] button.

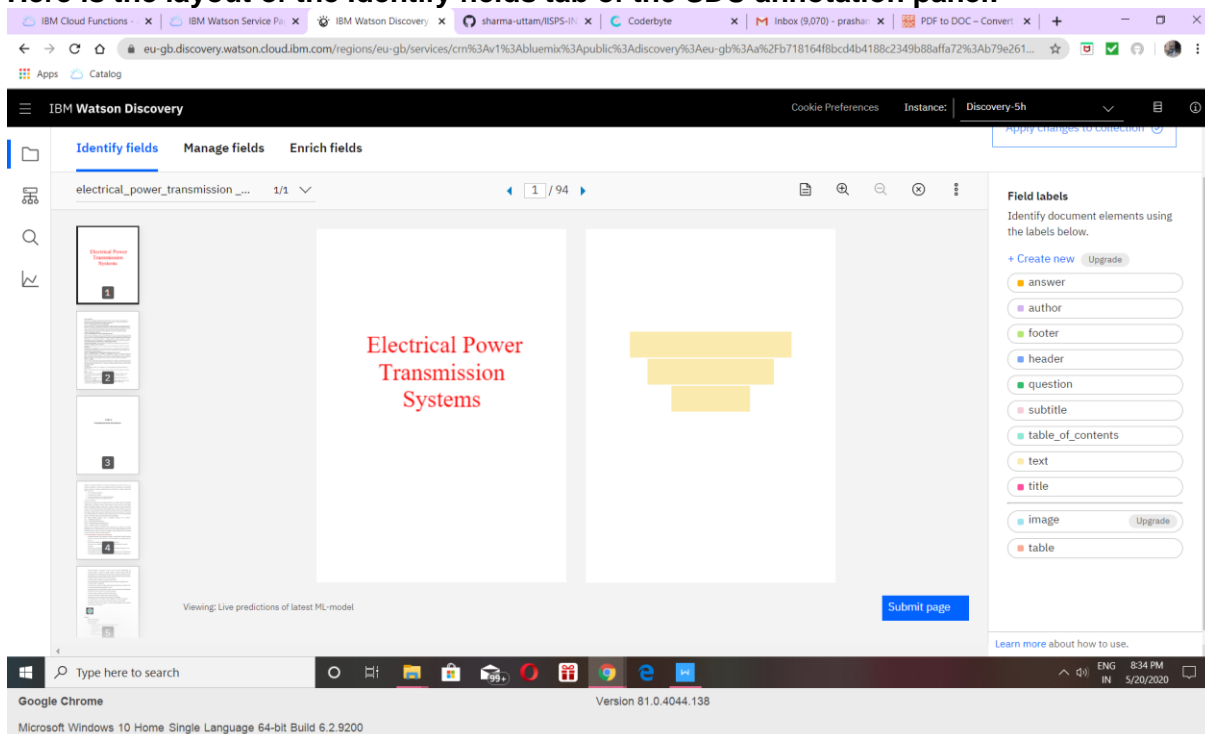
The screenshot shows the 'Build queries' page in IBM Watson Discovery. The left sidebar has a search bar with the query 'what are different types of conductors'. The main content area shows a 'Search for documents' section with options to 'Use natural language' or 'Use the Discovery Query Language'. Below this, there are checkboxes for 'Include analysis of your results' and 'Filter which documents you query'. A 'Run query' button is at the bottom right. The right sidebar shows a 'Summary' section with a 'JSON' tab, a 'Query URL', and a 'Passages' section containing text about ionization and transmission lines. A 'Results' section at the bottom indicates 'Showing 1 of 1 matching documents'.

Enter queries related to the performance of transmission lines and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question.

## Annotate with SDU

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

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



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

[1] is the list of pages in the manual. As each is processed, a green check mark will appear on the page.

[2] is the current page being annotated.



**[3] is where you select text and assign it a label.**

**[4] is the list of labels you can assign to the page**

**text. Click [5] to submit the page to Discovery.**

**Click [6] when you have completed the annotation process.**

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

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

**The main title page as title**

**The table of contents (shown in the first few pages) as table\_of\_contents**

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

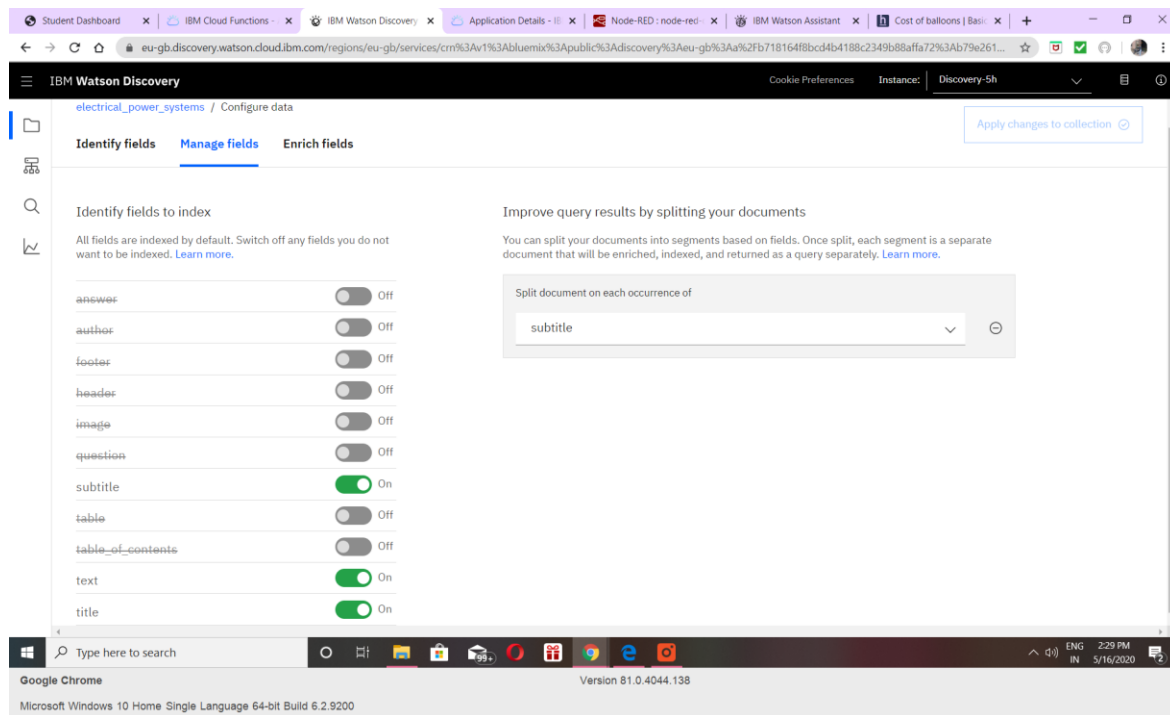
**All page numbers as footers**

**All warranty and licensing information (located in the last few pages) as a footer**

**All other text should be marked as text.**

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

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



[2] Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except subtitles and text.

