

# IBM Watson Discovery

## Watson Assistant Integration with Webhook (Cloud Function/Node.js)

### Cognitive Solutions Application Development

Updated: Apr 27, 2020  
Klaus-Peter Schlotter  
[kps@de.ibm.com](mailto:kps@de.ibm.com)



Version 5

Watson Library V5

## Table of Contents

Overview.....	3
Objectives.....	3
Prerequisites.....	3
Section 1: Create a Discovery service instance.....	4
Section 2 Integrate Discovery into a Chatbot (Webhook).....	10
Create a Watson Discovery Document Collection.....	10
Update the Assistant Skill to integrate the Discovery Collection.....	12
Add the Discovery Service call to the Cloud Function.....	15
Section 4 Integrate Discovery into a Chatbot (Search Skill).....	17
Create the Search Skill.....	17
Connect the Search Skill to a place in the dialog.....	19

## Overview

The [IBM Watson Developer Cloud](#) (WDC) offers a variety of services for developing cognitive applications. Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service. Some services, such as the *Speech to Text* service, provide additional interfaces.

The Watson Discovery service adds a cognitive search and content analytics engine to applications to identify patterns, trends and actionable insights that drive better decision-making. Securely unify structured and unstructured data with pre-enriched content, and use a simplified query language to eliminate the need for manual filtering of results.

The app created in this lab can run locally or in the IBM Cloud.

- In a first step, a Watson Discovery service instance is created on the IBM Cloud.
- Next, you will adjust the IBM Cloud demo to use the German document collections
- Then the chatbot from lab 3 will be enhanced with Discovery capabilities.

## Objectives

- Learn how to use the Watson Discovery service on IBM Cloud
- Learn how to integrate with Watson Assistant using the Search Skill feature.

## Prerequisites

Before you start the exercises in this guide, you will need to complete the following prerequisite tasks (see the [setup guide](#)).

- Create a IBM Cloud account

## Section 1: Create a Discovery service instance

**Step 1** In a web browser, navigate to the following URL

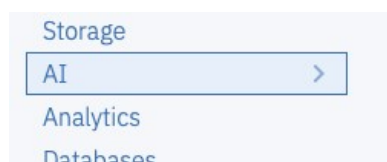
<https://cloud.ibm.com>

**Step 2** Log in with your IBM Cloud credentials. This should be your IBMid.

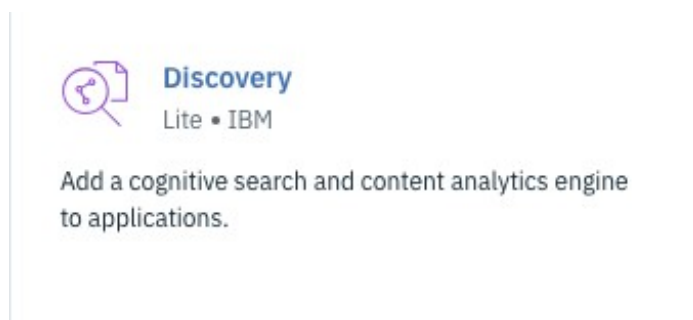
**Step 3** You should start on your dashboard which shows a list of your applications and services. Scroll down to the All Services section and click **Create Resource**.

Create resource


**Step 4** On the left, under Services, **click** on *AI* to filter the list and only show cognitive services.



**Step 5** **Click** on the Watson Discovery service.

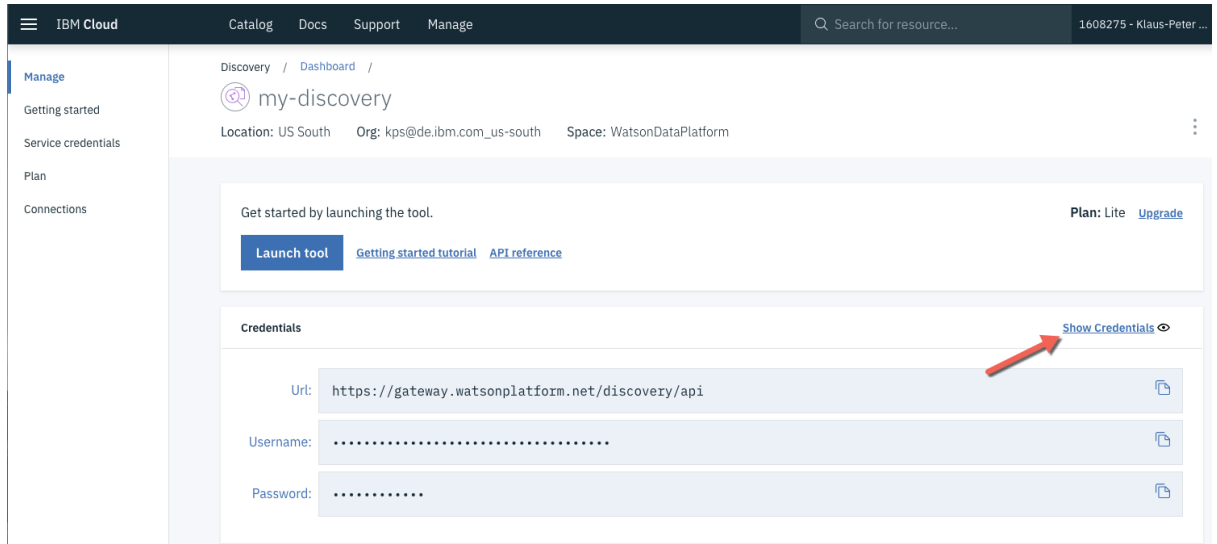



**Step 6** Review the details for this service. At the top, there will be a description of the service. At the bottom, you can review the pricing plans. The Lite plan for this service provides no cost monthly allowances for workspaces, intents, and API calls. Enjoy your demo!

**Step 7** At the top, you can enter information for your new service. In the middle, you can click on a pricing plan to select it. Fill out the fields as follows, then **click**  at the bottom.

Field	Value
Service name	my-discovery
Selected Plan	Lite

**Step 8** IBM Cloud has created a new service instance.



**Step 9** In the *Credentials* section **click** [Show Credentials](#) . You should see the user-name and password for your service. Later in this exercise, you will enter these values into a JSON configuration file for your Node.js application. Feel free to copy them to your clipboard, to a text file, or just return to this section of the IBM Cloud web interface when the credentials are needed.

