

Building AI based Banking assistant

Using Watson Conversation, Discovery and IBM Cloud Functions

Part 1

Configuring Watson Conversation -

Building intents, entities & dialogs, creating slots

Log into IBM Cloud (<https://console.bluemix.net/>) , click on the Catalog menu.
Click on Watson on bottom left and click on Conversation as highlighted below

The screenshot displays the IBM Cloud Catalog interface. On the left, a sidebar lists various categories: Infrastructure, Platform, and Security. The 'Watson' category is highlighted with a red circle. The main area shows a grid of services. The 'Conversation' service is highlighted with a red circle. The top navigation bar includes links for Catalog, Docs, Support, and Manage. A search bar and a filter button are located at the top right of the main content area.

IBM Cloud

Catalog Docs Support Manage

All Categories

Infrastructure

Compute

Storage

Network

Security

Containers

VMware

Platform

Boilerplates

APIs

Application Services

Blockchain

Cloud Foundry Apps

Data & Analytics

DevOps

Finance

Functions

Integrate

Internet of Things

Mobile

Network

Security

Watson >

Search Filter

Build cognitive apps that help enhance, scale, and accelerate human expertise.

Conversation

Add a natural language interface to your application to automate interactions with your users.

Lite IBM

Discovery

Add a cognitive search and content analytics engine to applications.

Lite IBM

Knowledge Studio

Build custom models to teach Watson the language of your domain.

IBM

Language Translator

Translate text from one language to another for specific domains.

Lite IBM

Natural Language Classifier

Natural Language Classifier performs natural language classification on question texts. A user-defined set of categories is used to classify the input text.

IBM

Natural Language Understanding

Analyze text to extract meta-data from content such as concepts, entities, emotion, relations, and more.

Lite IBM

Personality Insights

The Watson Personality Insights derives insights from transactional and social media data to identify personality traits.

Lite IBM

Speech to Text

Low-latency, streaming transcription

Lite IBM

Text to Speech

Synthesizes natural-sounding speech from text.

Lite IBM

Tone Analyzer

Tone Analyzer uses linguistic analysis to detect three types of tones from communications: positive, neutral, and negative.

Lite IBM

Visual Recognition

Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose from pre-trained models or create custom models.

Lite IBM

Give a service name for Conversation, select US South as region, choose your organization and space.
Click on Create

IBM Cloud

Catalog

Docs

Support

Manage

← View all

Conversation

Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device. Train Watson Conversation service through an easy-to-use web application, designed so you can quickly build natural conversation flows between your apps and users, and deploy scalable, cost effective solutions.

Lite

IBM

View Docs

AUTHOR

IBM

PUBLISHED

12/12/2017

TYPE

Service

LOCATION

Sydney, Germany, United Kingdom, US South

Service name:

Conversation-Codeday

Choose a region/location to deploy in:

US South

Choose an organization:

saurav.raiguru@in.ibm.com

Choose a space:

ourdev

Images

Click an image to enlarge and view screen captures, slides, or videos. Screen caps show the user interface for the service after it has been provisioned.

Car Dashboard Sample

Try it out

1. Create a new workspace

2. Add a new channel

3. Add a new intent

4. Add a new entity

5. Add a new dialog

1. Create a new workspace

2. Add a new channel

3. Add a new intent

4. Add a new entity

5. Add a new dialog

Building conversation bot

Workspaces

1. Create a new workspace

2. Add a new channel

3. Add a new intent

4. Add a new entity

5. Add a new dialog

1. Create a new workspace

2. Add a new channel

3. Add a new intent

4. Add a new entity

5. Add a new dialog

Need Help?
[Contact IBM Cloud Sales](#)

Estimate Monthly Cost
[Cost Calculator](#)

Create

The service will get created. Click on Dashboard menu, you will find the newly created service. Click on the service as shown below.

Dashboard

RESOURCE GROUP
All Resources ▾

REGION
US South ▾

CLOUD FOUNDRY ORG
saurav.raiguru@in.ibm.com ▾

CLOUD FOUNDRY SPACE
ourdev ▾

Filter by resource name...