[3] is telling Discovery to split the document apart, based on subtitle.

Click [4] to submit your changes.

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

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

Student Dashb...IBM Cloud Fun...IBM Watson Di...Application De...Node-RED : no...IBM Watson A...electrical\_pow...Download Win...Cost of balloo...+

eu-gb.discovery.watson.cloud.ibm.com/regions/eu-gb/services/cm%3Av1%3Abluemix%3Apublic%3Adiscovery%3Aeu-gb%3Aa%2Fb718164f8bcd4b4188c2349b88affa72%3Ab79e261...

IBM Watson DiscoveryCookie PreferencesInstance: Discovery-5h

electrical\_power\_systemsConfigure data

OverviewErrors and warnings (1)Search settings

IBM Cloud FunctionsIBM Watson Service P...IBM Watson Discoverysharma-uttam/ISPS-ib...CoderbyteInbox (9,071) - prash...PDF to DOC - Convers...+

eu-gb.discovery.watson.cloud.ibm.com/regions/eu-gb/services/cm%3Av1%3Abluemix%3Apublic%3Adiscovery%3Aeu-gb%3Aa%2Fb718164f8bcd4b4188c2349b88affa72%3Ab79e261...

AppsCatalog

IBM Watson DiscoveryCookie PreferencesInstance: Discovery-5h

electrical\_power\_systems / Build queriesTrain Watson to improve results

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](#)

what are different types of conductors

+ Include analysis of your results

+ Filter which documents you query

Run queryClose

SummaryJSON

Query URL: <https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/b...>

Passages

"Ionization produces the undesirable effect called corona. The symbols identifying different types of Aluminium conductors are as follows:-AAC: AllAluminiumconductors. AAAC: AllAluminiumAlloyconductors ACSR"

"It deals with basic theory of transmission lines modeling and their performance analysis. Also this course gives emphasis on mechanical design of transmission lines, cables and insulators. UNIT I : TRANSMISSION LINE PARAMETERS Types of Conductors – ACSR, Bundled and Standard"

"Sag and Tension Calculations: Sag and Tension Calculations with Equal and Unequal Heights of Towers, Effect of Wind and Ice on Weight of Conductor, Stringing Chart and Sag Template and Its Applications, Numerical Problems. UNIT IV :POWER SYSTEM TRANSIENTS & TRAVELLING WAVES Types of System Transients - Travelling or Propagation of Surges"

Results

Showing 1 of 1 matching documents

Type here to search

Google Chrome

Version 81.0.4044.138

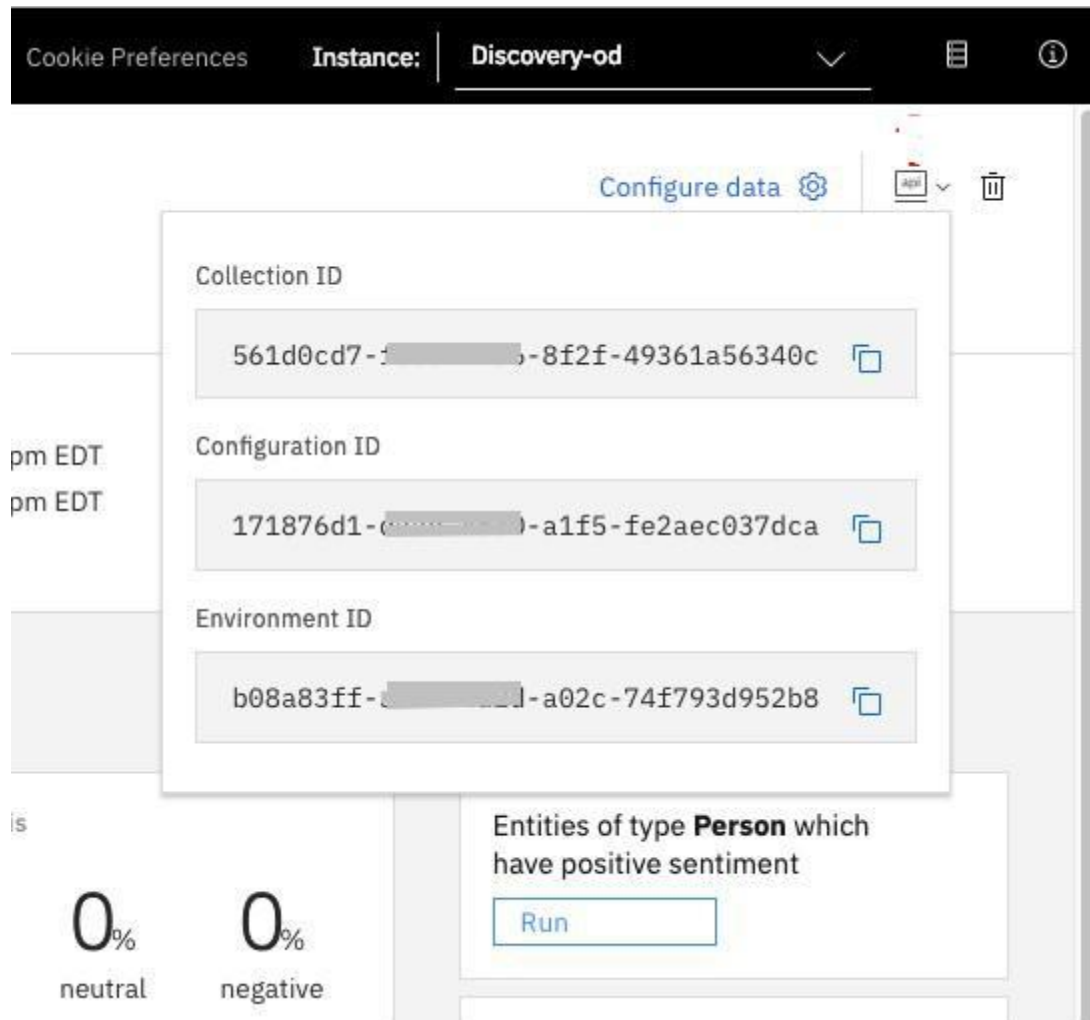
Microsoft Windows 10 Home Single Language 64-bit Build 6.2.9200

he

## Store credentials for future use

In upcoming steps, you will need to provide the credentials to access your Discovery collection. The values can be found in the following locations.

