**Intelligent Customer Help Desk with Smart Document Understanding**

**Name: Yash Gandhi**

**Category: Artificial Intelligence**

**Internship at:** smartinternz.com@2020

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

1. **INTRODUCTION**
   1. **Overview**

* 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 predetermined 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 preloaded 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. **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

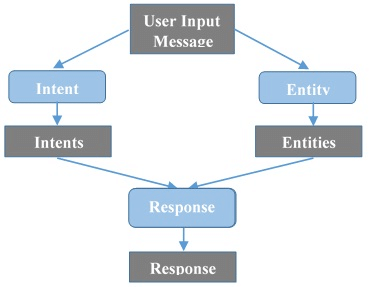
1. **LITERATURE SURVEY**
   1. **Existing Problem**

In some customer help desk all the agents are occupied when there is more traffic and because of that customer should wait more. Also there are some chatbots that not give proper answer of customer queries.

* 1. **Proposed Solution**

As solution of the problem I made a intelligent Chabot that is always free and it can chat with you and can also give solution to your problem. It also has smart document understanding so it will read the owner’s manual and gives answer based on that.

1. **THEORITICAL ANALYSIS**
   1. **Block Diagram**

****

* 1. **Hardware/Software Designing**

1. Create IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud function action
4. Configure Watson assistant
5. Create flow and configure node to integrate all services
6. Deploy and run the web dashboard
7. **EXPERIMENTAL INVESTIGATIONS**
8. **Create IBM Cloud Services**

Creating following services:

* + Watson Discovery
  + Watson Assistant
  + Node Red
  + IBM Cloud Function

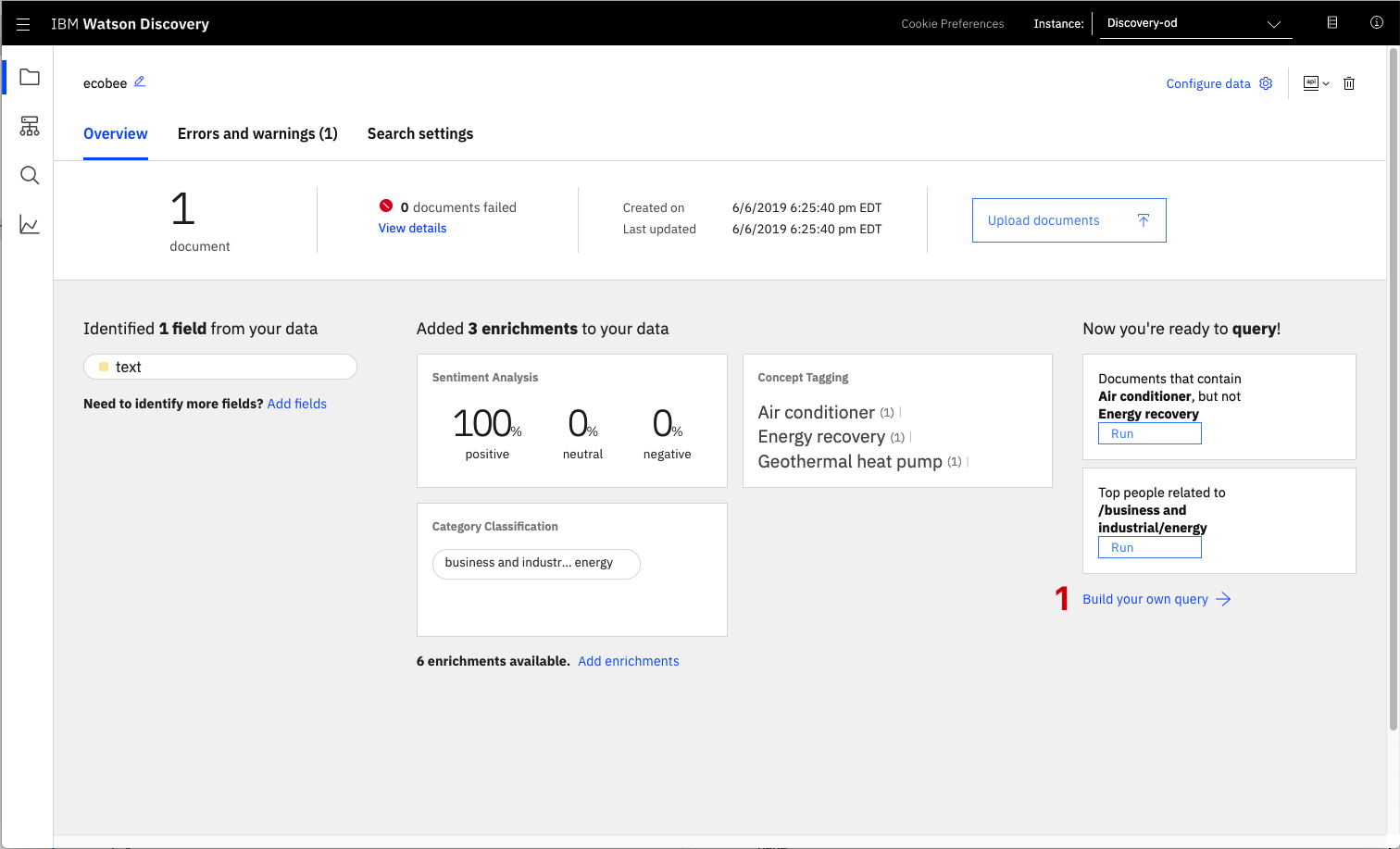
1. **Configure Watson Discovery**

Import the document

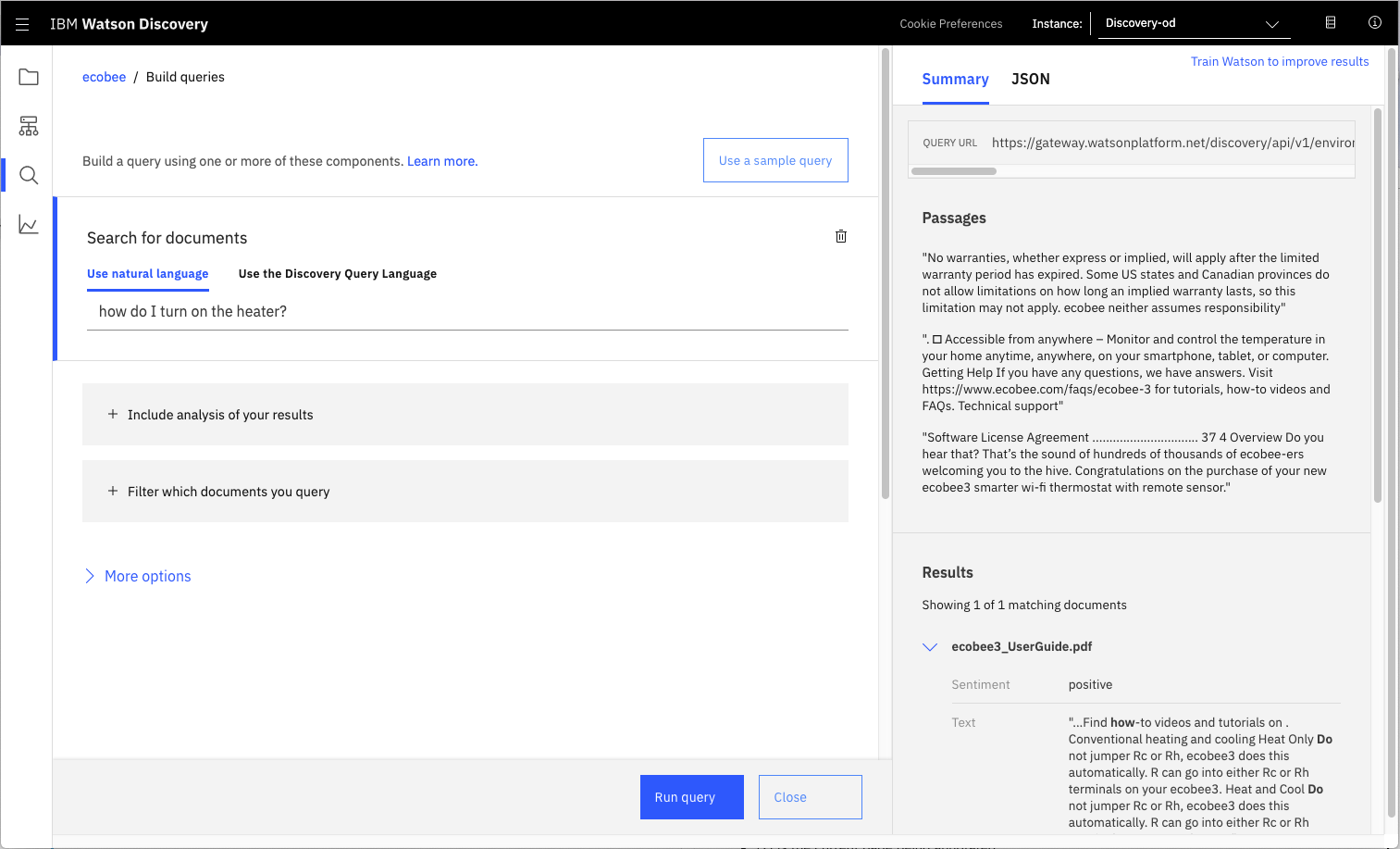
As shown below, launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option. Give the data collection a unique name. When prompted, select and upload the ecobee3\_UserGuide.pdf file located in the data directory of your local repo.

The Ecobee is a popular 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.



Click the Build your own query [1] button.