Create resource

Cloud Foundry Apps 1.625 GB/8 GB Used

Name ▲	Route	Memory (MB)	State
TalentManager-20170711054135292	talentmanager-20170711054135...	512	● Stopped (0/1) ⋮
watson-dojo-pm-tester-20170928063524450	watson-dojo-pm-tester-201709...	64	● Stopped (0/1) ⋮
watsonbankingchatbot	watsonbankingchatbot-crypto...	256	● Stopped (0/1) ⋮

Cloud Foundry Services 30/40 Used

Name ▲	Service Offering	Plan
Conversation-Codeday	Conversation	Lite
DSX-ObjectStorage	Object Storage	Lite
DSX-Spark	Apache Spark	Lite

Click on Launch tool button

Manage

Service credentials

Plan

Connections

Watson /



Conversation-Codeday

Location: US South


Org: saurav.raiguru@in.ibm.com

Space: ourdev



Conversation

Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device.

Launch tool 

Developer resources:

- [Documentation](#)
- [Demo](#)

Login with IBM ID.



Watson Conversation

Log in with IBM ID



[Sign up for IBM ID](#)

The workspace page will open. Click on Import button.

Watson Conversation



Workspaces

Create  

watson-banking-chatbot

A cognitive banking chatbot

English (U.S.)

Last modified: 6 days ago [Upgrade](#)

Car Dashboard - Sample

Cognitive Car Dashboard sample workspace which allows multi-turn conversations to perform tasks in the car.


English (U.S.)


[Edit sample](#)

Create a new workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

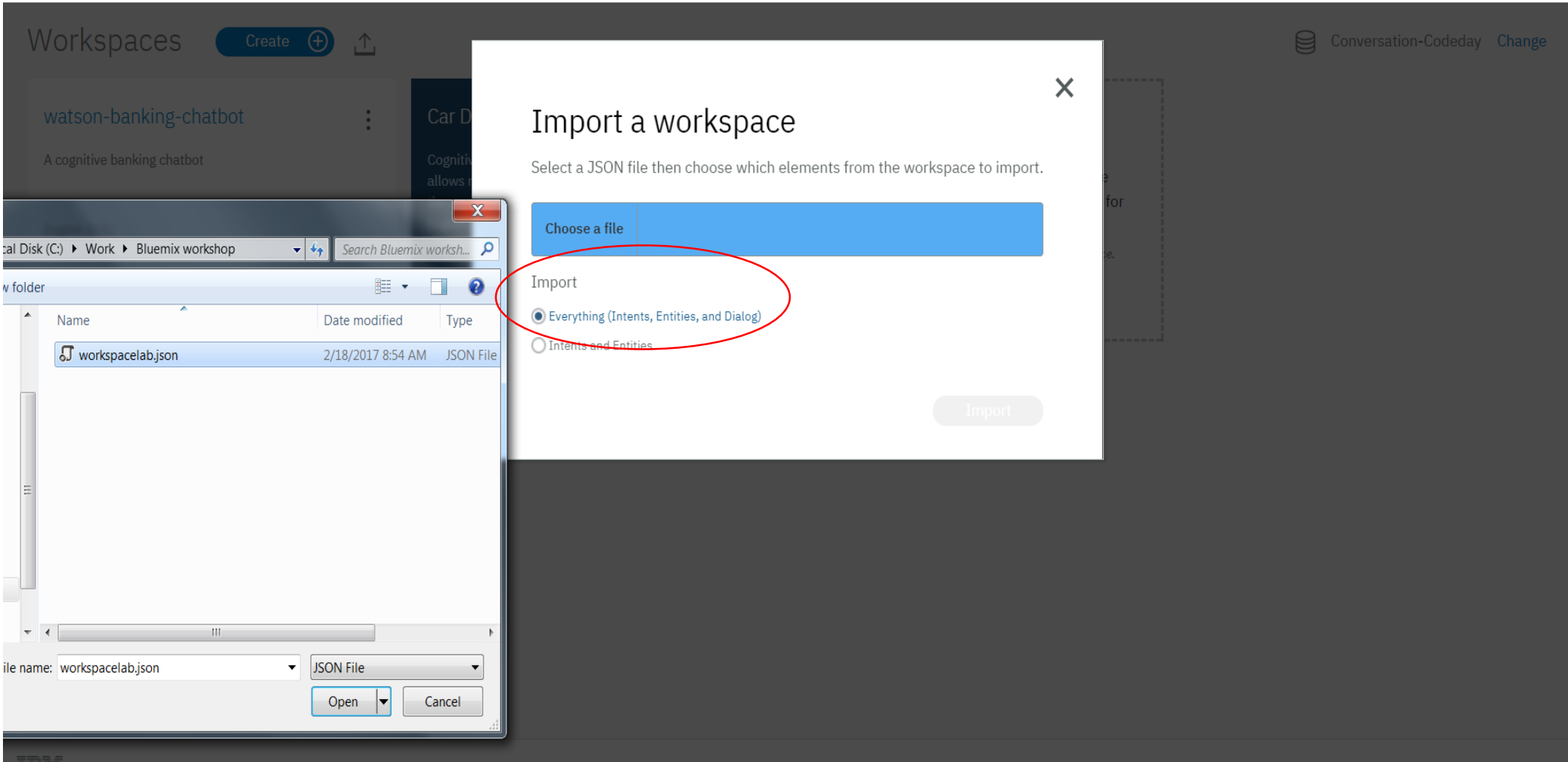
You are using 1 of 5 available workspaces in this instance.