The Collection ID and Environment ID values can be found by clicking the dropdown button [1] located at the top right side of your collection panel:



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

IBM Cloud

Search resources and offerings...

Resource list /

Discovery-od

Resource group: default Location: Dallas Add Tags

Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service. [Learn more](#)

Service credentials

New credential

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

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	FEB 5, 2019 - 09:26:31 AM	View credentials 2

```

3 {
  "apikey": "dry8f3aITnsy/ [REDACTED] #h1au8bko4fu10",
  "iam_apikey_description": "Auto generated apikey during resource-key operation for Instance - crn:v1:bluemix:public:discovery:us-south:a/bc1bd51c396536dc7d5f81d5a4e19533:acf2e2871-3b0d-4e04-a0f9-0daa59779852::",
  "iam_apikey_name": "auto-generated-apikey-45f36cdd-did2-4a17-b4id-8ca5d1f1c7a6",
  "iam_role_crn": "crn:v1:bluemix:public:iam:::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity:a/bc1bd51c396536dc7d5f81d5a4e19533::serviceid:ServiceId-01608efa-a050-4708-a191-0b71f43c0ddb",
4 "url": "https://gateway.watsonplatform.net/discovery/api"
}

```

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

### 3.Create IBM Cloud Functions action

Now let's create the web action that will make queries against our Discovery collection.

Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter [1], then select the Functions card [2]:

IBM Cloud

Search resources and offerings...

Catalog Docs Support Manage Richard Hagarty's Account

# Catalog

functions 1

All Categories (2) >

- VPC Infrastructure
- Compute (1)
- Containers
- Networking

## Compute

### Serverless Compute

Functions IBM • IAM-enabled 2

IBM Cloud Functions is a Function-as-a-Service (FaaS)

cloud.ibm.com/functions/create/action

IBM Cloud Functions / Create / Action

## Create Action

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

[Learn more about Actions](#)

[Learn more about Packages](#)

Action Name

cool\_monkey

An action by this name already exists.

Enclosing Package

(Default Package)

Create Package

Runtime

Node.js 10

Looking for Java, .NET or Docker? [Docker](#) Actions can be created with the [CLI](#)

Previous Cancel Create

Microsoft Windows 10 Home Single Language 64-bit Build 6.2.9200

Google Chrome Version 81.0.4044.138

ENG 8:49 PM 5/20/2020

Package  
function.

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

The screenshot shows the IBM Cloud Functions web interface. The browser tabs include 'IBM Cloud Functions', 'IBM Watson Service', 'IBM Watson Discovery', 'sharma-uttam/ISSP-IT', 'Coderbyte', 'Inbox (9,071) - prashant', and 'PDF to DOC - Convert'. The address bar shows the URL 'cloud.ibm.com/functions/details/action/prashant.verma54@gmail.com\_dev/cool\_monkey/code'. The page header includes the IBM Cloud logo, a search bar, and navigation links like 'Catalog', 'Docs', 'Support', 'Manage', and 'Prashant Verma's Account'. The main content area is titled 'cool\_monkey' and 'Web Action'. On the left, there's a sidebar with tabs for 'Code', 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' tab is active, showing a code editor with the following JavaScript code:

```
1 // **
2 //
3 // * main() will be run when you invoke this action
4 //
5 // * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6 //
7 // * @return The output of this action, which must be a JSON object.
8 //
9 //
10 const assert = require('assert');
11 const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');
12
13 function main(params) {
14   return new Promise(function (resolve, reject) {
15     let discovery;
16
17     if (params.iam_apikey) {
18       discovery = new DiscoveryV1({
19         'iam_apikey': params.iam_apikey,
20         'url': params.url,
21         'version': '2019-03-25'
22       });
23     } else {
24       discovery = new DiscoveryV1({
25         'username': params.username,
26         'password': params.password,
27         'url': params.url,
28         'version': '2019-03-25'
29       });
30     }
31
32     discovery.query({
33       'environment_id': params.environment_id,
34       'collection_id': params.collection_id,
35       'natural_language_query': params.input,
36       'passages': true,
37       'count': 3,
38     });
39   });
40 }
```

The code editor has a status bar at the bottom indicating 'Node.js 10' and 'Edit mode - press ESC to exit'. There are buttons for 'Invoke with parameters' and 'Invoke'. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray shows 'ENG IN', '8:51 PM', and '5/20/2020'.

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

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

## Select the Parameters tab [1]

The screenshot shows the IBM Cloud Functions console. The breadcrumb navigation is 'Functions / Actions / cool\_monkey'. The main heading is 'cool\_monkey' with a 'Web Action' icon. The namespace is 'prashant.verma54@gmail.com\_dev(London)'. On the left sidebar, the 'Parameters' tab is selected. The main area displays a table of parameters:

Parameter Name	Parameter Value
url	<a href="https://api.us-south.functions.cloud.ibm.com/functions/b0b244d-029d-4a2a-b444-62d4">https://api.us-south.functions.cloud.ibm.com/functions/b0b244d-029d-4a2a-b444-62d4</a>
environment_id	[REDACTED]
collection_id	[REDACTED]
iam_apikey	"YNY_uOMaVG_Hmyl3VdoSiWm68LtqesAFHhXTOeyk7nc"

At the bottom of the page, the Windows taskbar shows Google Chrome, Version 81.0.4044.138, and the system clock indicates 8:52 PM on 5/20/2020.