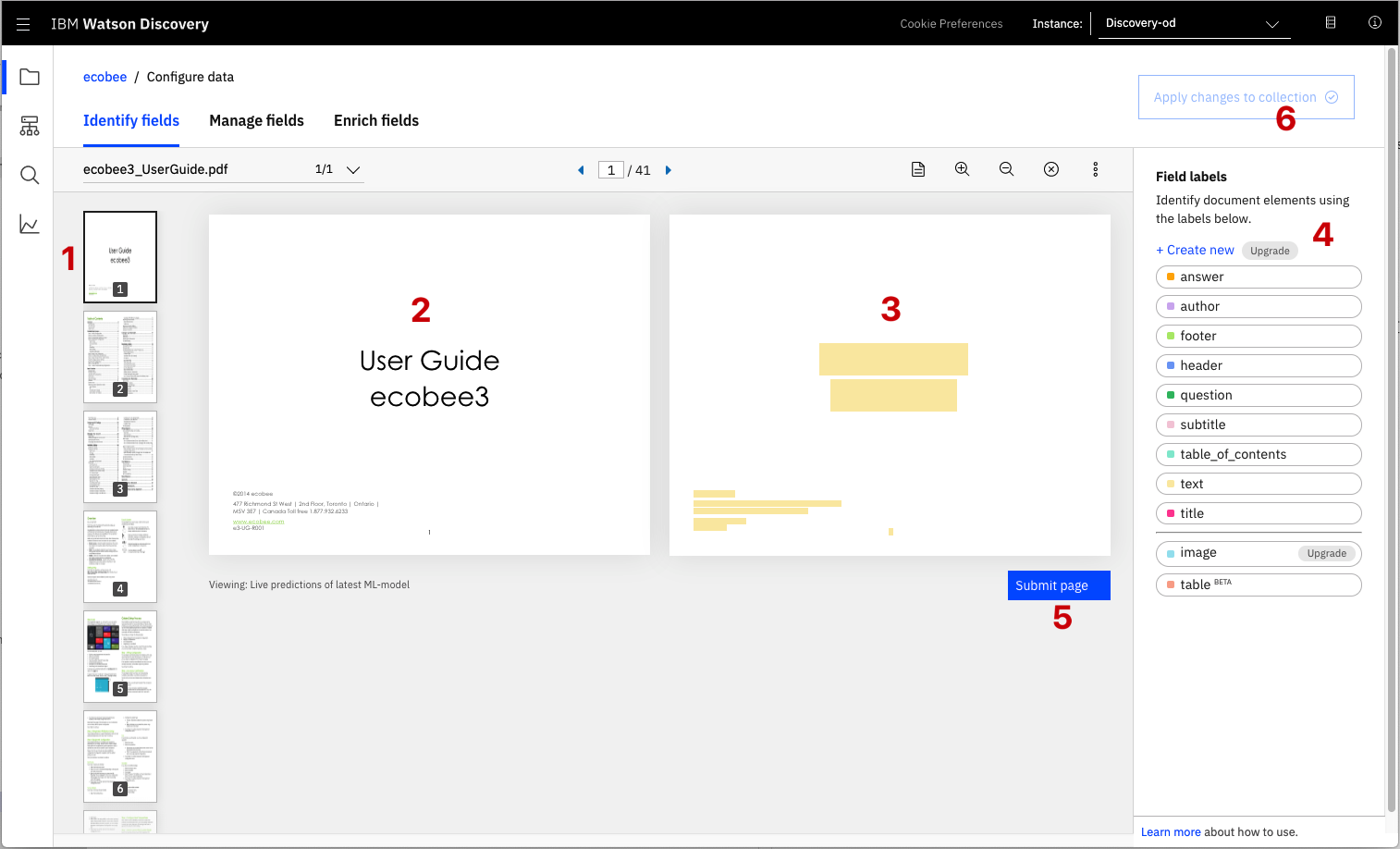


Enter queries related to the operation of the thermostat and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question.

#### 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 layout of Indentify field tab of the SDU annotation panal.



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

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

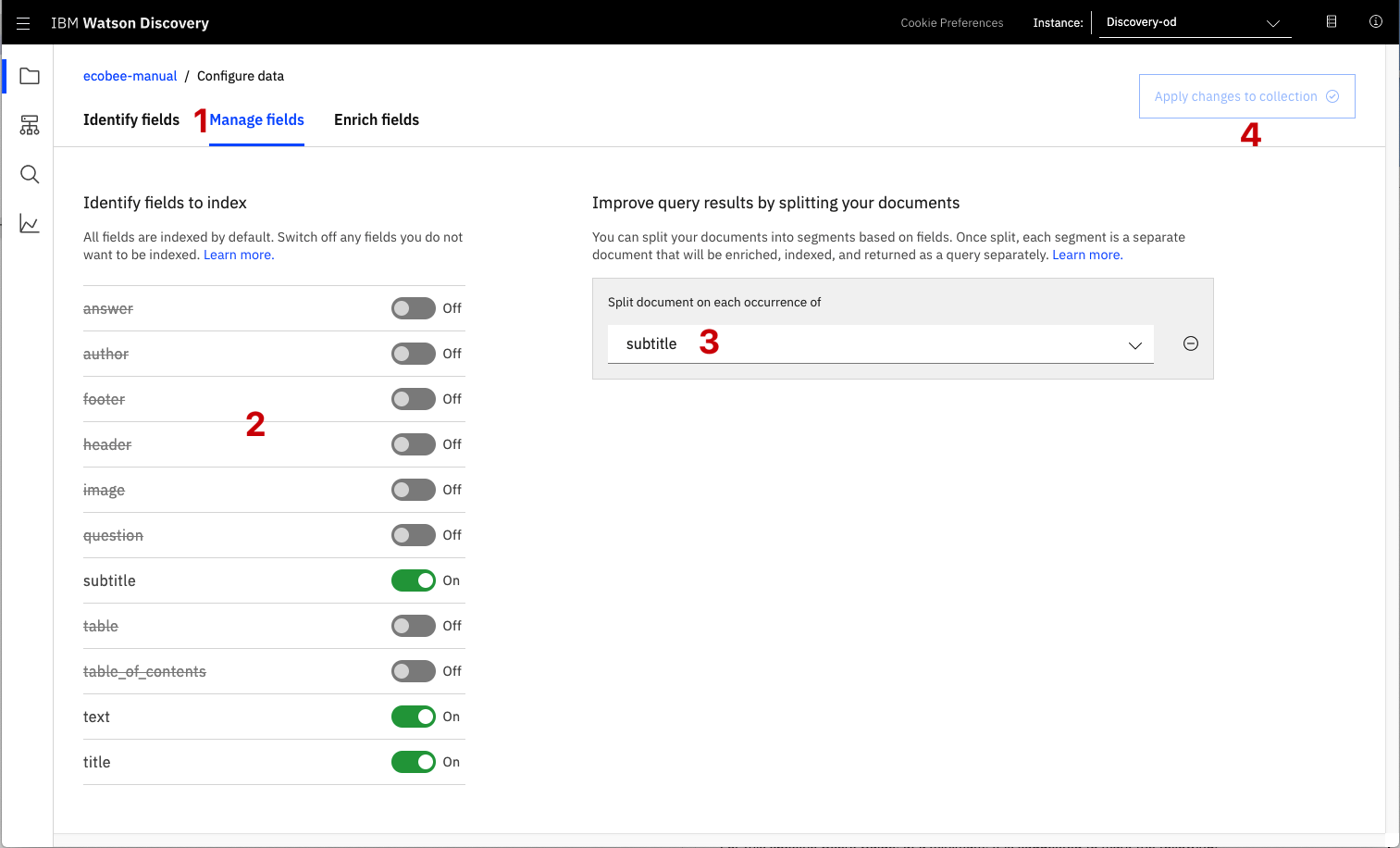
As you go though the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages. Once you get to a page that is already correctly annotated, you can stop, or simply click Submit [5] to acknowledge it is correct. The more pages you annotate, the better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

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

* The main title page as title
* The table of contents (shown in the first few pages) as table\_of\_contents
* All headers and sub-headers (typed in light green text) as a subtitle
* All page numbers as footers
* All warranty and licensing infomation (located in the last few pages) as a footer
* All other text should be marked as text.

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

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



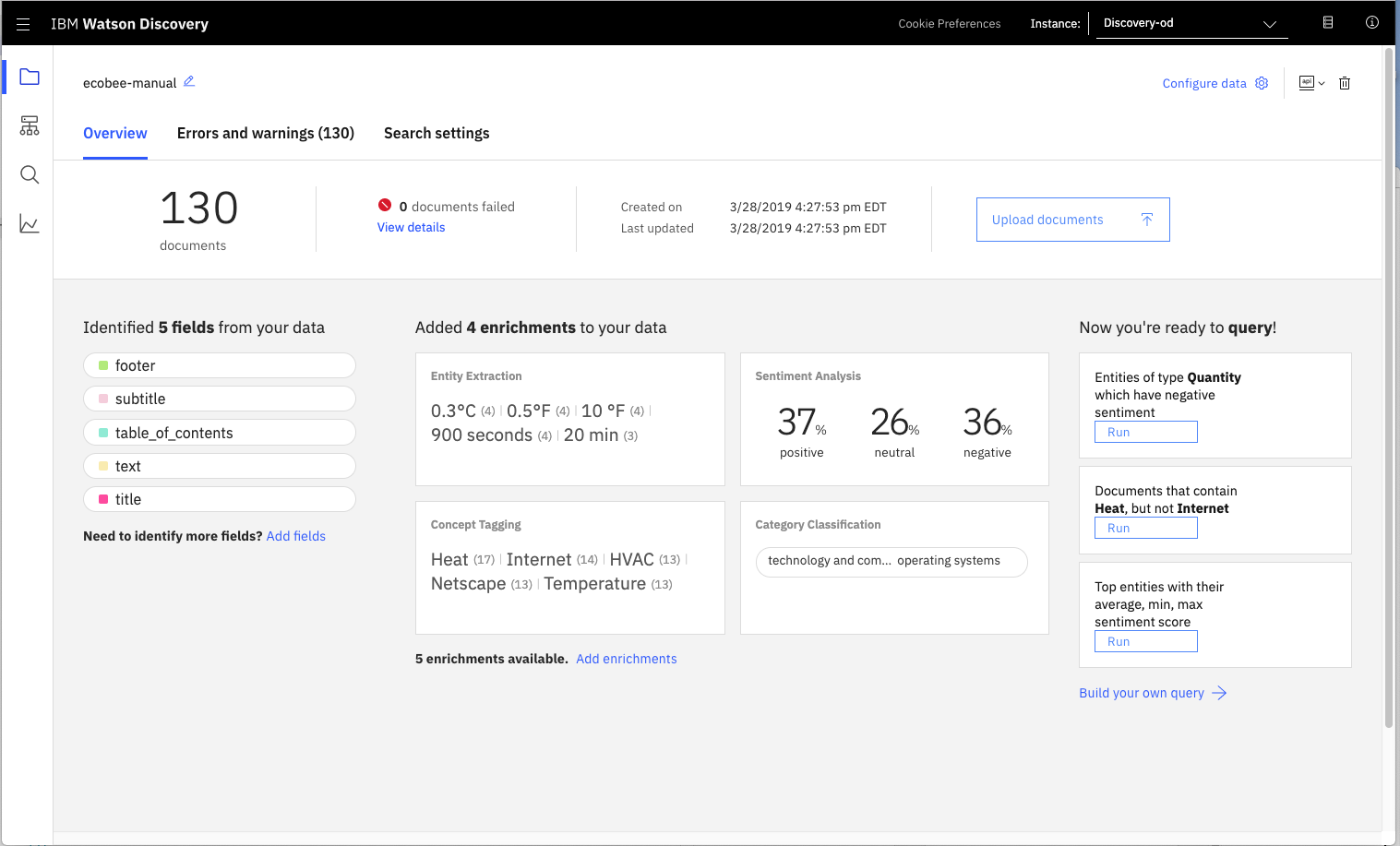
[2] Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except subtitles and text.