Create 

 Conversation-Codeday [Change](#)

Browse to the workspace json file downloaded and select “everything (Intents, Entities and Dialog)” and click on Import button.

Watson Conversation



After importing, all the intents/entities and dialog will appear as below. Click on Intents tab.

Watson Conversation / watson-banking-chatbot / Build

Intents

Entities

Dialog

Add intent

<input type="checkbox"/> Intent (13) ▾	Description	Modified ▾	Examples
<input type="checkbox"/> #accountbalance		2 minutes ago	9
<input type="checkbox"/> #branchInfo		2 minutes ago	9
<input type="checkbox"/> #feedback		2 minutes ago	6
<input type="checkbox"/> #Forgotpassword		2 minutes ago	5
<input type="checkbox"/> #FundTransfer		2 minutes ago	4
<input type="checkbox"/> #Greetings		2 minutes ago	50
<input type="checkbox"/> #LostCard		2 minutes ago	7
<input type="checkbox"/> #Ok		2 minutes ago	21
<input type="checkbox"/> #payment		2 minutes ago	9
<input type="checkbox"/> #register		2 minutes ago	11
<input type="checkbox"/> #representative		2 minutes ago	5
<input type="checkbox"/> #Viewtransactions		2 minutes ago	9
<input type="checkbox"/> #weather		2 minutes ago	144