Add the following keys:

**url**

**environment\_id**

**collection\_id**

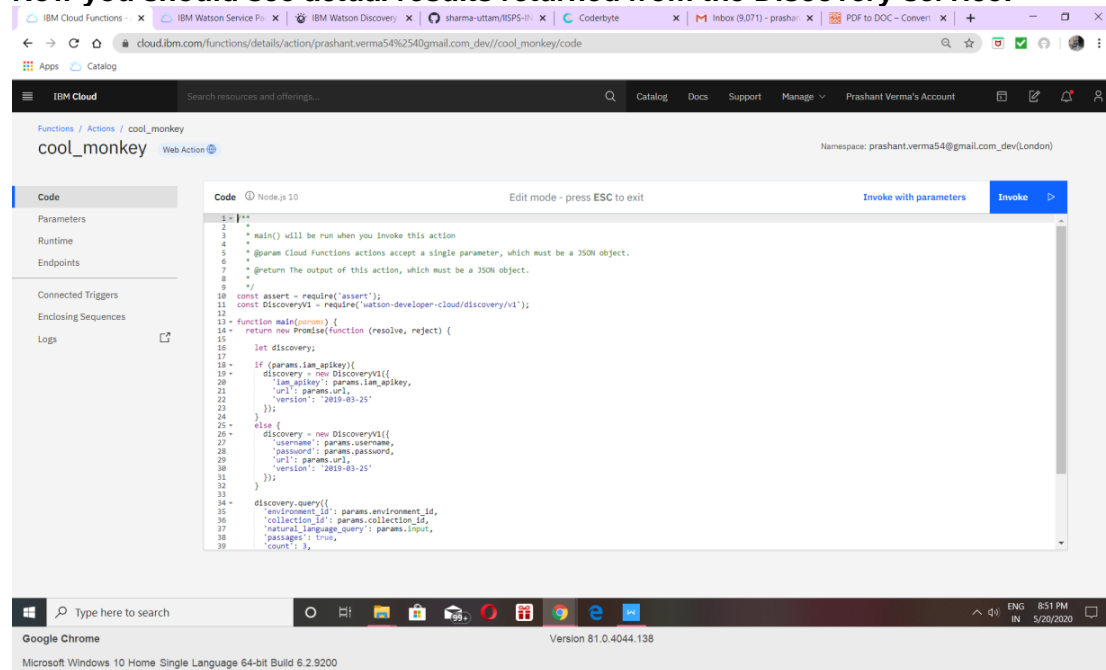
**iam\_apikey**



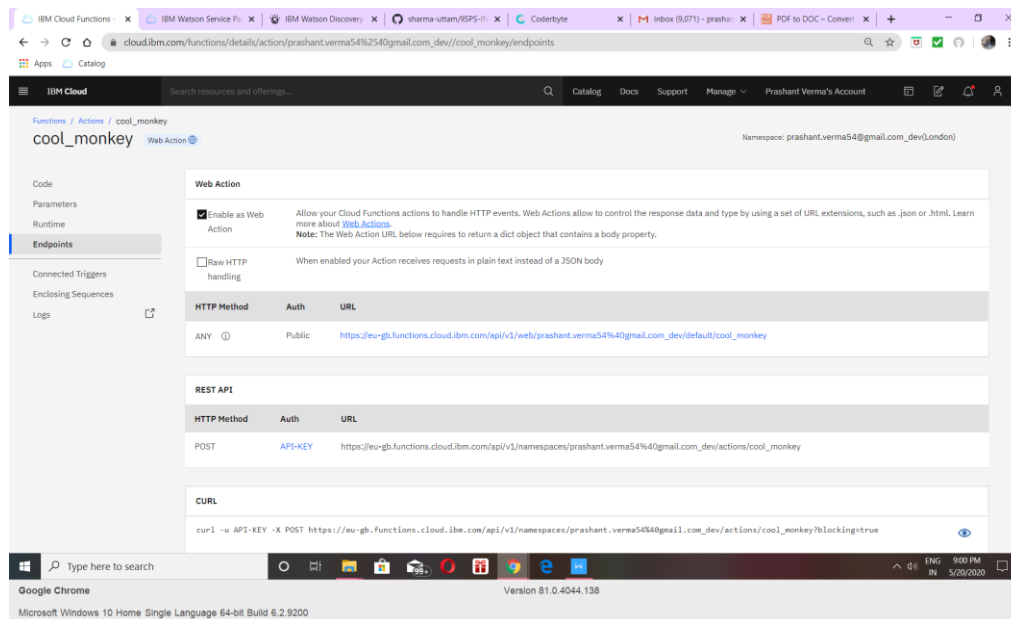
For values, please use the values associated with the Discovery service you created in the previous step.

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:



Next, go to the Endpoints panel [1]:



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

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

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

#### 4. Configure Watson Assistant

Launch the Watson Assistant tool and create a new dialog skill. Select the Use sample skill option as your starting point. This dialog skill contains all of the nodes needed to have a typical call center conversation with a user.

##### Add new intent

The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this.

Create a new intent that can detect when the user is asking about the types of conductors. From the Customer Care Sample Skill panel, select the Intents tab.

Click the Create intent button.

Name the intent **#Performance\_Of\_Transmission\_Lines**, and at a minimum, enter the following example questions to be associated with it.

The screenshot shows the IBM Watson Assistant Lite interface in a web browser. The browser's address bar displays the URL: `eu-gb.assistant.watson.cloud.ibm.com/eu-gb/cmv1/bluemixpublicconversationeu-gba~2fb718164f8bcd4b4188c2349b88affa728a48a701-a375-4937-890c-9164d3c46607/-/skills/41...`. The page title is "IBM Watson Assistant Lite" with an "Upgrade" link. The breadcrumb navigation shows the path: `< | #Performance_Of_Transmission_Lines`. The "Last updated" status is "3 days ago".

The main form for creating an intent is visible. The "Intent name" field contains `#Performance_Of_Transmission_Lines`. The "Description (optional)" field contains `User wants to know about transmission lines`. The "User example" section has a text input field with the placeholder `Type a user example here, e.g. I want to pay my credit card bill`. Below this are buttons for `Add example` and `Show recommendations`.

Below the form, there is a section for "User examples (6)". It includes a toggle for `Annotate entities` and a link for `What's this?`. The list of examples shows:

- ☐ User examples (6) ↑ Added 11
- ☐ How to calculate efficiency of transmission lines 4 days ago

The bottom of the interface shows a Windows taskbar with the search bar, task view, and several application icons. The system tray indicates the language is `ENG IN`, the time is `9:02 PM`, and the date is `5/20/2020`. The footer text reads: `Google Chrome Version 81.0.4044.138 Microsoft Windows 10 Home Single Language 64-bit Build 6.2.9200`.