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

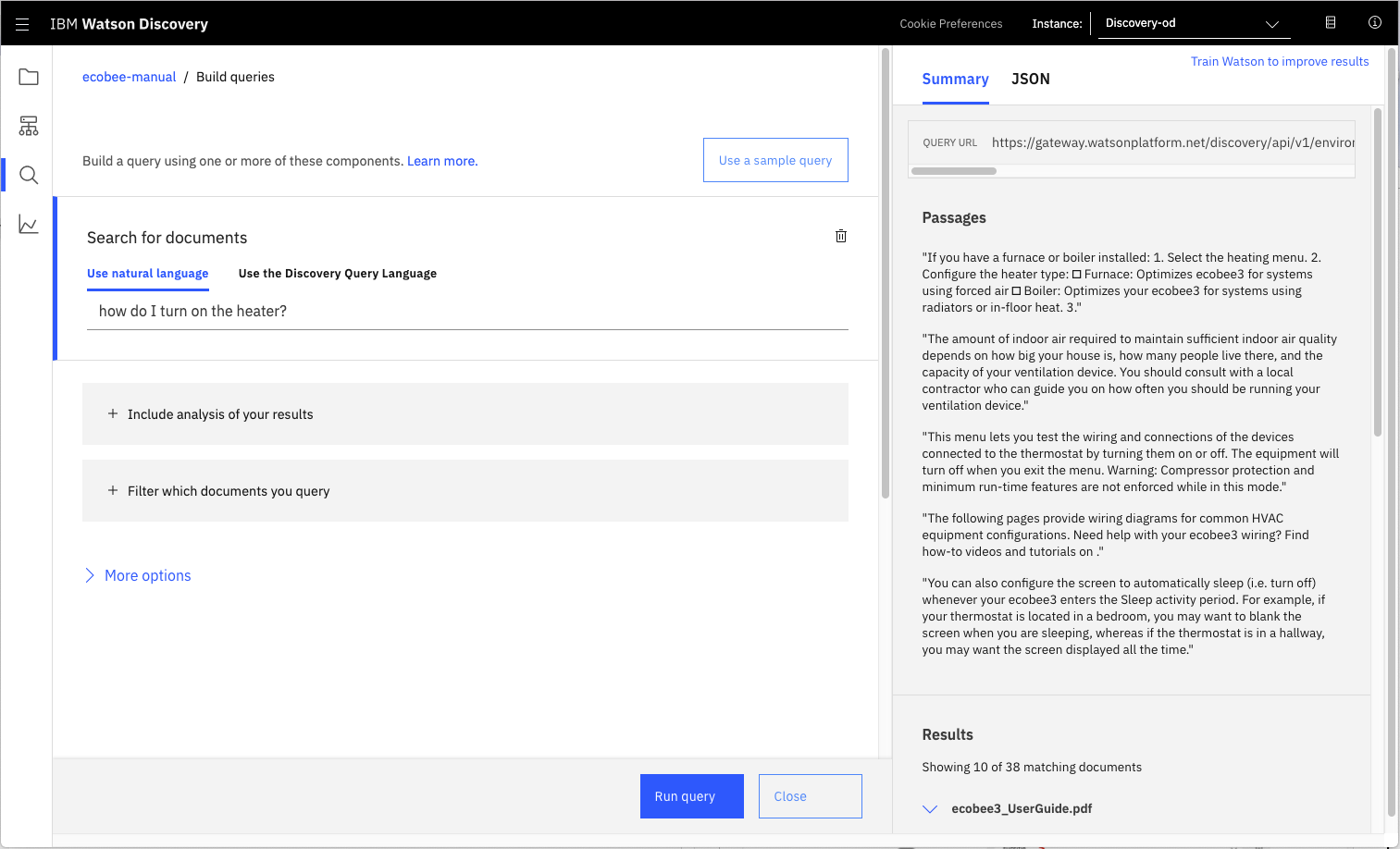
Click [4] to submit your changes.

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:



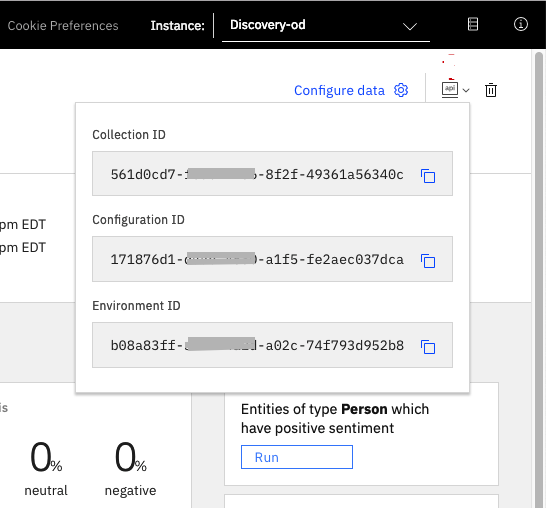
Return to the query panel (click Build your own query) and see how much better the results are.



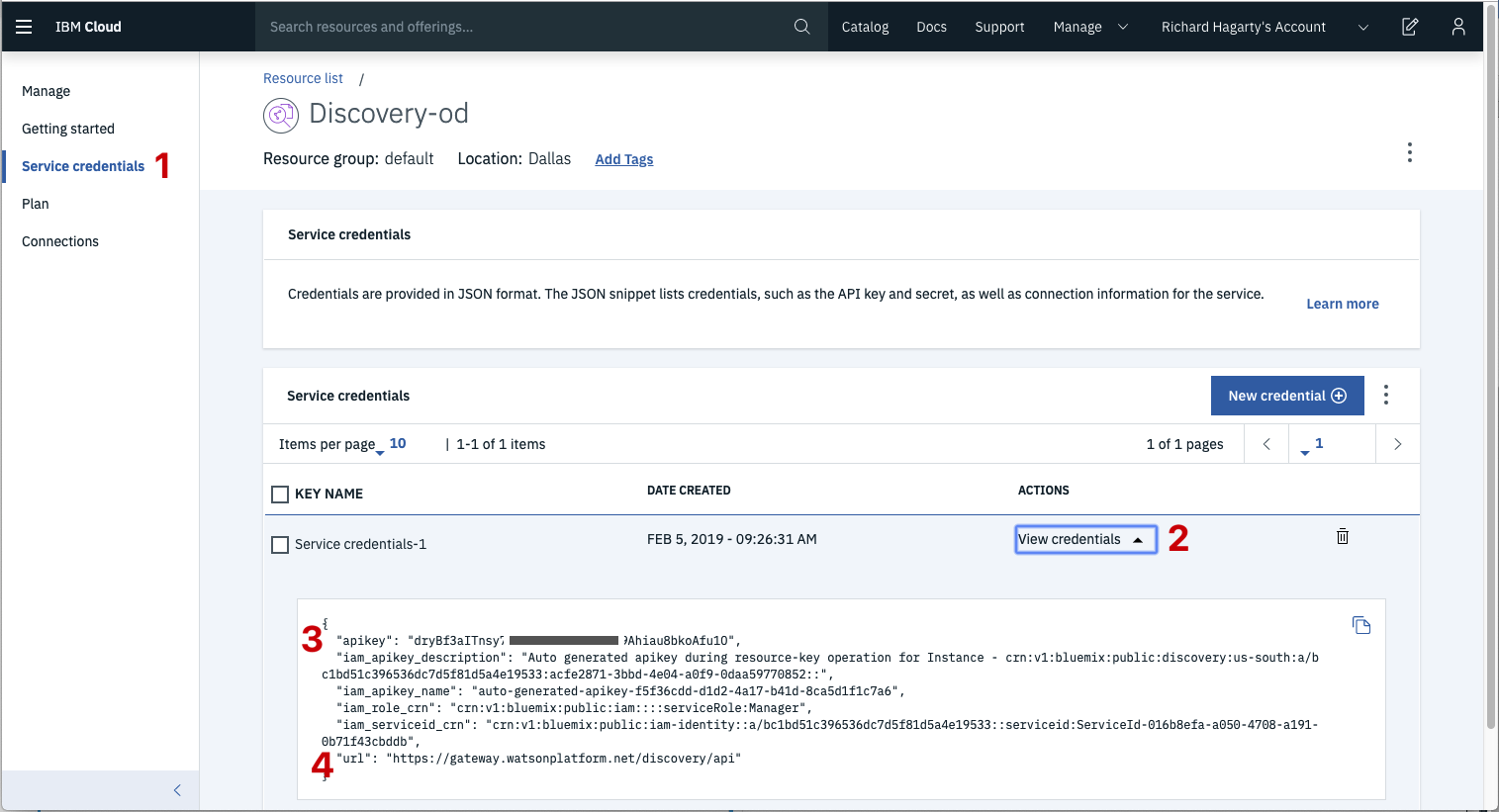
Store credentials for future use

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:

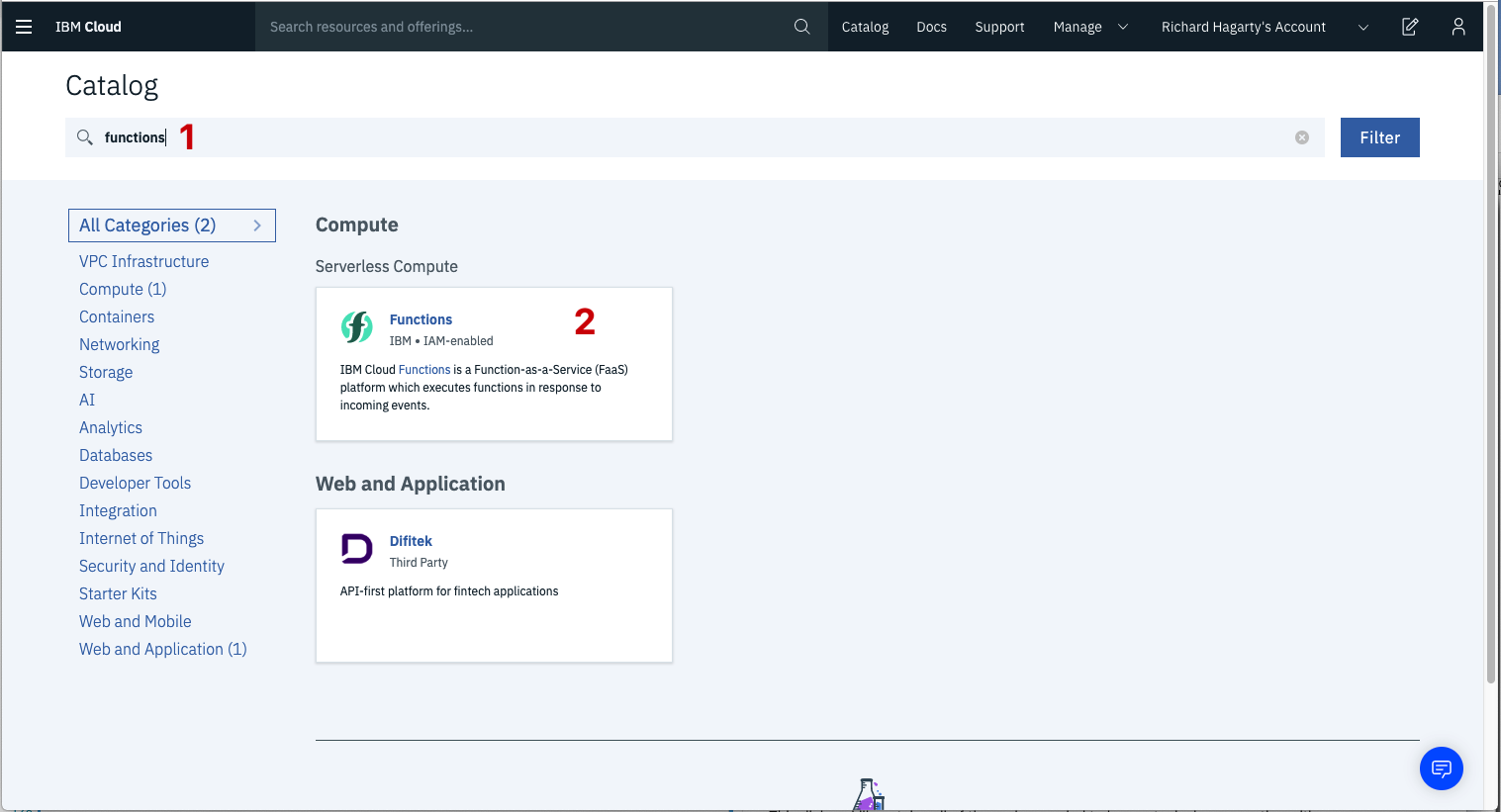


Click the View credentials [2] drop-down menu to view the IAM apikey [3] and URL endpoint [4] for your service.