**Step 10** Click the [Launch tool](#) button to open the Discovery service configuration tool.

**Step 11** **Select** your language and **click** on the *Watson Discovery News* collection, which is available in every service instance, to open it.



This collection is provided by IBM and gets updated daily, in German with approximately 40000 news articles per day containing a 60 day period of news (~ 3 Million entries).

On the overview page of the document collection you can see infos such as *Document count*, *Collection Info*, and *Insights from enriched data*. This enriched data is added by Watson when the documents get ingested.

IBM Watson Discovery

Cookie Preferences Instance: my-discovery

Watson Discovery News (German)

755,361 documents

### About Watson Discovery News

Watson Discovery News is updated continuously with new articles.

Language	New articles per day
English	300,000
Spanish	60,000
German	40,000
Japanese	17,000
Korean	10,000

The news sources vary by language, so the query results for each collection will not be identical.

### Added 5 enrichments to your data

#### Entity Extraction

Deutschland (70k) | USA (50k) | Schweiz (41k) | Europa (35k) | Der (35k)

#### Sentiment Analysis

44% positive | 12% neutral | 45% negative

#### Concept Tagging

Deutschland (110k) | Euro (86k) | Vereinigte Staaten (84k) | Schweiz (75k) | Römischer Kalender (65k)

#### Keyword Extraction

August (60k) | Jahren (53k) | Jahr (50k) | September (41k) | Bild (39k)

### Now you're ready to query!

- Company acquisitions relating to artificial intelligence [Run](#)
- Top 20 recently acquired artificial intelligence companies [Run](#)
- Top 10 companies with positive sentiment [Run](#)
- 50 most mentioned people in the Tech industry [Run](#)
- Sentiment of articles about IBM over the last 60 days [Run](#)
- News from Business Wire about Elon Musk [Run](#)

The *Use this collection in API* information is needed to access this information from your application, *Collection Id* and *Environment Id*.

Cookie Preferences Instance: my-discovery

API

Collection ID

news-de

Environment ID

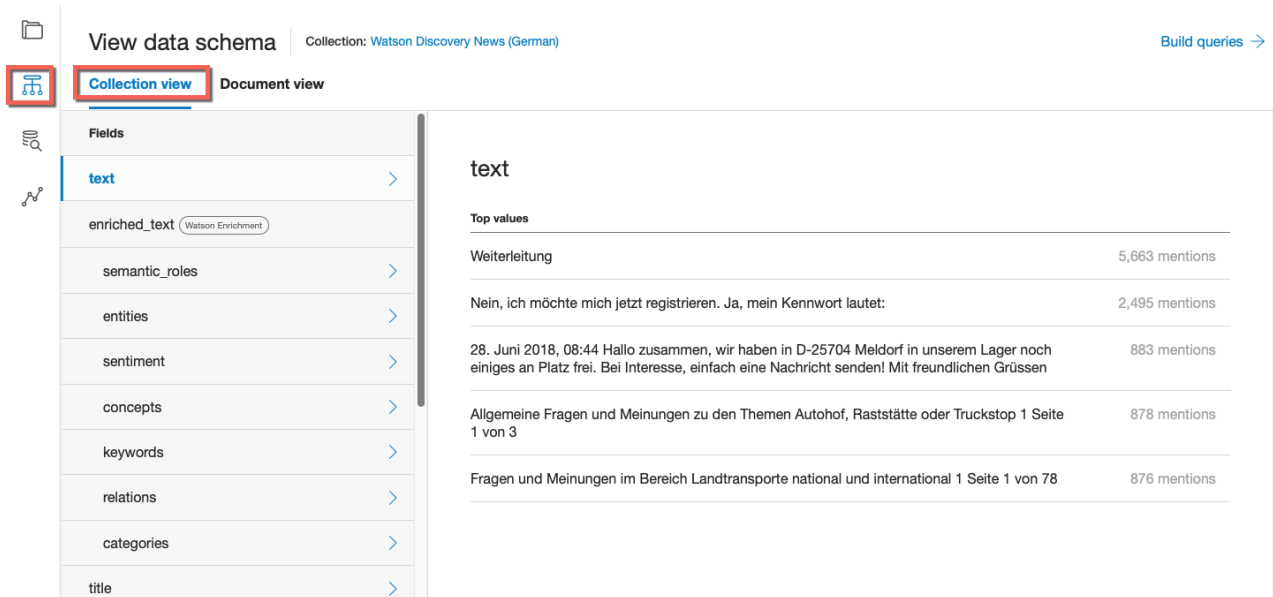
system

Now you're ready to query!

Company acquisitions relating to artificial intelligence

**Step 12** On the *View data schema* page you can look at your collection in a

- a) *Collections View* which shows you a summary of the Watson Enrichments on the collection scope