## 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.

The screenshot shows the IBM Watson Assistant Skills interface. The top navigation bar includes tabs for Intents, Entities, **Dialog** (marked with a red '1'), Analytics, Options, Versions, and Content Catalog. The main content area displays a list of dialog nodes:

- Directions and location**  
#Customer\_Care\_Store\_Location  
3 Responses / 0 Context Set / Skip user input / Returns
- Make an appointment**  
#Customer\_Care\_Appointments  
3 Responses / 7 Context Set / 5 Slots / Does not return
- Transfer to agent**  
#General\_Connect\_to\_Agent  
1 Responses / 0 Context Set / Does not return
- Small Talk** (marked with a red '2')  
3 Dialog nodes / No digressions
- anything\_else**  
1 Responses / 0 Context Set / Returns

A context menu is open for the 'Small Talk' node, showing the following options:

- Add node to folder
- Add node above
- Add node below** (marked with a red '3')
- Add folder
- Move
- Duplicate
- Jump to
- Delete

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

The screenshot displays the IBM Watson Assistant web interface. On the left, a sidebar menu includes 'Intents', 'Entities', 'Dialog', 'Options', 'Analytics', 'Versions', and 'Content Catalog'. The 'Dialog' section is active, showing a list of nodes for the 'Customer Care Sample Skill'. The node 'Ask about transmission lines' is highlighted with a blue border. To the right, the configuration panel for this node is shown. It includes a 'Node name' field with the text 'Ask about transmission lines', a 'Disambiguation' section with a 'Show node name' toggle set to 'On', and an 'If assistant recognizes' section with a parameter '#Performance\_Of\_Transmission\_Lines'. Below this, the 'Then callout to my webhook' section is visible, followed by a 'Parameters' table with columns 'Key' and 'Value'. The table contains one entry: 'input' with the value '<?input.text?>'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 9:03 PM on 5/20/2020.

IBM Watson Assistant

Customer Care Sample Skill

Intents

Entities

Dialog

Options

Analytics

Versions

Content Catalog

3 Responses / 0 Context Set / Skip user input / Does not return

I want to make an appointment

#Customer\_Care\_Appointments

3 Responses / 7 Context Set / 5 Slots / Does not return

Please transfer me to an agent

#General\_Connect\_to\_Agent

1 Responses / 0 Context Set / Does not return

Small Talk

3 Dialog nodes / Does not return

Ask about transmission lines

#Performance\_Of\_Transmission\_Lines

2 Responses / 0 Context Set / Does not return

anything\_else

1 Responses / 0 Context Set / Returns

#Thanks

Ask about transmission lines

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

Disambiguation

Choose whether to include this node in the list of options shown to customers when the assistant asks customer to clarify their meaning. [Learn more](#)

Show node name

On

[Hide settings](#)

If assistant recognizes

#Performance\_Of\_Transmission\_Lines

Then callout to my webhook [Learn more](#)

Parameters

Key	Value
input	<?input.text?>

Version 81.0.4044.138

Microsoft Windows 10 Home Single Language 64-bit Build 6.2.9200

This means that if Watson Assistant recognizes a user input such as "what are the different types of conductors?", it will direct the conversation to this node.

#### **Enable webhook from Assistant**

Set up access to our WebHook for the IBM Cloud Functions action you created in Step #4.

Select the Options tab [1]:

IBM Watson Assistant

Skills /

Customer Care Sample Skill for Disco

Sample simple customer service skill to get you started.

Options

Webhooks

Autocorrection

System Entities

Webhooks

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

[Learn more](#)

URL

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

Headers

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

HEADER NAME	HEADER VALUE
-------------	--------------

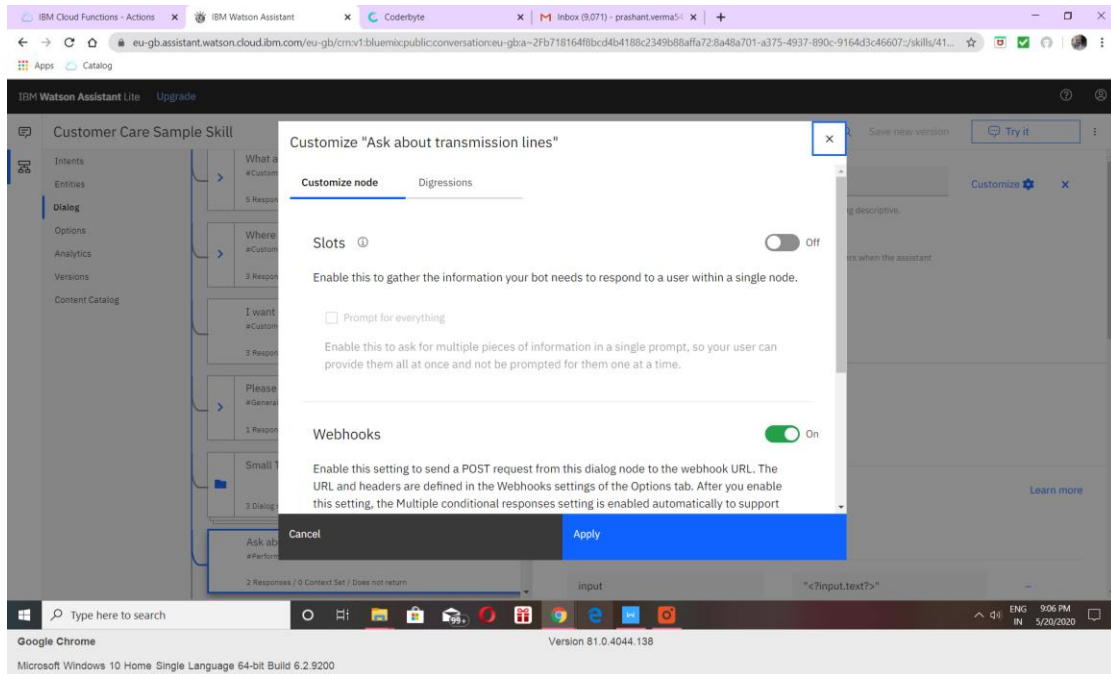
[Add header](#) [Add authorization](#)

Next step

To trigger this webhook from an individual dialog node, enable the webhook from the Customize page in node details. [Go to dialog](#).