1. **Create IBM Cloud Functions action**

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

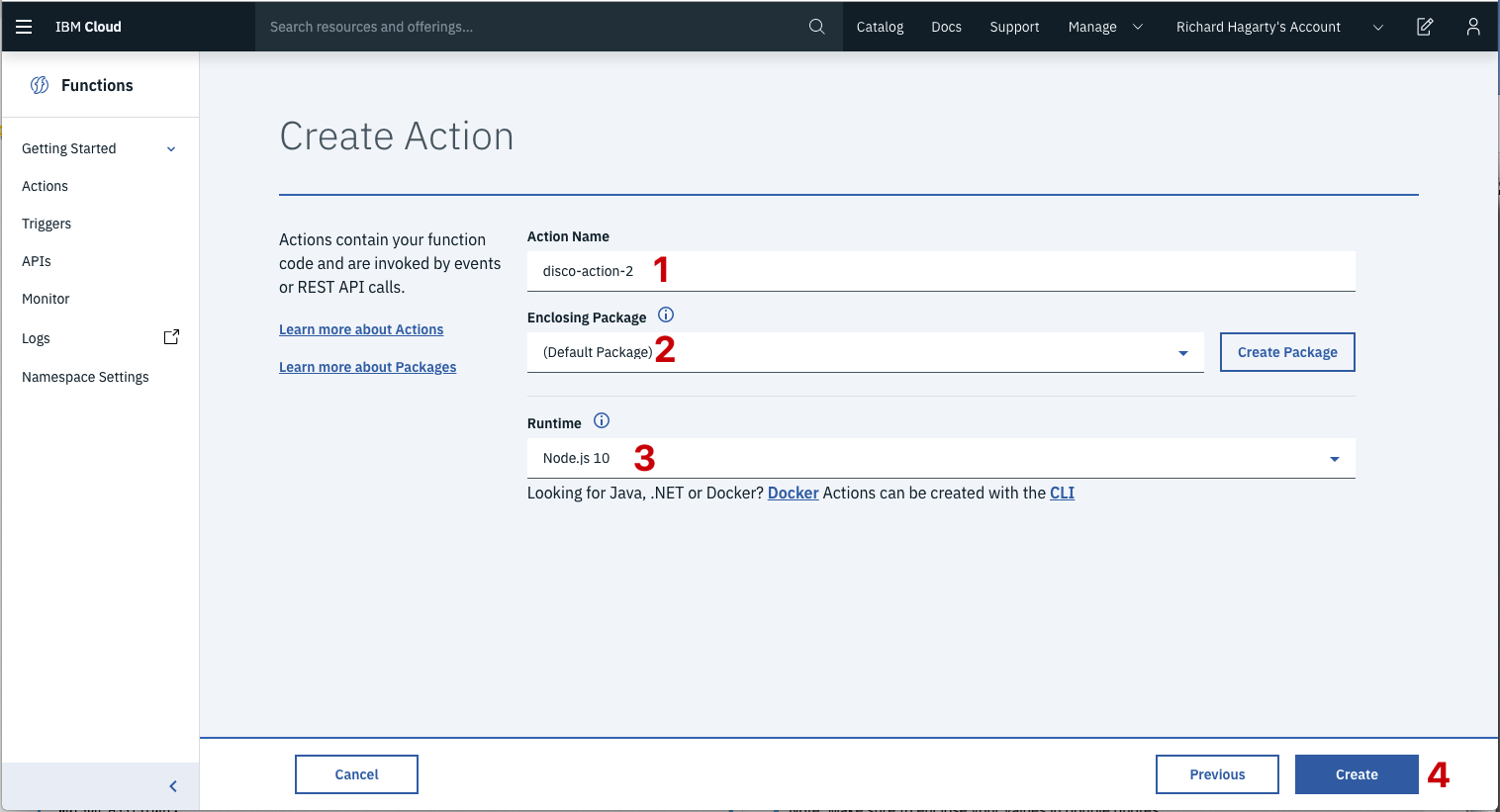
Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter [1], then select the Functions card [2]:



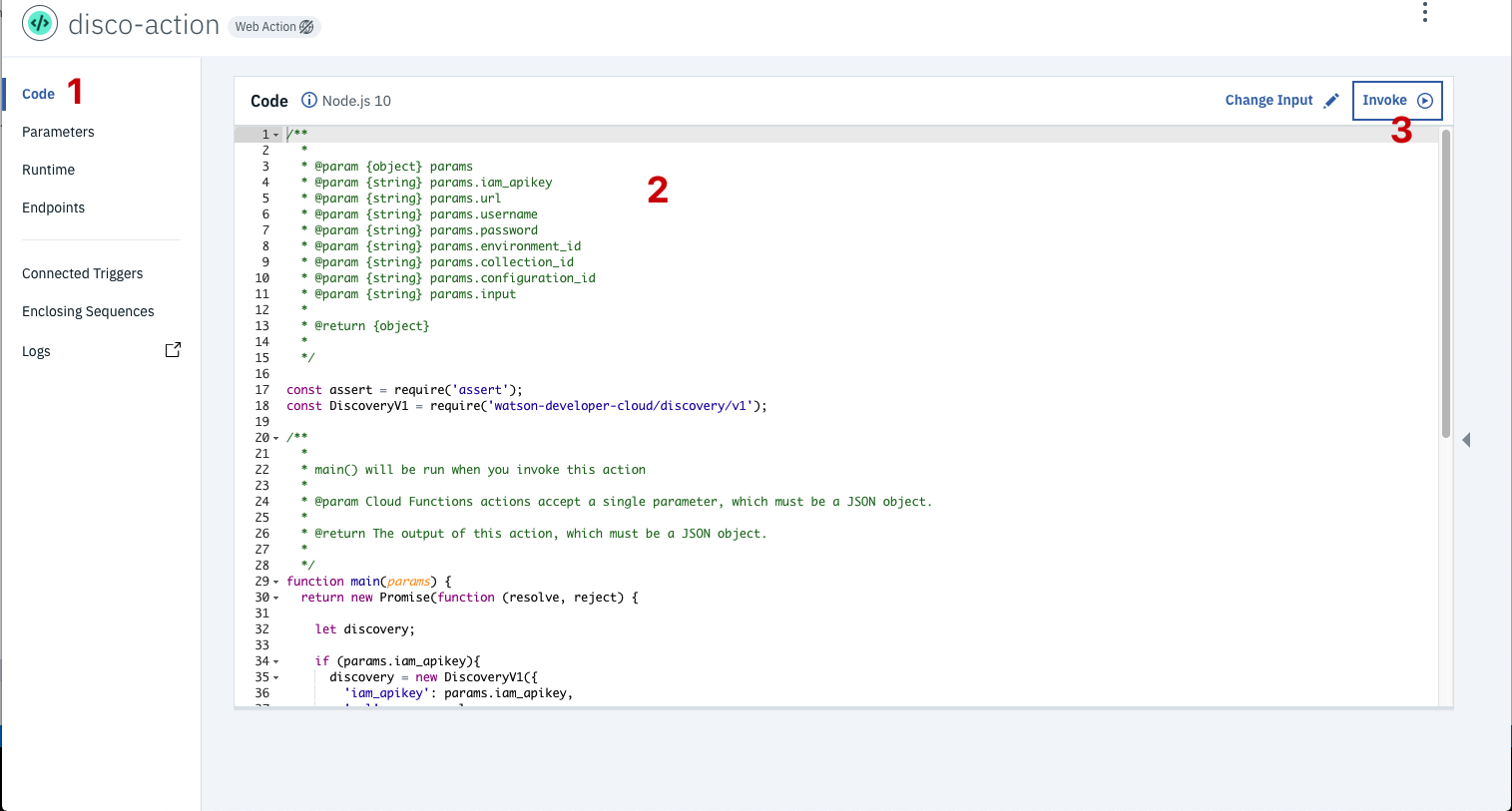
From the Functions main panel, click on the Actions tab. Then click on Create.

From the Create panel, select the Create Action option.

On the Create Action panel, provide a unique Action Name [1], keep the default package [2], and select the Node.js 10 [3] runtime. Click the Create button [4] to create the action.



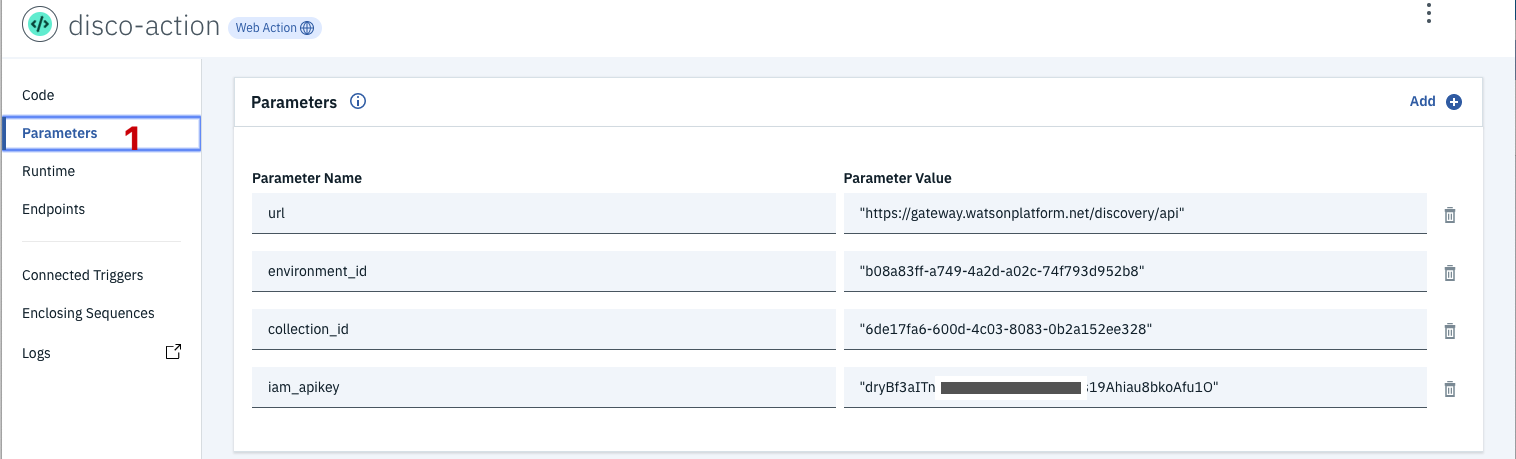
Once your action is created, click on the Code tab [1]:



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

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

Select the Parameters tab [1]:

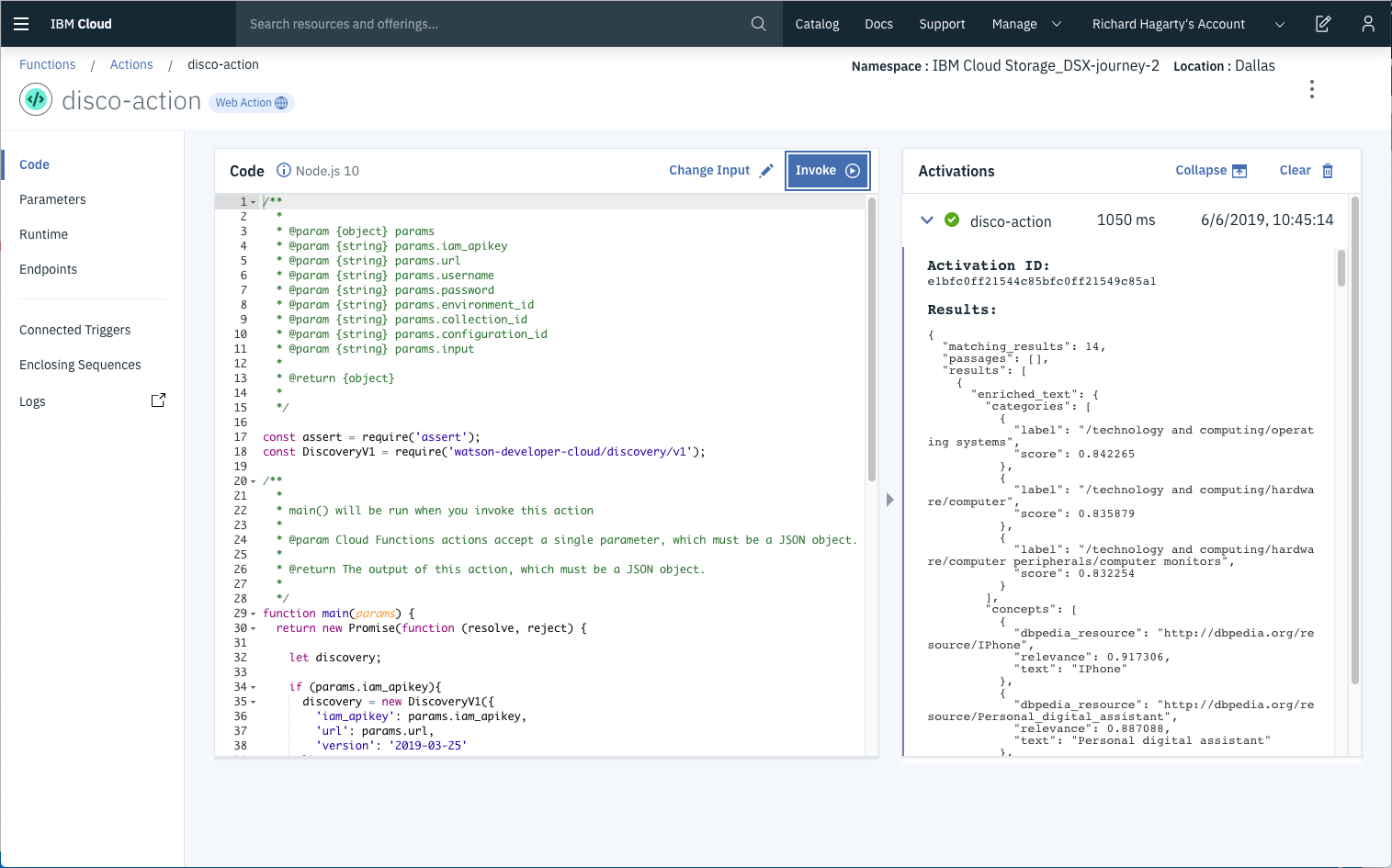


Add the following keys:

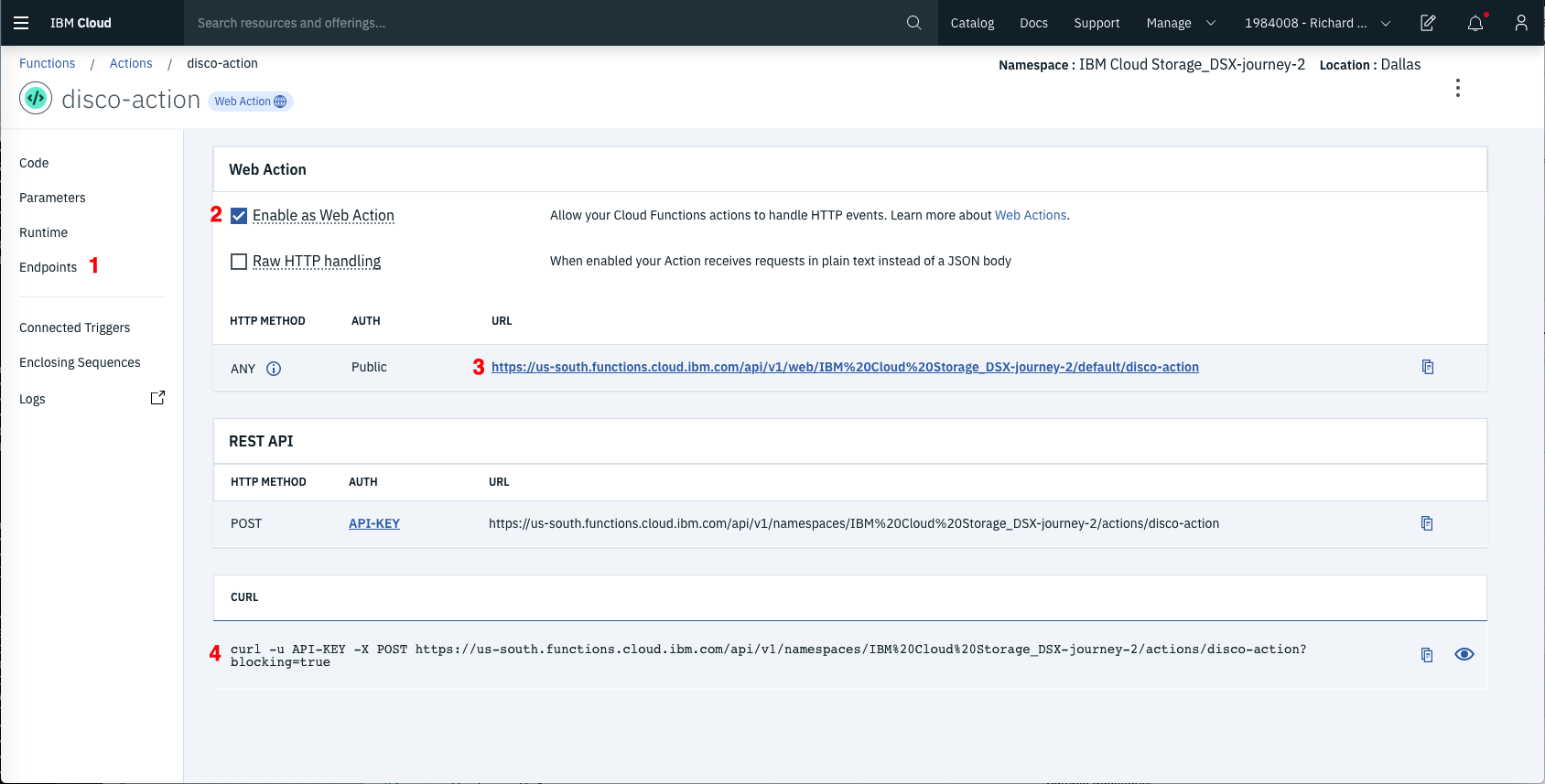
* url
* environment\_id
* collection\_id
* iam\_apikey

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 provied curl command [4]. If it fails, re-check your parameter values.

1. **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 intent**

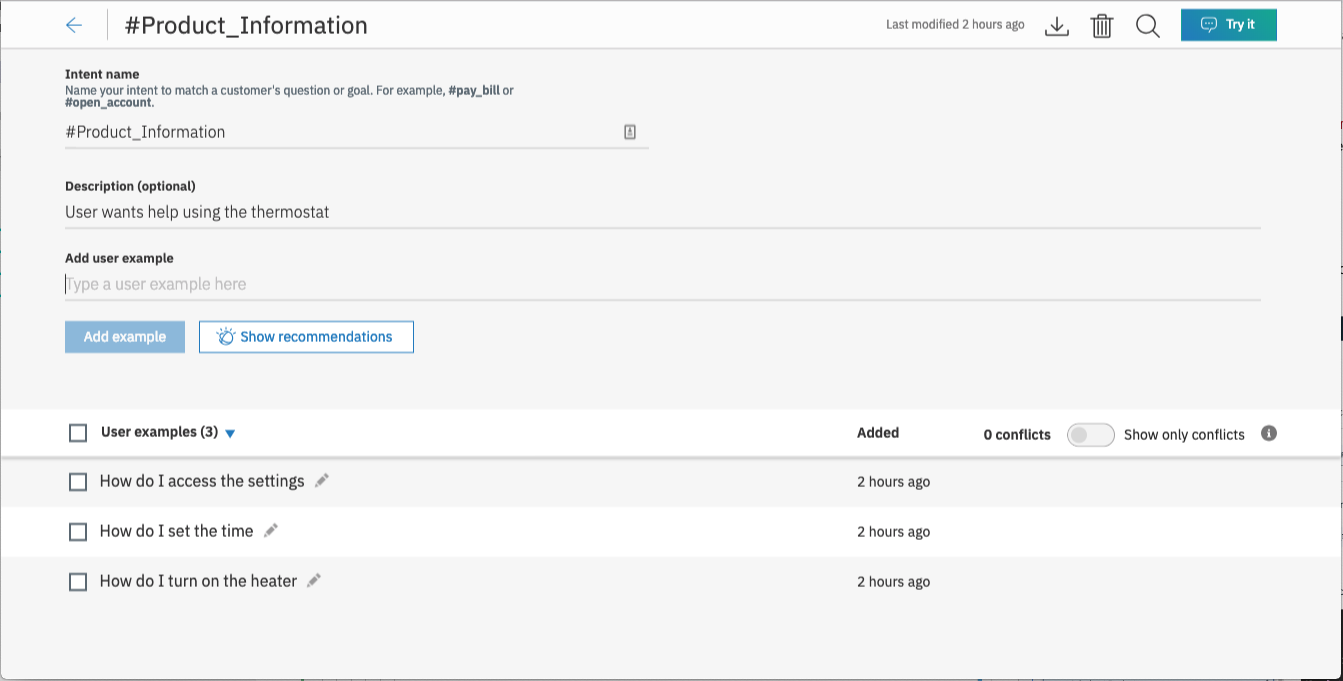
The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this.

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.