View data schema | Collection: Watson Discovery News (German) | Build queries →

**Collection view** | Document view

**Fields**

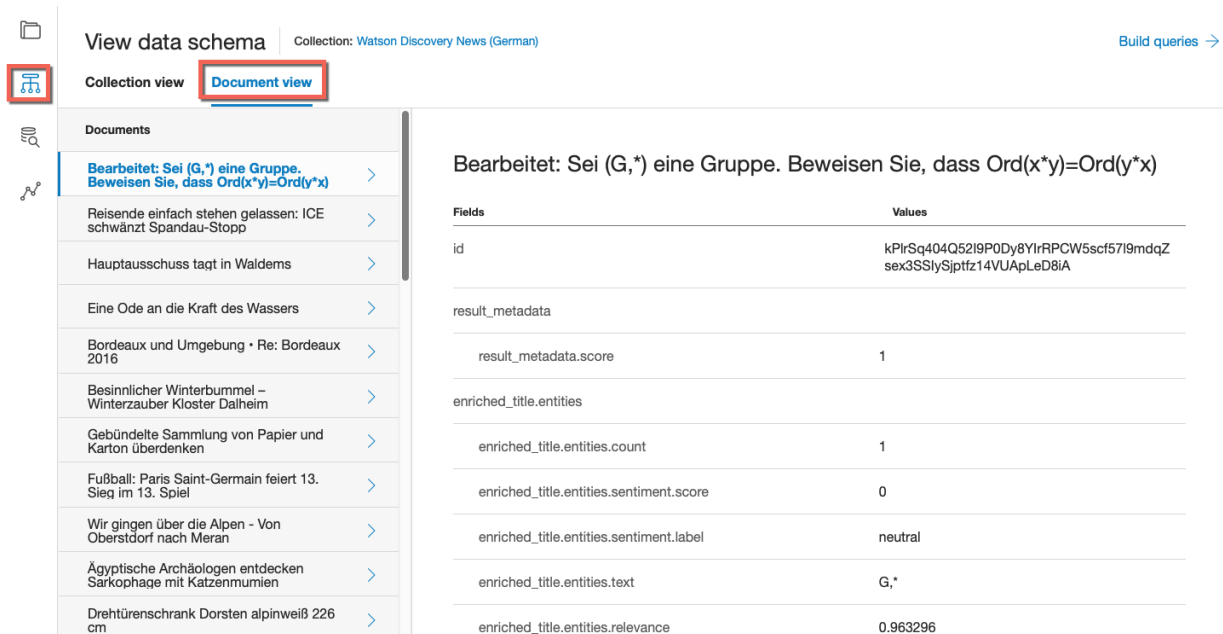
- text
- enriched\_text (Watson Enrichment)
- semantic\_roles
- entities
- sentiment
- concepts
- keywords
- relations
- categories
- title

**text**

Top values

Value	Mentions
Weiterleitung	5,663 mentions
Nein, ich möchte mich jetzt registrieren. Ja, mein Kennwort lautet:	2,495 mentions
28. Juni 2018, 08:44 Hallo zusammen, wir haben in D-25704 Meldorf in unserem Lager noch einiges an Platz frei. Bei Interesse, einfach eine Nachricht senden! Mit freundlichen Grüßen	883 mentions
Allgemeine Fragen und Meinungen zu den Themen Autohof, Raststätte oder Truckstop 1 Seite 1 von 3	878 mentions
Fragen und Meinungen im Bereich Landtransporte national und international 1 Seite 1 von 78	876 mentions

- b) *Document View* page which shows you Watson Enrichments on a document level.



View data schema | Collection: Watson Discovery News (German) | Build queries →

Collection view | **Document view**

**Documents**

- Bearbeitet: Sei (G,\*) eine Gruppe. Beweisen Sie, dass Ord(x\*y)=Ord(y\*x)
- Reisende einfach stehen gelassen: ICE schwänzt Spandau-Stopp
- Hauptausschuss tagt in Waldems
- Eine Ode an die Kraft des Wassers
- Bordeaux und Umgebung • Re: Bordeaux 2016
- Besinnlicher Winterbummel – Winterzauber Kloster Dalheim
- Gebündelte Sammlung von Papier und Karton überdenken
- Fußball: Paris Saint-Germain feiert 13. Sieg im 13. Spiel
- Wir gingen über die Alpen - Von Oberstdorf nach Meran
- Ägyptische Archäologen entdecken Sarkophag mit Katzenmumien
- Drehbüchschrank Dorsten alpinweiß 226 cm

**Bearbeitet: Sei (G,\*) eine Gruppe. Beweisen Sie, dass Ord(x\*y)=Ord(y\*x)**

Fields	Values
id	kPirSq404Q52I9POdy8YlrPCW5scf57I9mdqZsex3SSlySjptfz14VUApLeD8iA
result_metadata	
result_metadata.score	1
enriched_title.entities	
enriched_title.entities.count	1
enriched_title.entities.sentiment.score	0
enriched_title.entities.sentiment.label	neutral
enriched_title.entities.text	G,*
enriched_title.entities.relevance	0.963296

See the documentation of the service for more information.

**Step 13** On the *Build queries* page you can search your collection f.e. *Use natural language* and include an analysis of the Watson Enrichment.

**Step 14** Click the JSON tab at the top right then you see the result that you would get when you call this from the API.

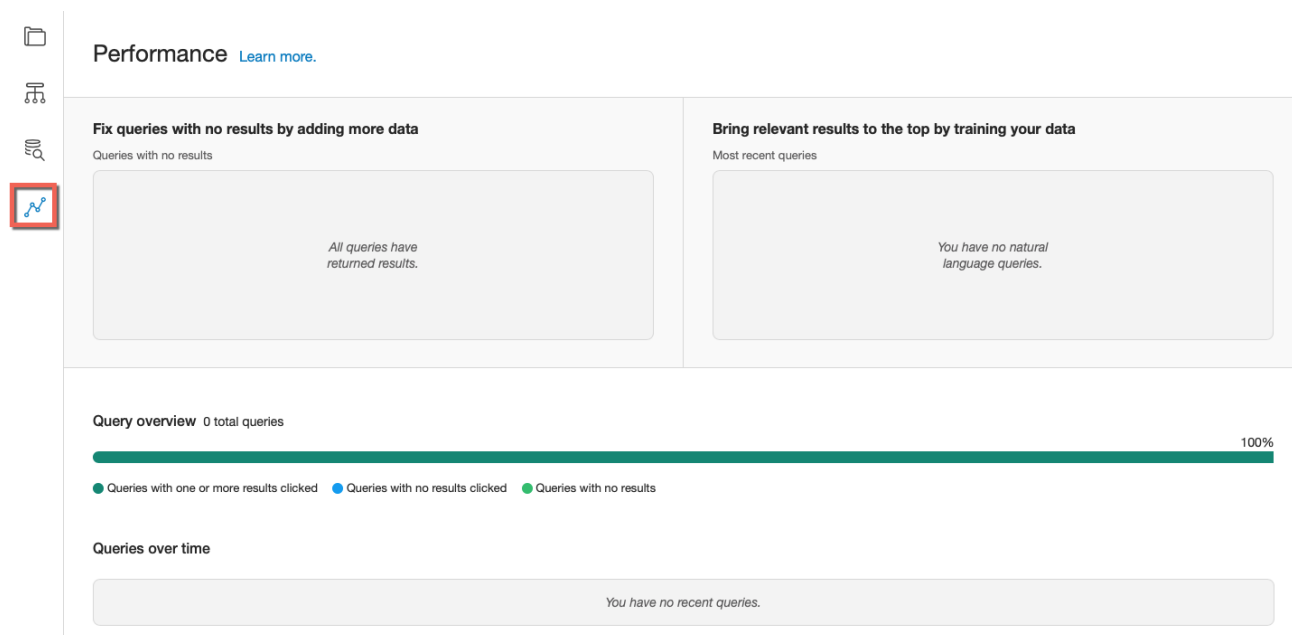
Here is a nice video of the tutorial listed at the end of this document on how to work with the Discovery enriched-data to build powerful queries using this IBM News collection.