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

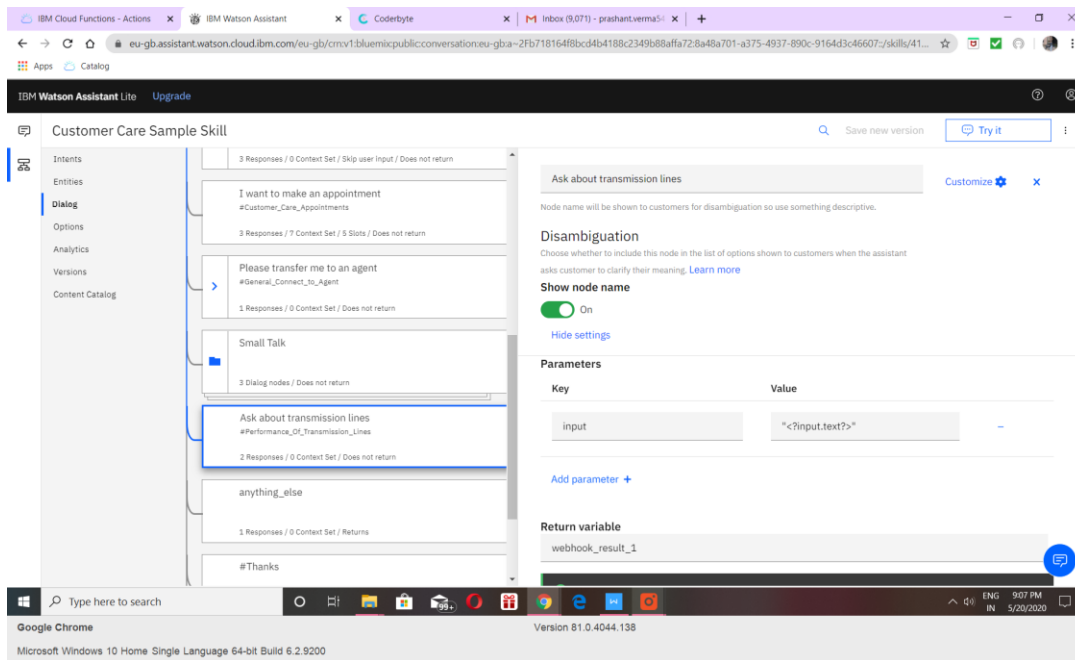
Return to the Dialog tab, and click on the Ask about product node. From the details panel for the node, click on Customize, and enable Webhooks for this node:



**Click Apply.**

**The dialog node should have a Return variable [1] set automatically to \$webhook\_result\_1. This is the variable name you can use to access the result from the Discovery service query.**





**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:**

#### Return variable

\$webhook\_result\_1

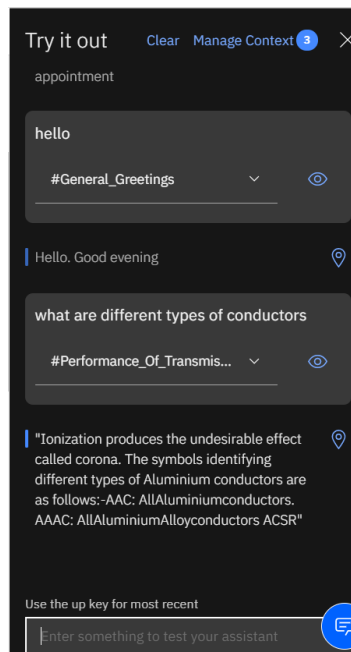
#### Then respond with

	IF ASSISTANT RECOGNIZES	RESPOND WITH		
1	\$webhook_result_1	\$webhook_result_1		
2	anything_else	Try again later		

Add response 

### Test in Assistant Tooling

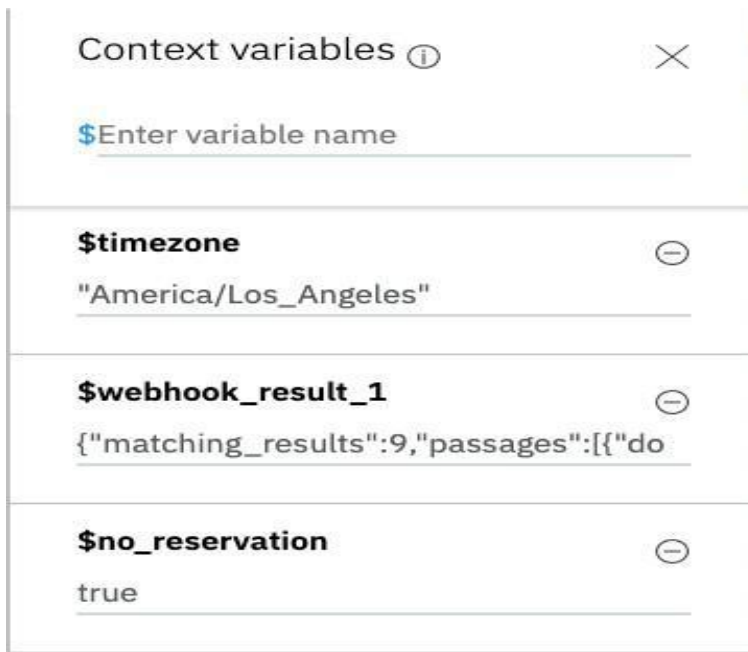
From the Dialog panel, click the Try it button located at the top right side of the panel.  
Enter some user input:



Note that the input "what are the different types of conductors?" has triggered our Ask about transmission lines dialog node, which is indicated by the #Performance\_Of\_Transmission\_Lines response.

And because we specified that \$webhook\_result\_1.passages be the response, that value is displayed also.

You can also verify that the call was successfully completed by clicking on the Manage Context button at the top right. The response from the Discovery query will be stored in the \$webhook\_result\_1 variable:



Context variables ⓘ	
\$Enter variable name	
\$timezone	⊖ "America/Los_Angeles"
\$webhook_result_1	⊖ {"matching_results":9,"passages":[{"do
\$no_reservation	⊖ true

5.Create flow and configure node:

### 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

**Edit assistant V2 node**

Delete Cancel Done

**Properties**

Name: My first assistant

Username: Username

Password: Password

API Key: .....