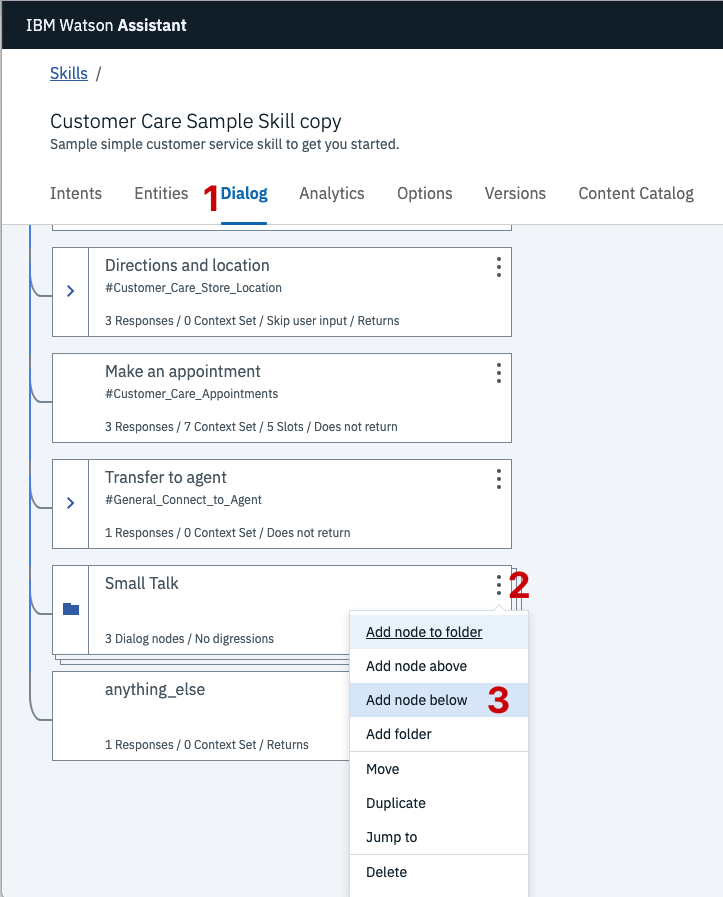
Click the Create intent button.

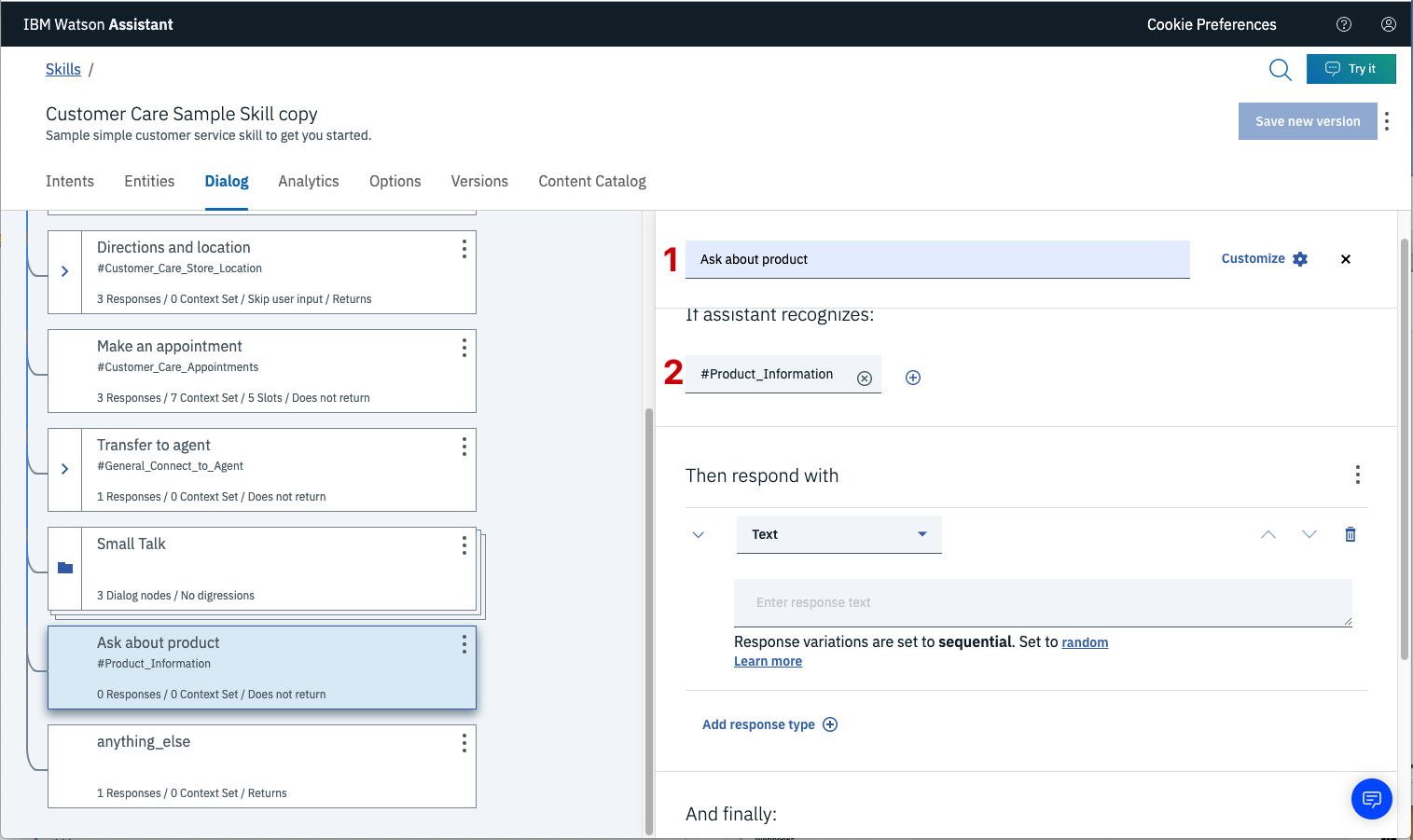
Name the intent #Product\_Information, and at a minimum, enter the following example questions to be associated with it.



**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.



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

This means that if Watson Assistant recognizes a user input such as "how do I set the time?", it will direct the conversation to this node.

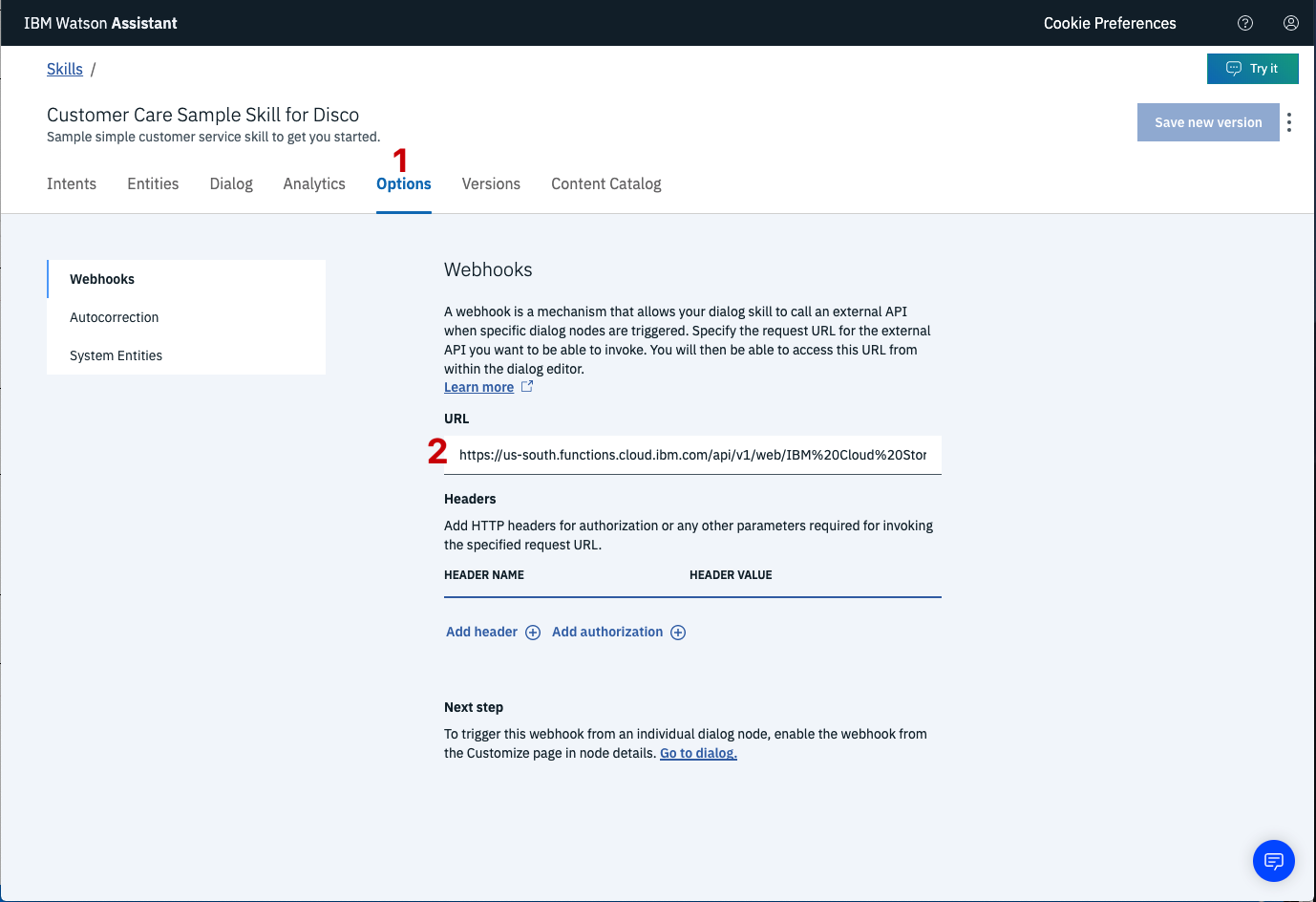
Enable webhook from Assistant

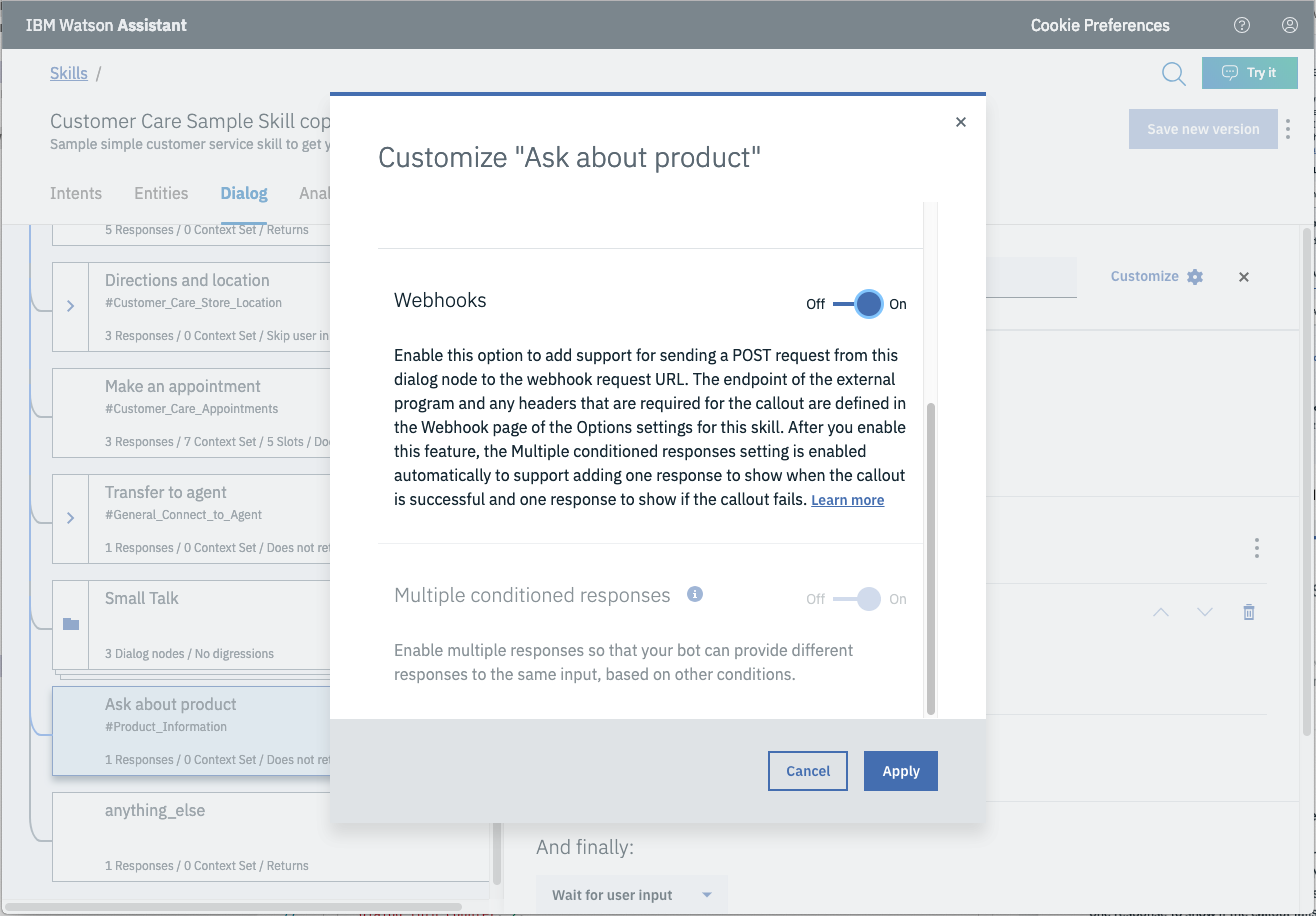
Set up access to our WebHook for the IBM Cloud Functions action you created in Step #4.