Click on Add Intent

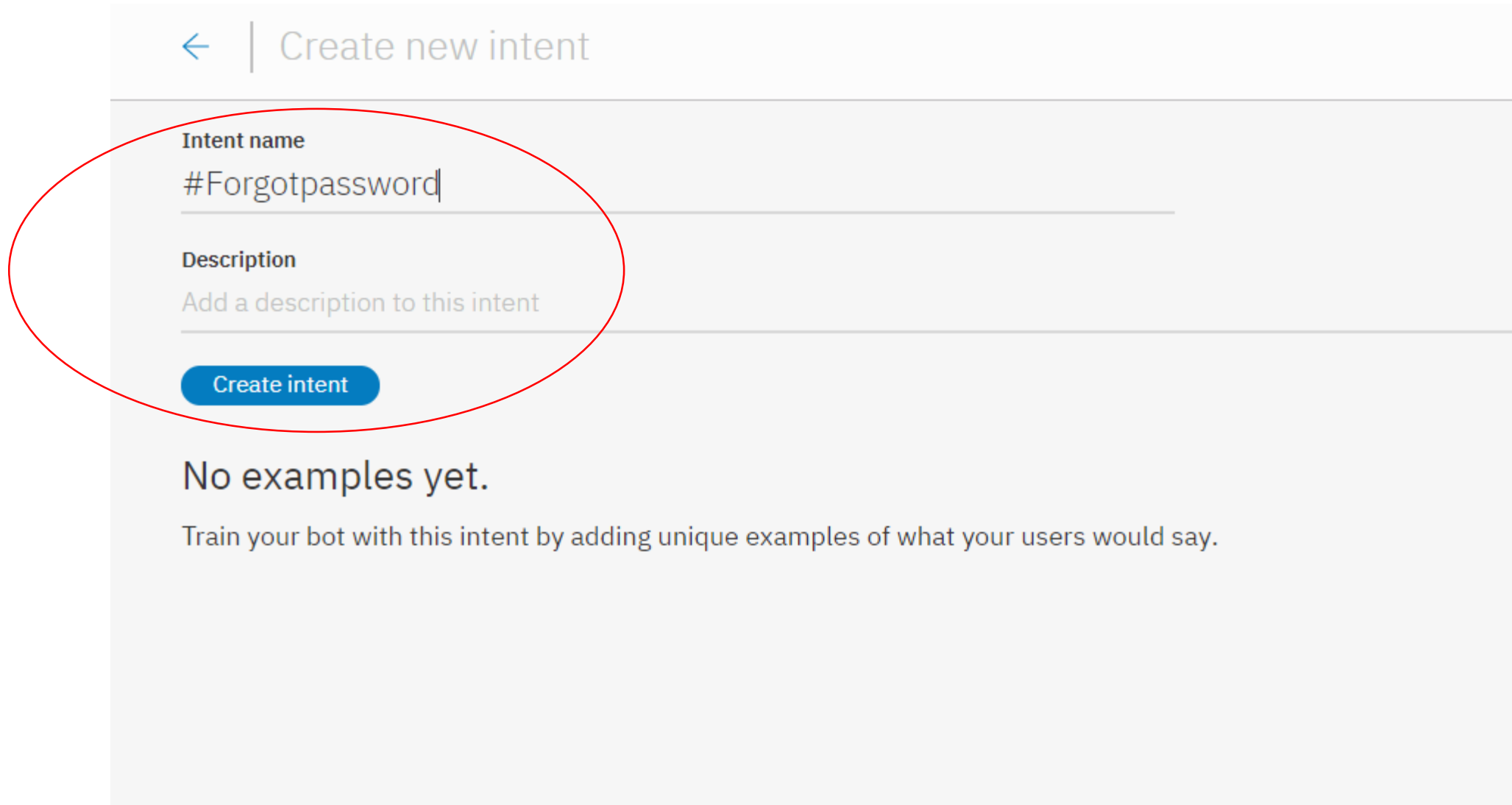
Watson Conversation / watson-banking-chatbot / Build

Intents Entities Dialog

Add intent   

<input type="checkbox"/> Intent (13) ▼	Description	Modified ▼	Examples
<input type="checkbox"/> #accountbalance		2 minutes ago	9
<input type="checkbox"/> #branchInfo		2 minutes ago	9
<input type="checkbox"/> #feedback		2 minutes ago	6
<input type="checkbox"/> #Forgotpassword		2 minutes ago	5
<input type="checkbox"/> #FundTransfer		2 minutes ago	4
<input type="checkbox"/> #Greetings		2 minutes ago	50
<input type="checkbox"/> #LostCard		2 minutes ago	7
<input type="checkbox"/> #Ok		2 minutes ago	21
<input type="checkbox"/> #payment		2 minutes ago	9
<input type="checkbox"/> #register		2 minutes ago	11
<input type="checkbox"/> #representative		2 minutes ago	5
<input type="checkbox"/> #Viewtransactions		2 minutes ago	9
<input type="checkbox"/> #weather		2 minutes ago	144

Provide Intent name as Forgotpassword and click on Create intent.



← | Create new intent

Intent name
#Forgotpassword

Description
Add a description to this intent

Create intent

No examples yet.

Train your bot with this intent by adding unique examples of what your users would say.

Click on Forgotpassword from the list of intents.
Start training the intent by adding few examples as shown below.
Provide 5-10 examples.

<

#Forgotpassword

Intent name

#Forgotpassword

Description

Add a description to this intent

Add user examples

Add user examples to this intent

Add example

☐ User examples (8) ▼

☐ change my password

☐ forgot pwd

☐ Help me reset the password

☐ i dont remember the password


☐ I forgot my password

☐ not able to login

☐ process to reset the password

☐ resetting password

Once you have added enough examples, click on the left arrow button.