<https://www.youtube.com/watch?v=N-HalPGde0&feature=youtu.be>

More information can be found in the documentation.



**Step 15** On the *Performance* page you can see how your collection performs over time. Right now, there is no data available.



## Section 2 Integrate Discovery into a Chatbot (Webhook)

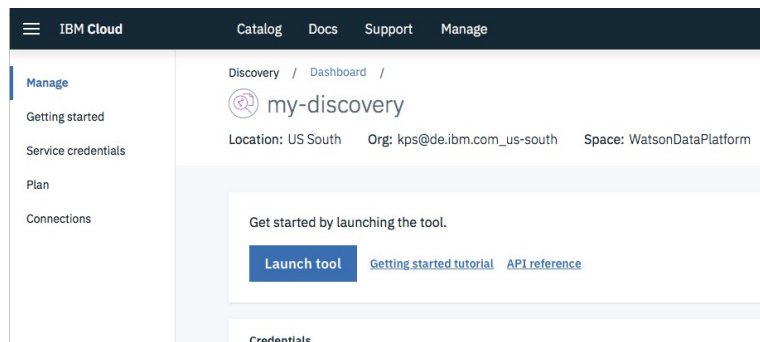
In this section we will integrate the Discovery Service into the Chatbot created in Lab 3a.

**Step 16** As a starting point for this section you should follow Lab 3a Section 1. This section will list shortcuts on how to get started quickly such as importing the Conversation workspace or cloning the app source code.

### Create a Watson Discovery Document Collection

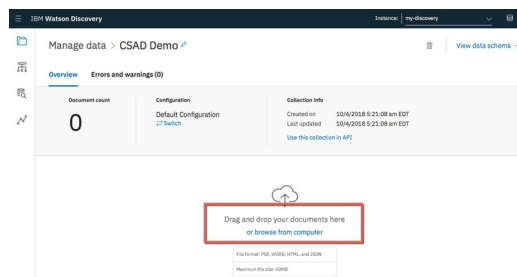
**Step 17** Download the zip file manualdocs.zip from [github](#) (click the [Download](#) button) and extract it into a folder. It's a web manual for a Ford car.

**Step 18** In your cloud console dashboard click on the “my-discovery” service created above and click the [Launch tool](#) button.

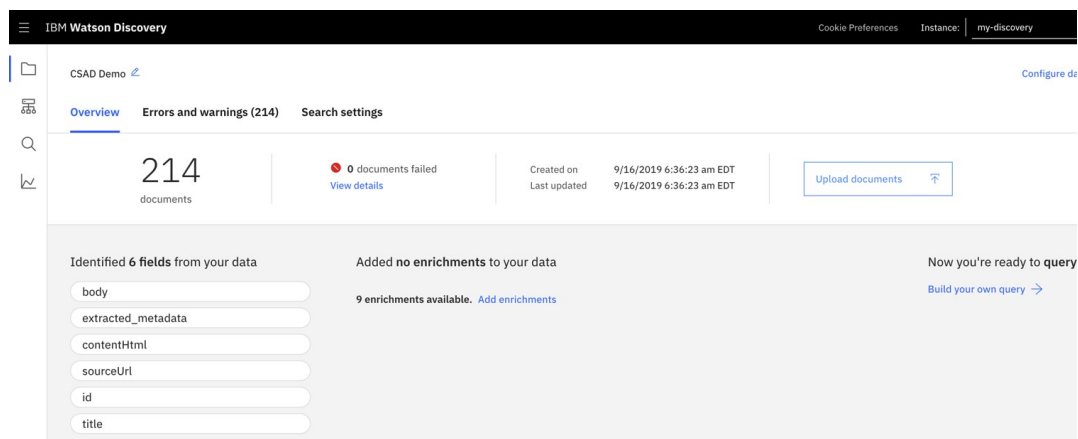


**Step 19** On the *Manage data* page click [Upload your own data](#). As collection Name enter *CSAD Demo* and click [Create](#).

**Step 20** The new collection opens and you can either **browse** or **drag ‘n’ drop** all the documents extracted above (234 json files).

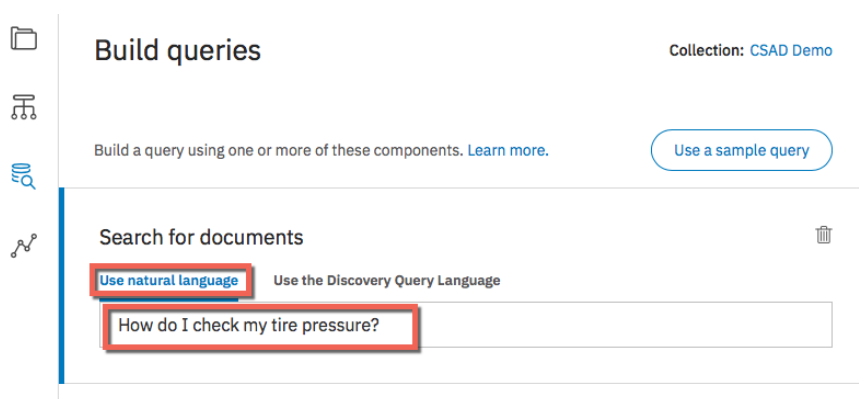


**Step 21** The Discovery Service now indexes the documents. This may take a few minutes.



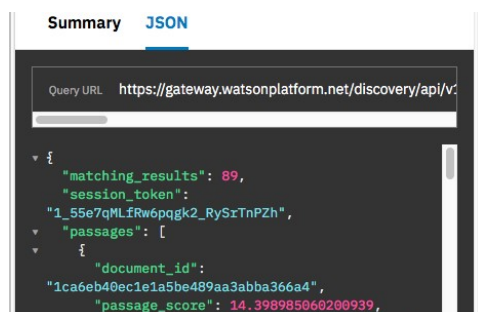
**Step 22** On the left, click the  button to open the *Build queries* page.