Select the Options tab [1]:

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

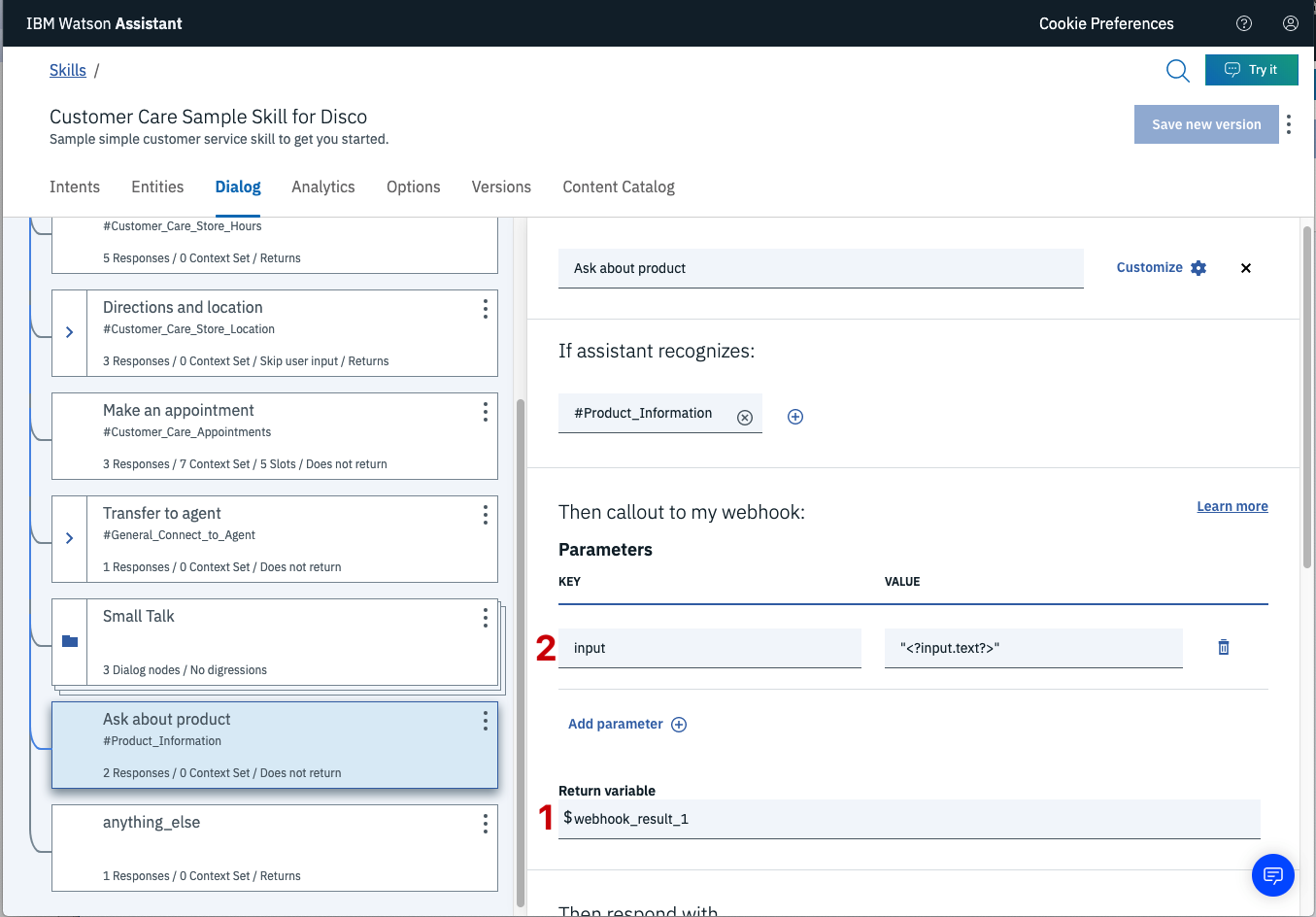
Return to the Dialog tab, and click on the Ask about product node. From the details panel for the node, click on Customize, and enable Webhooks for this node:





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.



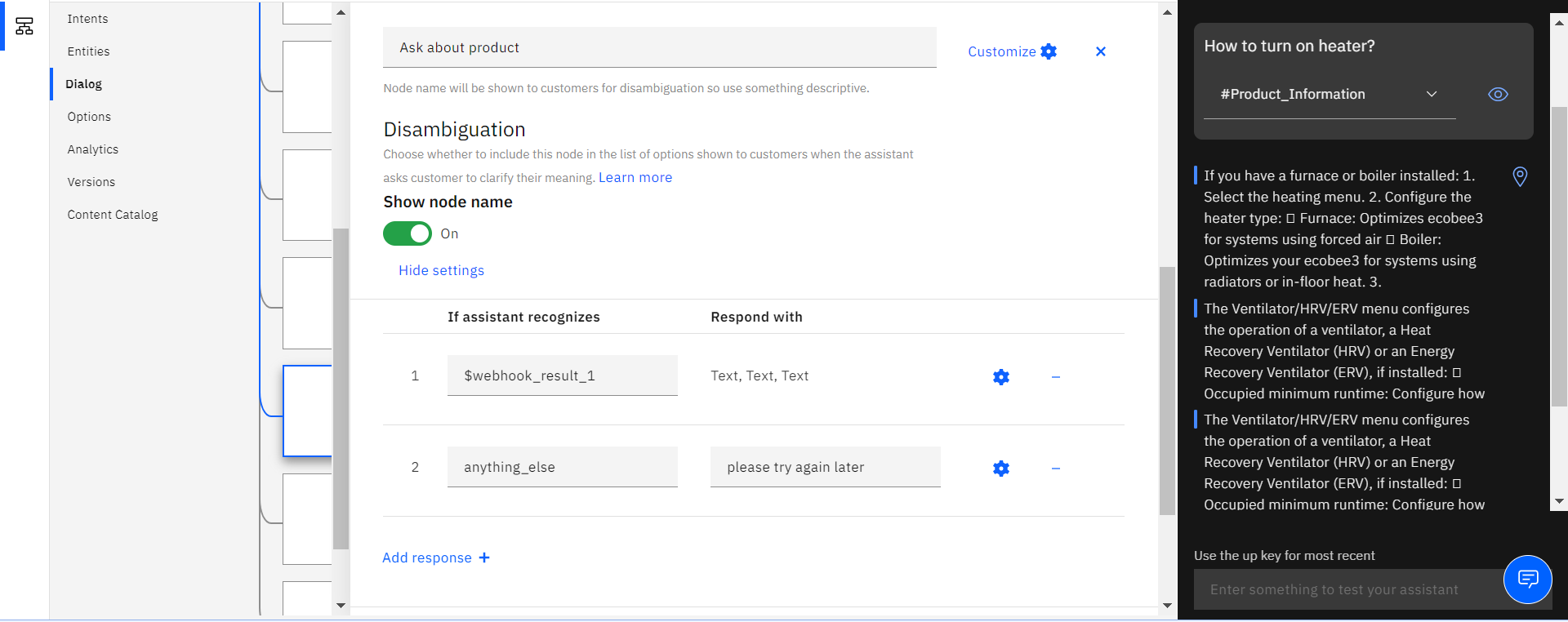
You will also need to pass in the users question via the parameter input [2]. The key needs to be set to the value:

"<?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:

Add “<?webhook\_result\_1.Passages[0].passage\_text?>” in respond with in assiatant responds block as shown below.

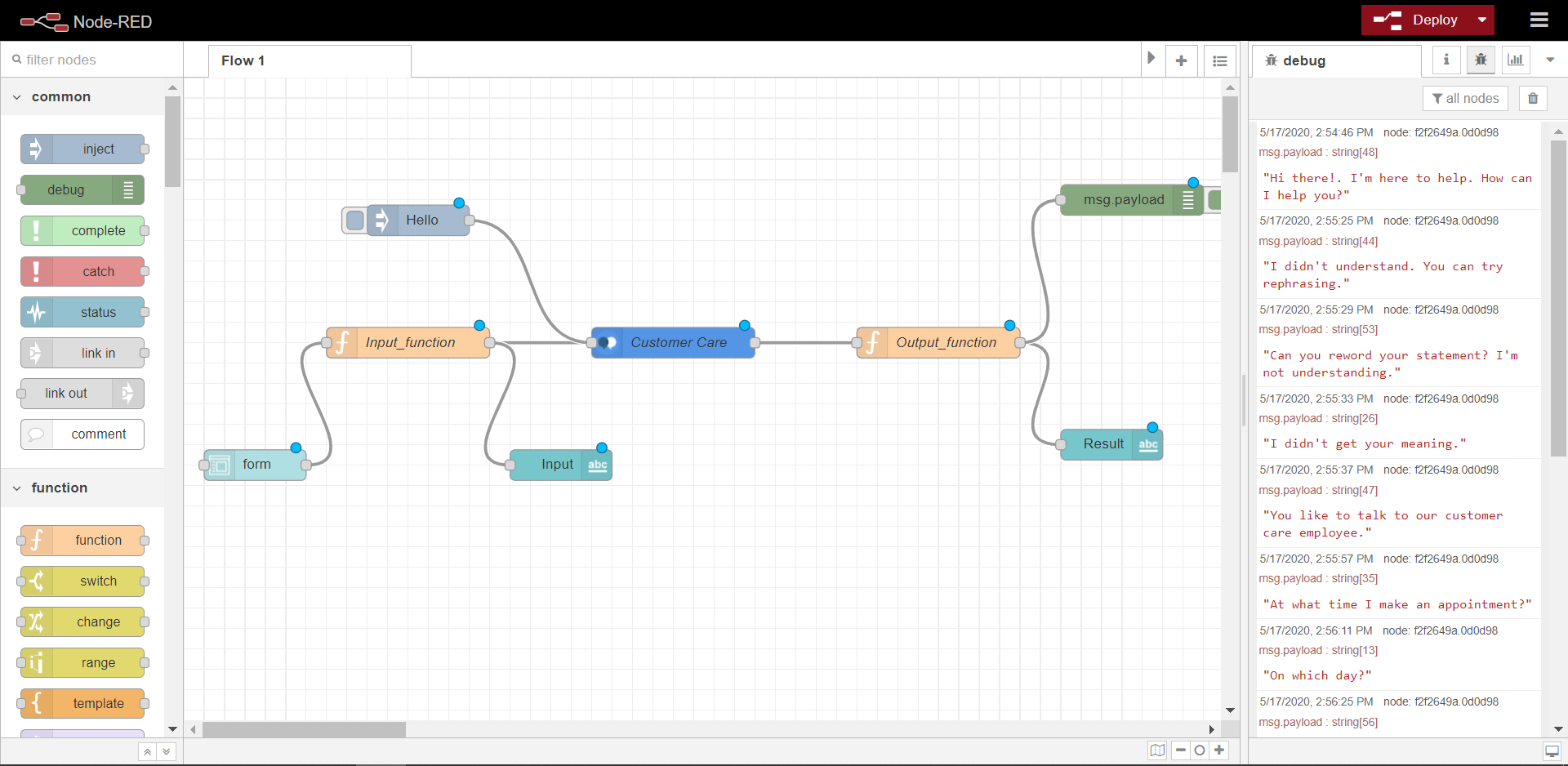


1. **Create flow and configure node to integrate all services**

First install dashboard from manage palette.