 #Forgotpassword

Intent name
#Forgotpassword

Description
Add a description to this intent

Add user examples
Add user examples to this intent

Add example

☐ User examples (8) ▼




☐ change my password

☐ forgot pwd

Click on the “Dialog” tab

Watson Conversation / watson-banking-chatbot / Build

Intents Entities **Dialog**

Add intent   

☐ Intent (13) ▼ Description

☐ #accountbalance

☐ #ApplyLoan

☐ #branchInfo

☐ #feedback

☐ #Forgotpassword

☐ #FundTransfer

☐ #Greetings

Once in the Dialog tab, click on “Add node”

Watson Conversation / watson-banking-chatbot / Build

Intents Entities **Dialog**

Add node Add child node

watson-banking-chatbot

- conversation_start
1 Response / 0 Context set
- > detecting-weather
#weather
1 Response / 0 Context set / Skip user input
- #representative
1 Response / 1 Context set
- #FundTransfer
1 Response / 0 Context set
- #accountbalance

On the new node, type # into the place for “If bot recognizes” and select #Forgotpassword. Provide a suitable response similar to shown below.

Watson Conversation / watson-banking-chatbot / Build

IntentsEntitiesDialog

1 Response / 0 Context set

#LostCard

0 Responses / 0 Context set

#branchInfo

0 Responses / 0 Context set

#Greetings

1 Response / 0 Context set

#Forgotpassword

1 Response / 0 Context set

anything_else

1 Response / 0 Context set

Name this node...

If bot recognizes:

#Forgotpassword

Then respond with:

1. Sure, I can help you with password reset

Add a variation to this response

And finally

Wait for user input

Click on Entities tab -> my entities -> Add entity

Watson Conversation / watson-banking-chatbot / Build




Entities

My entities

Add entity

Entity (3) ▲	Values	Modified ▼
@spending_category	utility, travel, investment, grocery, fuel, education, dining	a few seconds ago
@acknowledgement	yes, no	a few seconds ago
@account_type	investment, FD, current, CC, savings, preloan, persloan, homeloan, carloan	a few seconds ago


Provide a name as @card_type and click on Create entity.

 | Create new entity  

Entity name

@card_type

Create entity

Fuzzy Matching ^{BETA}  ☐ Off

No values yet.

Once you've named your entity, begin by adding values, synonyms, and patterns to entities to help your bot learn and understand important details that your users mention.

Start adding values to the entity as CreditCard and provide few synonyms as shown.
Add value of Debit card and provide few synonyms.

<

@card_type

Last modified a few seconds ago

↓

🗑️

Entity name

@card_type

Fuzzy Matching BETA

●

 Off

Value name

Enter value

Synonyms

Synonyms

▼

Enter synonym

+

Add value

☐

 Entity values (2) ▼

Type

☐

 CreditCard

Synonyms

▼

credit card

-

CC

-

credit

-

Add synonyms...

+

☐

 DebitCard

Synonyms

Debit card debit

Click on System entities tab, here you will find pre built entities. These can be used in your dialog. Enable sys currency, sys-date and sys-number entities.

Intents **Entities** Dialog

My entities **System entities**

These are common entities created by IBM that could be used across any use case. They are ready to use as soon as you add them to your workspace. *System entities cannot be edited. [Learn more](#)

Name (7) ▼	Description	Status
> @sys-currency	Extracts currency values from user examples including the amount and the unit. (20 cents)	<input checked="" type="checkbox"/> On
> @sys-date	Extracts date mentions (Friday)	<input checked="" type="checkbox"/> On
> @sys-location <small>BETA</small>	The @sys-location system entity extracts place names (country, state/province, city, town, etc.) from the user's input. (Boston)	<input type="checkbox"/> Off
> @sys-number	Extracts numbers mentioned from user examples as digits or written as numbers. (21)	<input checked="" type="checkbox"/> On
> @sys-percentage	Extracts amounts from user examples including the number and the % sign. (15%)	<input type="checkbox"/> Off
> @sys-person <small>BETA</small>	The @sys-person system entity extracts names from the user's input. (Anna)	<input type="checkbox"/> Off
> @sys-time	Extracts time mentions (at 10)	<input type="checkbox"/> Off

We will now look into creating slots.
Click on Dialog tab, go the node with #Applyloan.
Click on Customize button highlighted below.

Watson Conversation / watson-banking-chatbot / Build



Intents Entities **Dialog**

A list of dialog nodes in the Watson Conversation interface. The nodes are: #FundTransfer, #accountbalance, #Viewtransactions, #ApplyLoan (highlighted with a blue border), #payment, and #feedback. Each node shows its name, a small menu icon (three dots), and its configuration (e.g., '1 Response / 0 Context set').

Name this node...