**Step 23** Enter the following query: How do I check my tire pressure?




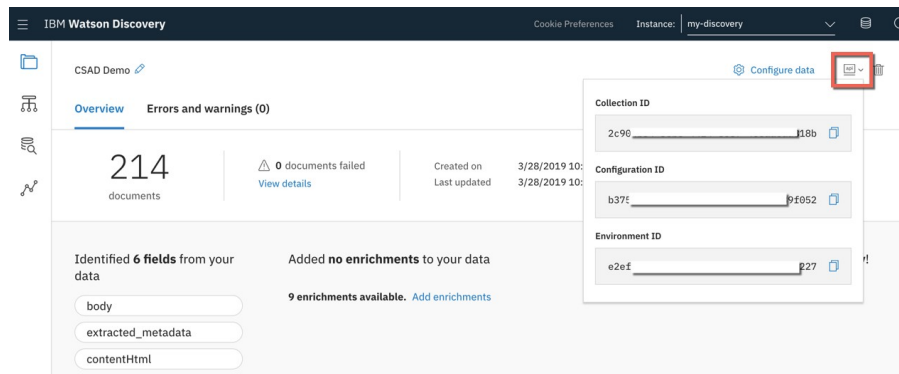
and press the  button.

On the left you can then see the results. Especially switch between the Summary and the JSON view of the results.



There is a lot more to explore here, but this is out of scope for this short tutorial. We just want to use the question submitted above later in our extended chatbot.

**Step 24** On the top right of the *Overview* page **click**  to show the ids you have to copy for later use by the application in addition to the service credentials.




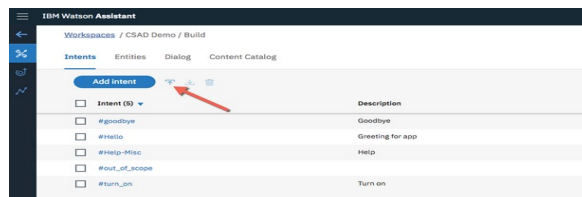
## Update the Assistant Skill to integrate the Discovery Collection

**Step 25** Download the `out_of_scope_intents.csv` from the [Github repository](#) to your project folder. Use the following in the terminal:


```
wget https://github.com/iic-dach/csadConversation/blob/master/resources/Out_of_scope_intents.csv
```

**Step 26** Go back to your Watson Assistant console that should still be open in the browser (Step 11)


**Create** a new intent `out_of_scope` and return to the list of intents. Click the  button to import the user examples from the file downloaded above.




131 examples will be imported.

**Step 27** On the *Dialog* tab **click** the menu  button on the *Turn on* node and **click** *Add node below*. **Name it** `Out of Scope`. In the field *If bot recognizes type* `#out_of_scope`.

**Note:** Due to modularity we create the node in this location. Because this is also some kind of help question, it might be also appropriate, to insert this in the child tree of the *help* node.

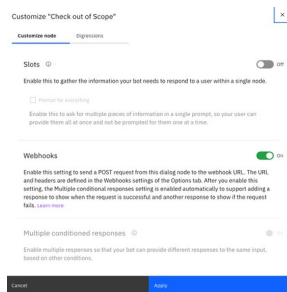
**Step 28** Now **click** the menu  button on the `#out_of_scope` node. And **click** *Add child node*.

**Step 29** In this child node, **name** it *Check out of Scope*, in **If bot recognizes**, just **type true**.

**Step 30** Now **click** the menu  button on the *#out\_of\_scope* node. And **click Jump to (Recognizes Condition)**.

**Step 31** Click the *Check out of Scope* node.

**Step 32** Click  **Customize** and **enable Webhooks**. Then click 



The webhook was already customized in Lab3a.

**Step 33** Now you can add parameters for the webhook callout.

Add the following:

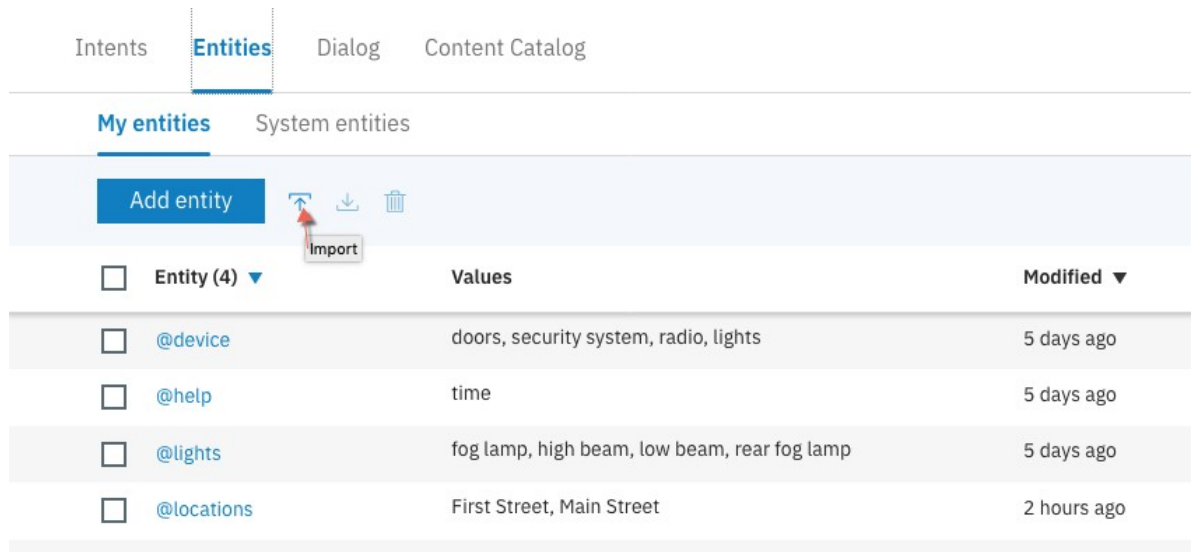
Key	Value
action	discovery
query	<? input.text ?>