Now, create the flow with the help of following node:

* Inject
* Assiatant
* Debug
* Function
* Ui\_Form
* Ui\_Text

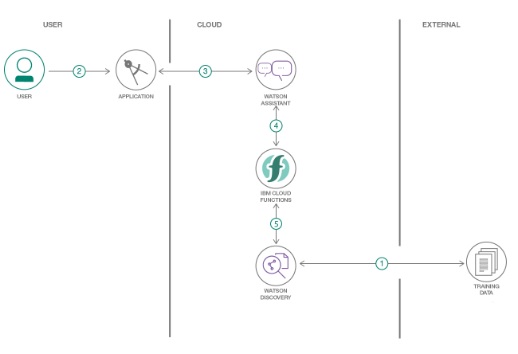


1. **Deploy and run Node Red app.**

Now you can click on deploy button and then see the ui by using

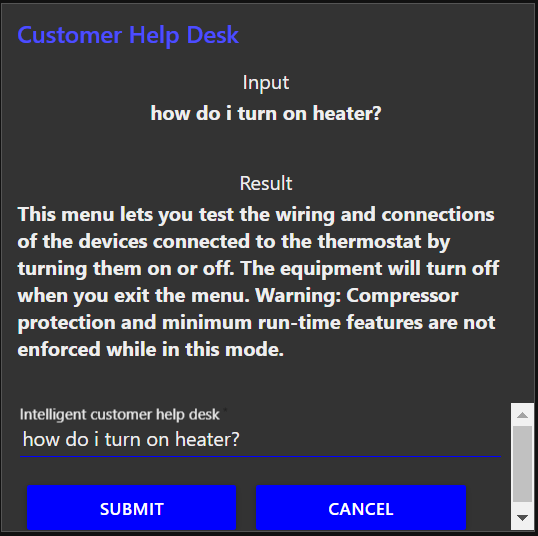
URL : <https://node-red-gwhup.eu-gb.mybluemix.net/ui>

1. **FLOWCHART**

****

1. **RESULT**

Finally our Node Red dashboard integrates all the components and display in the Dashboard UI by typing URL- <https://node-red-gwhup.eu-gb.mybluemix.net/ui>



1. **ADVANTAGES AND DISADVANTAGES**

**Advantage:**

1. Improve Customer Satisfaction

The conversations that your customers have with your business matter a lot. They help shape the opinions consumers have about your business. Customer service executives may respond differently to your customers based on their mood. And, if they’re in a bad mood, they may end up providing an unsatisfactory experience to your customers. However, a chatbot abides by the rules that you’ve set for it. This ensures that it will respond in the exact manner that you want it to at any moment. Additionally, you can configure your chatbot to speak multiple languages. This can help your brand resolve customers’ questions and problems in oher languages as well.

1. Immediateness

You have two options, to give or not to give immediate answers. Clientes prefer that you give the answers as soon as possible.Now, a chatbot will evolve over time. You may start giving automatic responses to 30% or 50% of the issues, until you reach a percentage and can exceed 80%.Sometimes, there are chatbots that guide you throughout the experience to give you an answer within the chat because it is technically possible, and others that indicate the easiest way to resolve the doubt if the previous solution is more complex.

1. Available 24/7

These conversational assistants can answer questions or questions from users anywhere and at any time.

**Disadvantages**

1. Lack Emotions

Unlike humans, a chatbot has no emotions. However, they are pretty essential to keep a conversation going the right way. Your customer service executives can understand your customers’ emotions and respond accordingly, but a chatbot may not be able to do so. Bots can be too mechanical. With pre-programmed conversations, they can handle customer requests when the flow of the conversation follows a specific path. But they may not be able to handle when the conversation takes an unexpected turn.

Since chatbots have no feelings and emotions, it can become critical for them to effectively interact with humans in some cases.

1. Poor Memory

Chatbots are not able to memorize the past conversation which forces the user to type the same thing again & again. This can be cumbersome for the customer and annoy them because of the effort required. Thus, it is important to be careful while designing chatbots and make sure that the program is able to comprehend user queries and respond accordingly.

1. Complex Interface

Chatbots are often seen to be complicated and require a lot of time to understand user’s requirement. It is also the poor processing which is not able to filter results in time that can annoy people.x

1. **APPLICATIONS**

**e-Commerce**

Chatbots can add a new layer of interactivity to e-commerce, allowing customers to interact beyond menus and buttons. Major use cases are:

* Set price alerts
* Order physical goods like clothes with conversational commerce unlock more options
* Buy gifts
* Reserve services

**Travel:**

All the way from booking travel to solving travel related problems, chatbots have the potential to help. New ventures like Instalocate are already making money by solving people’s travel related problems.

* Vacation planning
* Reservations
* Queries & complaints

1. **CONCLUSION**

The chatbot is useful to user for asking about product, hours, location etc. and gives answer based on that and provide proper response. If it is unable to answer the question then it will give response as “can you reword your sentence” or “I didn’t understand” and also gives option of talking to real agent for quarries. It gives more customer satisfaction and immediate answers also available 24/7. This chat bot is used at various places like e-commerce, travel, health care etc.

1. **FUTURE SCOPE**

To make chatbot more advance there are many features can add in it:

**Giving human-like experience**

If you are talking about personalized customer experience and offering services which are similar to the ones provided by humans, then chatbots aren’t far behind. The advancement in artificial intelligence and [machine learning](https://www.2basetechnologies.com/machine-learning-python-a-new-combo-for-futuristic-businesses) in today’s era has made chatbot services more similar to human-like and even impeccable.

**Faster problem solving**

If you really want faster clarifications for your various queries, it is easy. You can achieve that by using handheld devices like tablets, smartphones, and wearables such as smartwatches. In addition, smart TVs are becoming one of the major integrations for voice assistants like SIR, Alexa, and Google Assistant, along with other wearable and handheld devices. The best part is that users, nowadays won’t have to wait in line or stay on hold while using phone calls because of the integration of chatbots and virtual assistants.

**Better insights & consumer analytics**

For providing in-depth customer insight, one needs a system which performs the correct analysis of a huge amount of data and that too without any scope of errors. AI technology is getting incorporated for providing correct customer data, collection and automatize data analysis.

1. **BIBLIOGRAPHY**
2. <https://developer.ibm.com/patterns/enhance-customer-help-desk-with-smart-document-understandin>
3. <https://github.com/ibm/watson-discovery-sdu-with-assistant>
4. <https://developer.ibm.com/components/watson-assistant/series/learning-path-watson-assistant>
5. <https://developer.ibm.com/articles/introduction-watson-discovery/>
6. <https://www.youtube.com/watch?v=5z3i5IsBVnk>
7. <https://www.youtube.com/watch?v=Jpr3wVH3FVA>

**APPENDIX: SOURCE CODE**

1. **Cloud Function(node.js) for discovery integration webhook generation**

/\*\*

\*

\* @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);

});

});

}

1. **Node Red Flow**

[{"id":"a7f5803f.eb2a","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"f2f2649a.0d0d98","type":"debug","z":"a7f5803f.eb2a","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":930,"y":120,"wires":[]},{"id":"33b2afb8.991bf","type":"watson-conversation-v1","z":"a7f5803f.eb2a","name":"Customer Care","workspaceid":"02d314fc-7497-4e4a-a618-8c798643f58f","multiuser":false,"context":true,"empty-payload":false,"service-endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/a6ec7aac-cf8c-48bb-96fe-e96e56ea88ed/v1/workspaces/02d314fc-7497-4e4a-a618-8c798643f58f/message","timeout":"","optout-learning":false,"x":480,"y":260,"wires":[["9627b217.5ae8"]]},{"id":"29d5e528.c2294a","type":"function","z":"a7f5803f.eb2a","name":"Input\_function","func":"msg.payload = msg.payload.text;\nreturn msg;","outputs":1,"noerr":0,"x":220,"y":260,"wires":[["33b2afb8.991bf","23b31ea6.2638d2"]]},{"id":"5ead9275.60f56c","type":"ui\_form","z":"a7f5803f.eb2a","name":"","label":"","group":"f4fb2e27.51cff","order":6,"width":0,"height":0,"options":[{"label":"Intelligent customer help desk","value":"text","type":"text","required":true,"rows":null}],"formValue":{"text":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":70,"y":380,"wires":[["29d5e528.c2294a"]]},{"id":"9627b217.5ae8","type":"function","z":"a7f5803f.eb2a","name":"Output\_function","func":"msg.payload=msg.payload.output.text[0];\nreturn msg;","outputs":1,"noerr":0,"x":740,"y":260,"wires":[["1f32df.4d26ed22","f2f2649a.0d0d98"]]},{"id":"1f32df.4d26ed22","type":"ui\_text","z":"a7f5803f.eb2a","group":"f4fb2e27.51cff","order":3,"width":"0","height":"0","name":"","label":"Result","format":"{{msg.payload}}","layout":"col-center","x":910,"y":360,"wires":[]},{"id":"23b31ea6.2638d2","type":"ui\_text","z":"a7f5803f.eb2a","group":"f4fb2e27.51cff","order":1,"width":0,"height":0,"name":"","label":"Input","format":"{{msg.payload}}","layout":"col-center","x":370,"y":380,"wires":[]},{"id":"886c2c99.9446b","type":"inject","z":"a7f5803f.eb2a","name":"","topic":"","payload":"Hello","payloadType":"str","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":230,"y":140,"wires":[["33b2afb8.991bf"]]},{"id":"f4fb2e27.51cff","type":"ui\_group","z":"","name":"Customer Help Desk","tab":"2f5b50f.1d5f9b","order":1,"disp":true,"width":"8","collapse":false},{"id":"2f5b50f.1d5f9b","type":"ui\_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]