Service Endpoint: https://api.eu-gb.assistant.watson.cloud.ibm.com

Assistant ID: d3fa58b9-7946-4784-b120-5e025c58c9ba

Timeout Period: Leave empty to disable

☒ Switch on Debug

☒ Enabled

After entering all the information click on Done

Drag inject node on to the flow from the Input

section Drag Debug on to the flow from the

output section Double-click on the inject node

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

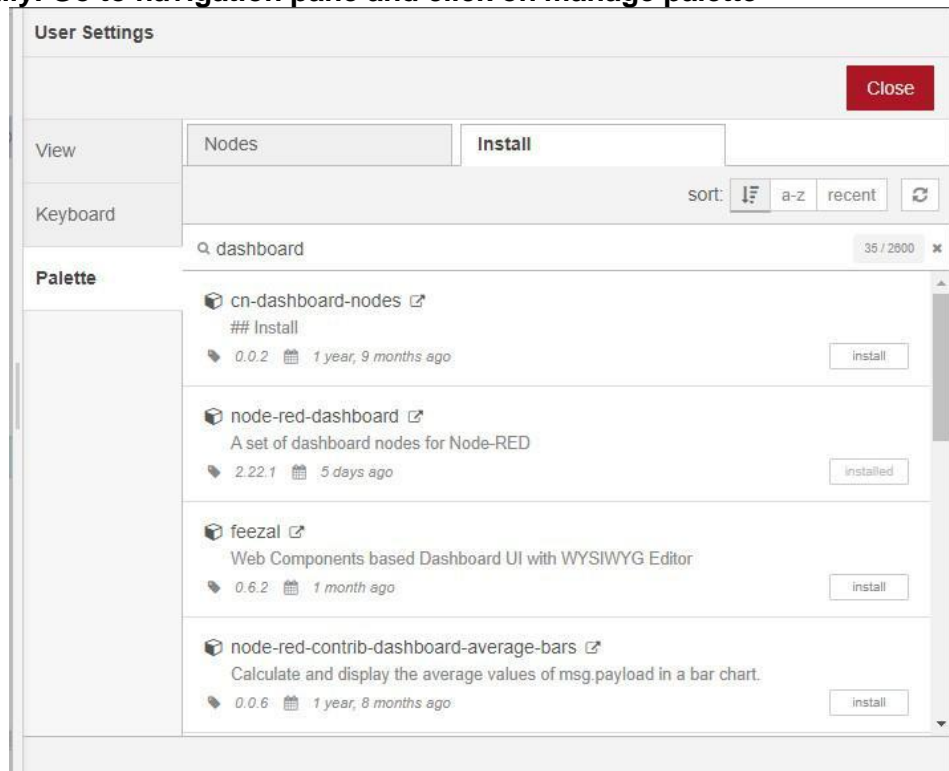
and click on done • Connect the nodes as shown below and click on Deploy

Re-inject the flow and observe the parsed output

**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**

**Double click on the “form” node to configure**

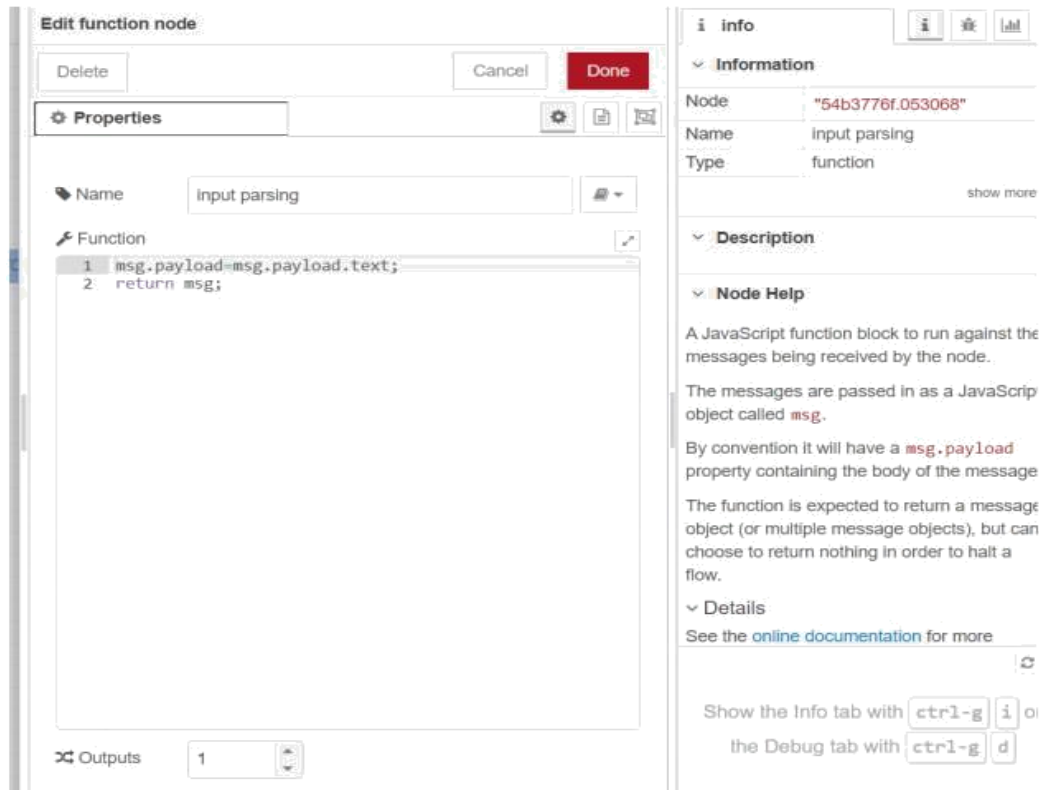
**Click on the edit button to add the “Group” name and “Tab” name**

**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**



**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 “You” and click on Done**

**Double click on the second text node, change the label as “Bot” and click on Done**

**Connect the output of “input parsing” function node to “ You” text node and output of “Parsing” function node to the input of “Bot” text node Click on Deploy**