**Step 34** Add the following in Assistant responds. Complex responses can be easily built on the customize screen .

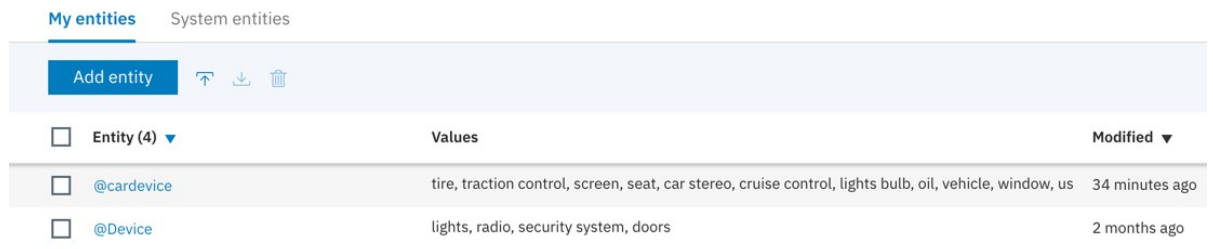
If assistant recognizes	Respond with
\$webhook_result_2 && @cardevice	<? \$webhook_result_2.message ?>
true	Sorry I haven't learned answers to questions like this.

**Step 35** In your Skill's Entities add an Entity by importing the cardevice\_entity.csv CSV file.

```
wget https://github.com/iic-dach/csadConversation/blob/master/resources/cardevice_entity.csv
```



This will add an Entity @cardevice



**Step 36** Now change the *Check out of Scope* node (Step 48) to the following

If bot recognizes	Respond with
@cardevice	That question is out of scope for this application, take a look at the Conversation Enhanced application to handle questions like these.
true	Sorry I haven't learned answers to questions like this.

“true” always executes when the above condition is not met.

**Step 37** The backend is now adjusted for integrating the Discovery service into our chatbot application.

## Add the Discovery Service call to the Cloud Function

In Lab 3a we created a Webhook/Cloud Function to provide the system time for the chatbot. The following code shows the Cloud Function with the added Watson Discovery Service integration. **Copy** and **Paste** from [here](#).

**Step 38** Update/replace the Cloud Function with the following code

```
const axios = require("axios");
const discoveryUrl =
  "https://gateway.watsonplatform.net/discovery/api/v1/environments/{your environmentId}/collectionsId/{your collection}/query?version=2019-04-30";
const auth = "Basic " + new Buffer.from("apikey" + ":" +
  "{your_apikey}").toString("base64");

async function main(params) {
  if (params.action === "gettime") {
    const deTime = new Date().toLocaleString("de-DE", {timeZone: "Europe/Berlin"});
    return {
      statusCode: 200,
      headers: {
        Content_Type: "application/json",
      },
      body: { message: deTime },
    };
  } else if (params.action === "discovery") {
    try {
      let response = await axios({
        method: "post",
        url: discoveryUrl,
        data: {
          natural_language_query: params.query,
          passages: true,
          "passages.count": 1,
        },
        headers: {
          Authorization: auth,
          "Content-Type": "application/json"
        },
      });
      return {
        statusCode: response.status,
        headers: {
          "Content-Type": "application/json",
          Accept: "application/json",
        },
        body: { message: response.data.passages[0].passage_text }
      };
    } catch (err) {
      return { body: {message: err.message} };
    }
  } else {
    return {
      statusCode: 404,
      headers: { "Content-Type": "application/json" },
      body: { message: `action ${params.action} not defined` }
    };
  }
}
```

Some constants were added to get the **axios** Promised base HTTP client, to define the **Watson Discovery collection url** and the to create a **Basic Auth token** from username (the string “apikey”) and as password the services apikey.

Because getting the data from the Discovery Service is an asynchronous function, **function main(...)** has to be preceded with the **async** keyword. This corresponds with the **await** keyword in the axios command.

### Step 39 Test your Assistant with Try it out

### Step 40 Enter some statements into the Assistant input line:

Switch on the lights

The headlights

What time is it?

Where can I find my wallet?

How do I check my tire pressure?

#### IBM EAG Watson Assistant Lab

Send

**Conversation History:**  

Watson: Welcome to CSAD Demo!  
You: Switch on the lights  
Watson: OK! Which light would you like to turn on?  
You: The headlights  
Watson: OK! Turning on low beam lights.  
You: What time is it?  
Watson: The current time is 16:53:58  
You: Where can I find my wallet?  
Watson: Sorry I haven't learned answers to questions like this.  
You: How do I check my tire pressure?  
Watson:  
You can check the tire pressure any time within the 200 km by performing the procedure from Second stage: Checking tire pressure listed previously.  
Be sure to check the sealant compound's use by date regularly.

IBM Ecosystem Advocacy Group - 2018

The question for the tire pressure should return the same as in **Step 23**, the first passage.



## Section 4 Integrate Discovery into a Chatbot (Search Skill)

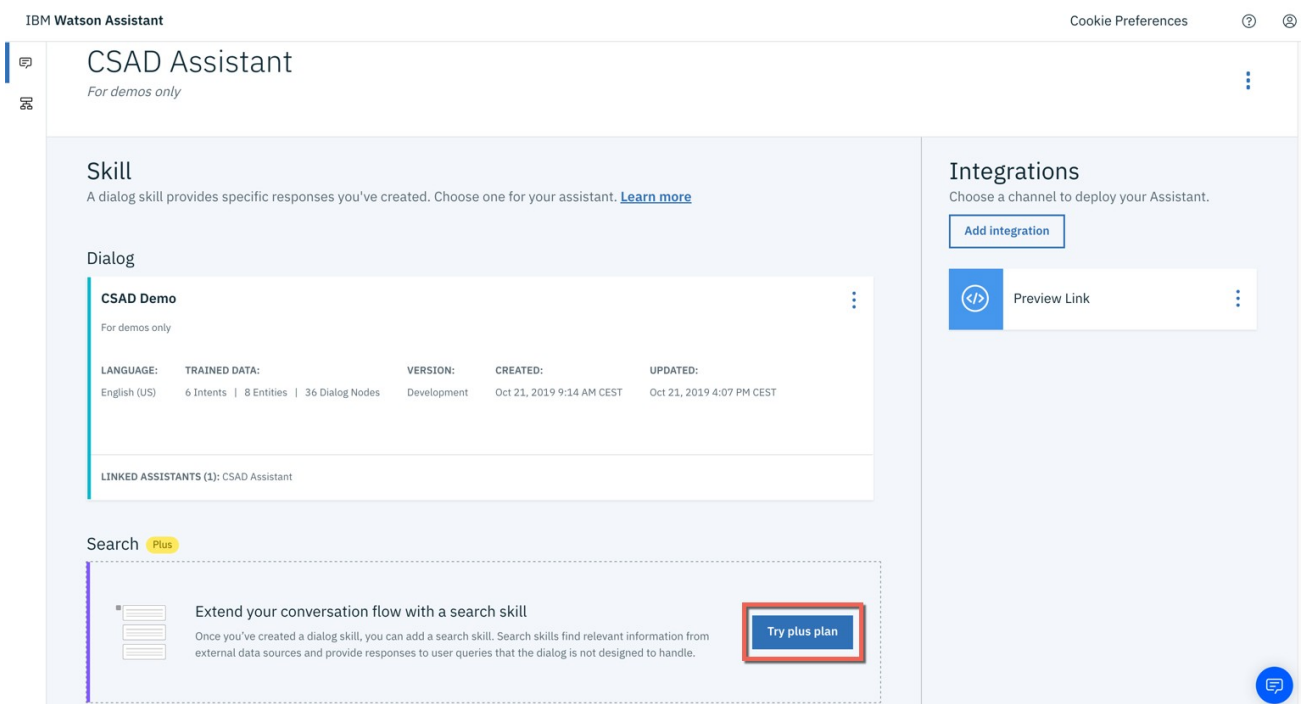
In the previous section we have seen in how to integrate the Watson Discovery service into a chatbot using the API.

In the Watson Assistant service there is now a *Search Skill* option available that allows to directly integrate a Discovery document collection into a chatbot.

To use this search skill we have to upgrade our Discovery service instance to the **Plus Trial** plan.

### Create the Search Skill

**Step 41** In the browser **open** your *Watson Assistant* console and **click** [Try plus plan](#)



**Step 42** Then **click** [Start free trial](#) and **click** [Agree](#) to start the 30 day trial period.

**Step 43** After the upgrade **click** [Add search skill](#).

**Step 44** **Name** it f.e. *CSAD Search* and **click** *Continue*. Your previously created Discovery service will be displayed with the already defined collections.

**Step 45** **Click** [Create new collection](#). Then **click** [Let's get started](#). You cannot have more than two collection is your lite service.

**Step 46** **Click** [Upload documents](#). You can find these documents [here](#) or in the resources folder of the Github repository. **Name** your collection *CSAD Search*. Now upload the three PDF documents. **This may take a while!!**.

**Step 47** Once the documents have been ingested, **click**  
*Finish setup in Watson Assistant*

**Step 48** Make sure the new collection is selected and **click** *Configure*.

For **Title** select the field: *extracted\_metadata.filename*  
For **Body**, select the field: *text*.  
For **url**, leave empty.

Then **click** *Create*.

## Connect the Search Skill to a place in the dialog

**Step 49** Open the *Intents* section of your Assistant skill

**Step 50** Click [Create intent](#) enter the following values and click [Create intent](#)

Field	Value
Intent name	Credit_Card_Info
Description	Search Skill integration

**Step 51** The user examples are phrases that will help Watson recognize the new intent. (Enter multiple examples by **pressing** “Enter” or by **clicking** the *Add example*). When finished, click [←](#) at the top of the page.

Field	Value
User example	are there any international fees if i use my card in paris atm fees for my credit card fees for using my card abroad what are the apr fees for my silver credit card What are the fees for using an ATM in another country?

**Step 52** Open the *Entities* section of you Assistant skill

**Step 53** Click [Create entity](#), enter `card_info` and click [Create entity](#)

**Step 54** Click [Add value](#). When finished, click [←](#) at the top of the page.

Field	Value	Synonym
Entity name	card_info	
Value	APR	annual percentage rate
	cash advance	advanced cash, money advance
	international fees	International charges, foreign fees, foreign charges, fees while abroad, foreign exchange
	annual fee	yearly fee, annual membership, yearly membership, yearly cost, entry fee

**Step 55** Open the *Dialog* section of your Assistant skill

**Step 56** Click the menu [⋮](#) on the *Book a Table* node and then click *Add node below*, with the following values.

Field	Value
Name this node...	Information for a Card
If bot recognizes:	#Credit_Card_Info

**Step 57** Click [Customize](#) and enable *Multiple responses*. The click [Apply](#)

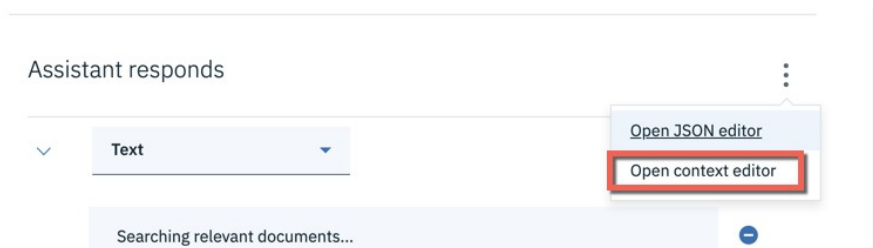
**Step 58** Enter the following responses:

If Assistant recognizes	Respond with
@card_info:(annual fee)	Searching relevant documents...
@card_info:(international fees)	Searching relevant documents...
@card_info:(APR)	Searching relevant documents...
@card_info:(cash advance)	Searching relevant documents...
anything_else	Searching relevant documents...

**Step 59** Now customize the response for response 1, click 

IF ASSISTANT RECOGNIZES	RESPOND WITH
1 @card_info:(annual fee)	Searching relevant documents...