## 5.FLOWCHART

**At first go to manage pallette and install dashboard.**

**Now,Create the flow with the help of following node:**

**Inject**

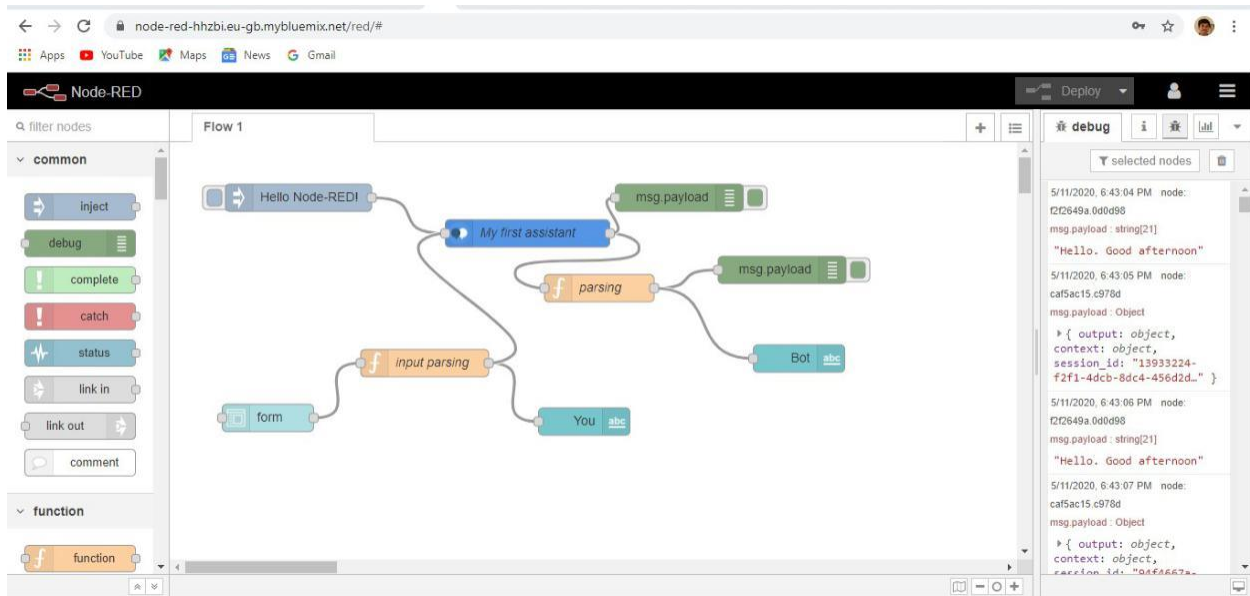
**Assistant**

**Debug**

Function

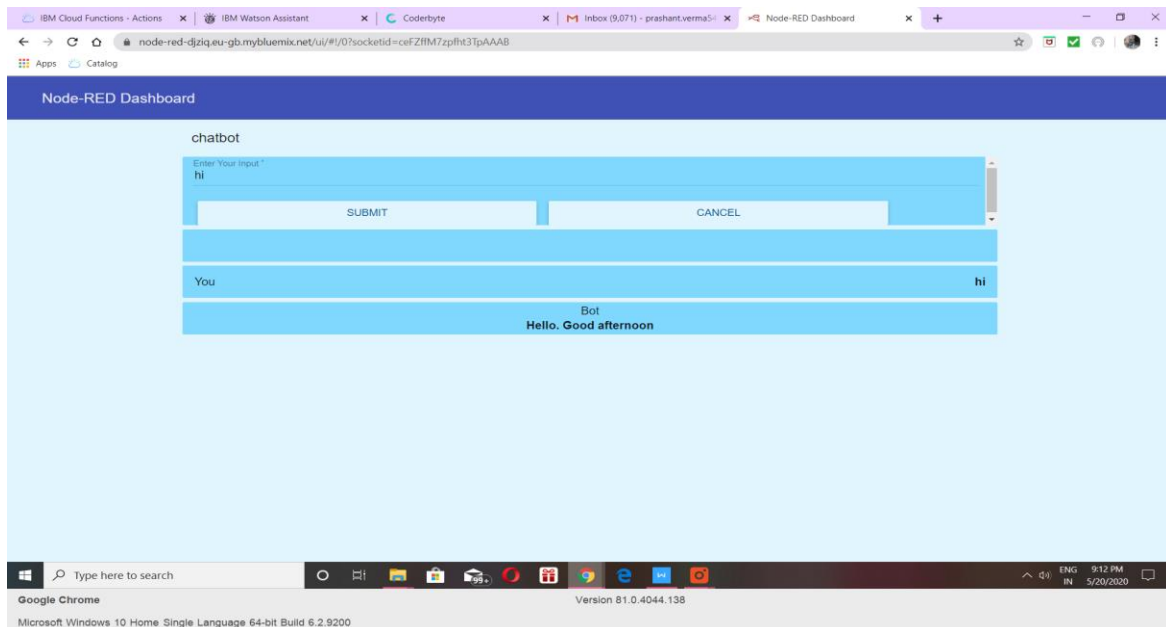
Ui\_Form

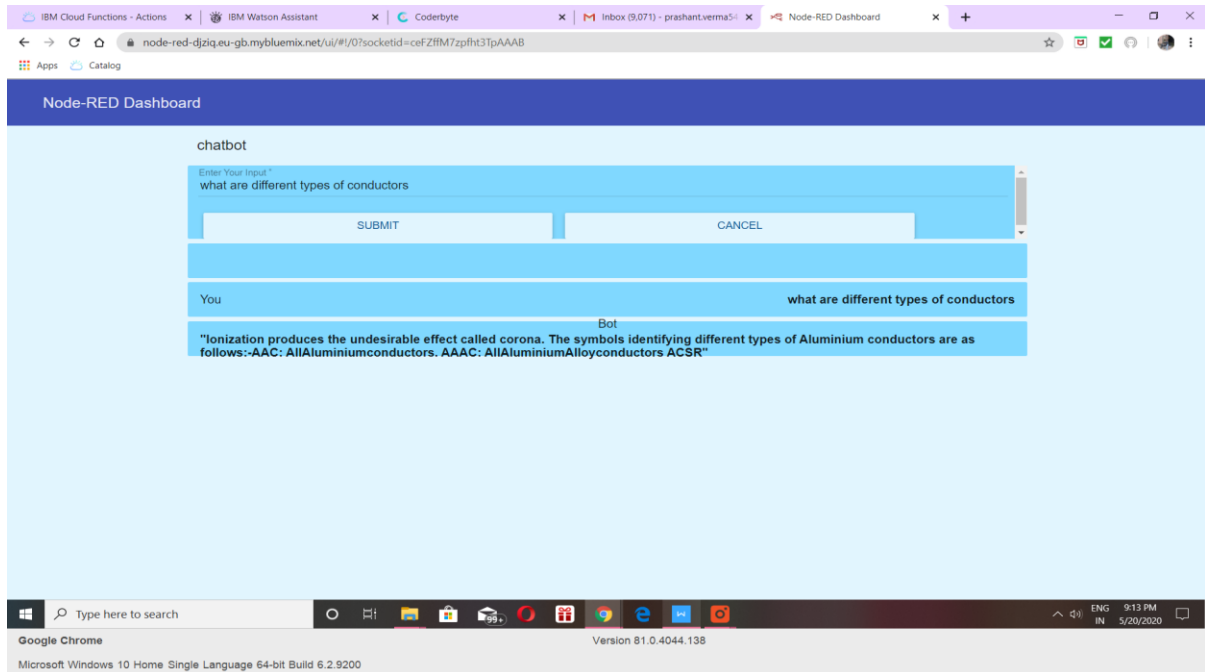
Ui\_Text



## 6.RESULTS

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





## 7.ADVANTAGES & DISADVANTAGES

### Advantages:

**Companies can deploy chatbots to rectify 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 above procedure and all we successfully created Intelligent helpdesk smart chartbot 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

#### Reference:

1. [https://www.ibm.com/cloud/architecture/tutorials/cognitive\\_discovery](https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery)
2. <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>
3. <https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/>
4. <http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red>
5. <https://github.com/IBM/watson-discovery-sdu-with-assistant>
6. <https://www.youtube.com/watch?v=Jpr3wVH3FVA>