**Step 60** Open the Context Editor



**Step 61** Add two context variables

Variable	Value
actions	"WDS_SEARCH"
passages	{"fields":"annual_membership"}

Configure response 1

If assistant recognizes

@card\_info:(annual fee) 

Then set context

VARIABLE	VALUE
\$ actions	"WDS_SEARCH"
\$ passages	{"fields":"annual_membership"}

**Step 62** Change the *Assistant responds* option to *Search skill*. Click Save

Configure response 1

\$ passages {"fields": "annual\_membership"}

Add variable +

Assistant responds

Search skill

[Customize](#) how your search skill is called. [Learn more](#)

Add response type +

**Step 63** Do the same for the second response, but for passages enter the following:

Variable	Value
actions	"WDS_SEARCH"
passages	{"fields": "international_fees"}

**Step 64** Then click ✕ to close the dialog.

**Step 65** All other responses just return “*Searching relevant documents...*”

**Step 66** Open your Assistant and click the *Preview Link*. Click the URL shown to open the Assistant in the browser.

IBM Watson Assistant

CSAD Demo

Skills

Dialog

CSAD Demo

LANGUAGE: English (US) TRAINED DATA: 7 Intents | 9 Entities | 42 Dialog Nodes VERSION: Development CREATED: May 17, 2019 8:23 AM CEST UPDATED: Oct 22, 2019 7:08 PM CEST

LINKED ASSISTANTS (1): CSAD Demo

Search Plus

CSAD Search

Integrations

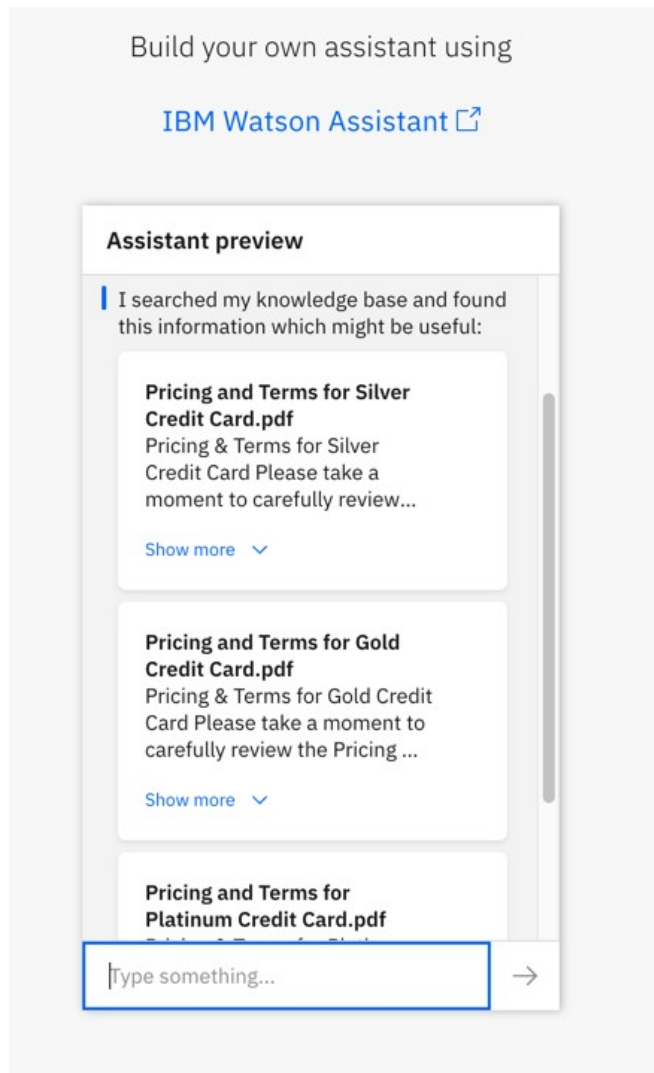
Choose a channel to deploy your Assistant.

Add integration

Preview Link

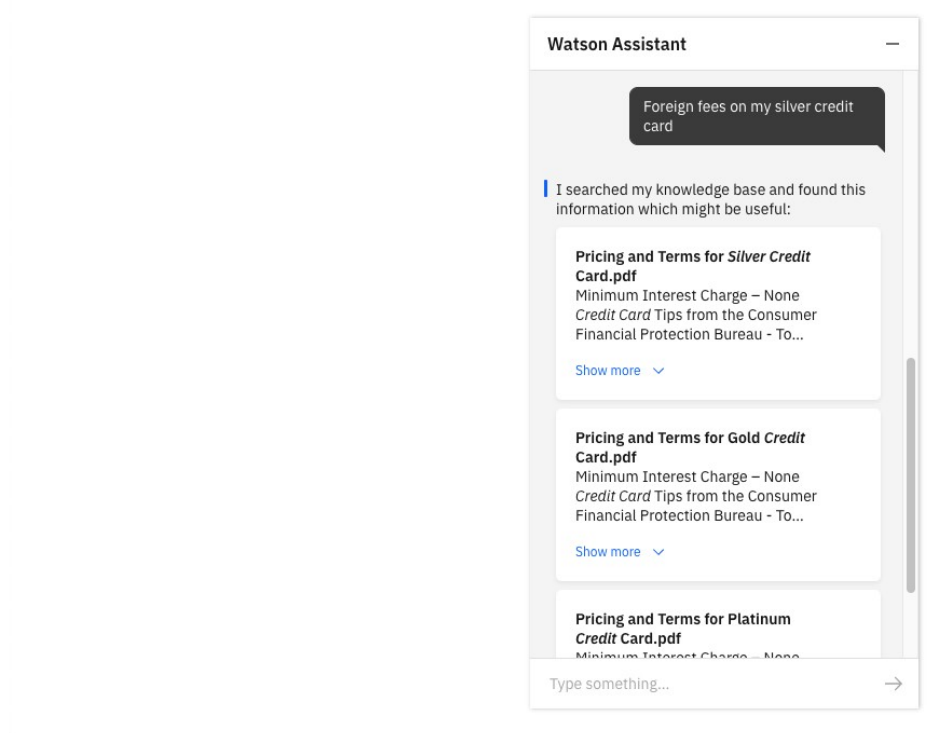
**Step 67** Enter the following into your Assistant preview:

Foreign fees on my silver credit card



**Step 68** In Lab 3a you have integrated the chat bot into your web site using the Webchat feature of Watson Assistant. The page should no look like the following when “Foreign fees on my credit card” is entered.

### Home Page to integrate the Chatbot



You have successfully completed this lab.