Customize

If bot recognizes:

#ApplyLoan

Then respond with:

Enter a response...

And finally

Wait for user input

Enable slots and multiple responses as shown below and click on Apply button

The image shows a 'Customize #ApplyLoan' dialog box with a dark sidebar on the left. The sidebar contains several menu items, each with three dots to its left. The dialog box has a title bar and a main content area. The 'Slots' section is active, showing a toggle switch set to 'on' and a checkbox for 'Prompt for everything' which is currently unchecked. The 'Multiple responses' section is also visible, with its toggle switch set to 'on'. At the bottom of the dialog are 'Cancel' and 'Apply' buttons.

Customize "#ApplyLoan"

Slots ⓘ on

Enable this to gather the information your bot needs to respond to a user within a single node.

☐ **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.

Multiple responses ⓘ on

Enable multiple responses so that your bot can provide different responses to the same input, based on other conditions.

Cancel Apply

Enter the values as shown below.
Here we are checking for @account_type and @sys-number entities.

Name this node...

Customize

X

If bot recognizes:

#ApplyLoan

Then check for:

Manage handlers

	Check for	Save it as	If not present, ask	Type		
1	@account_type	\$account_type	What loan are you lc	Required		
2	@sys-number	\$number	What is the amount	Required		

+ Add slot

Change the field “Save it as” to \$loan_type and \$amount as shown below.
Enter a response in the field “Respond with”. Click on the settings button next to the response.

Name this node...

Customize

X

Then check for:

Manage handlers

	Check for	Save it as	If not present, ask	Type		
1	@account_type	\$loan_type	What loan are you lc	Required	⚙	🗑
2	@sys-number	\$amount	What is the amount	Required	⚙	🗑

+ Add slot

Then respond with:

	If bot recognizes	Respond with		
1	Enter an intent, entity or context vari	[Thanks for providing the info, we ha	⚙	🗑


+ Add response

And finally


The configure response window opens up. Click on three dots to open the JSON editor.

Configure response 1

If bot recognizes:

Enter an intent, entity or context variable... 

Then respond with:

1. Thanks for providing the info, we have taken your request for \$loan_type for \$amount 


Add a variation to this response



Modify the JSON response as below

Configure response 1

If bot recognizes:

Enter an intent, entity or context variable... 

Then respond with:

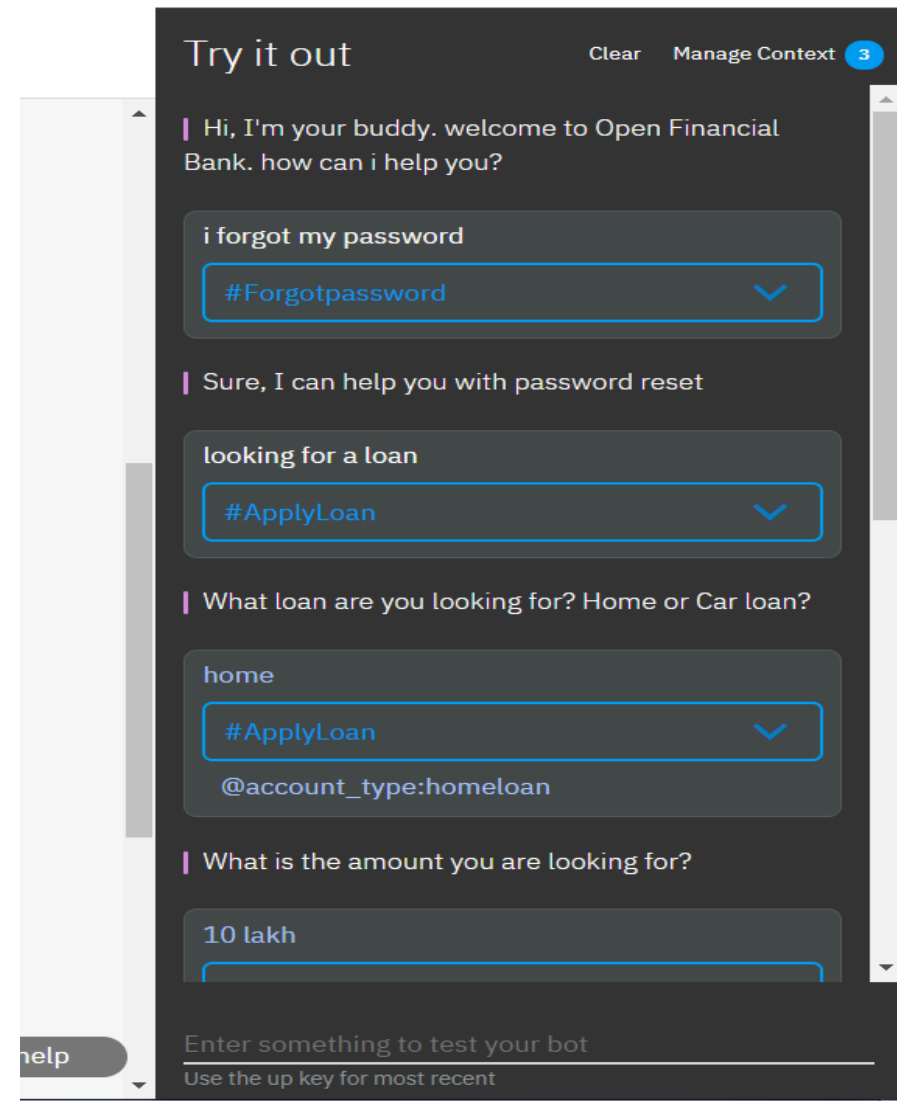
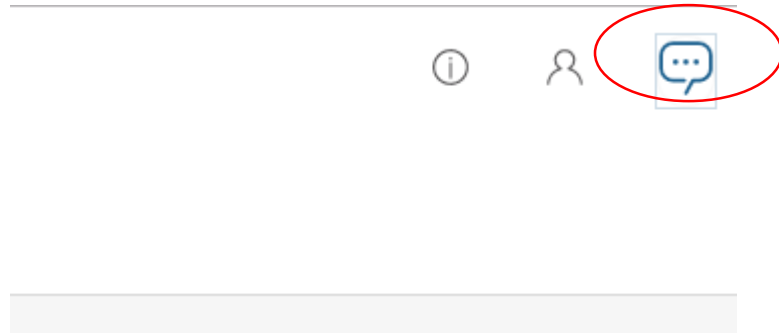


```
1 {  
2   "output": {  
3     "text": {  
4       "values": [  
5         "Thanks for providing the info, we have taken your request for $loan_type  
6         for $amount, we will soon get back to you."|  
7       ],  
8       "selection_policy": "sequential"  
9     }  
10  }
```

Cancel

Save

You can now try out the bot response by clicking on Try out



Part 2

Configuring Watson Discovery and Cloud Functions

- Lets create our **discovery** service to get started with this part of the lab.
- Ensure that your space is in the 'US-South' region, and go to Catalog -> Watson
 - Search for 'Watson Discovery' service, give it the name you like and create a '**Lite**' plan.

Discovery

Add 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.

Service name:

Discovery-codeday

Choose a region/location to deploy in:

US South

Choose an organization:

saurav.raiguru@in.ibm.com

Choose a space:

dev

Lite

IBM

[View Docs](#)

AUTHOR

IBM

PUBLISHED

12/16/2017

TYPE

Service

LOCATION

Sydney, Germany, United Kingdom, US South

Pricing Plans

Monthly prices shown are for country or region: [India](#)

	PLAN	FEATURES	PRICING
✓	Lite	0 - 2,000 documents (or 200 MB) per month 1,000 news queries per month 1 custom model 500 element classification pages per month See documentation for plan details	Free

The Lite plan gets you started with 2,000 documents, 1,000 news queries, 1 custom model and 500 element classification pages per month at no cost. Assumes average document size. Storage limits may also apply. When you upgrade to a paid plan, you'll keep any content you have ingested. See documentation for details.

Need Help?

[Contact IBM Cloud Sales](#)

Estimate Monthly Cost

[Cost Calculator](#)

Create

- Click on '**service credentials**' on the left menu and click on '**New Credentials**' -> '**Add**'

Add new credential

Name:

Credentials-2

Add Inline Configuration Parameters (Optional): ⓘ

Provide service-specific configuration parameters in a valid JSON object

Choose File...

Cancel Add

New credential +

1 of 1 pages < 1

ACTIONS


View credentials ▼

Once added the credentials, click on **View Credentials**


Note: Make a note of the Discovery **Username** and **Password** to be used later.

- Once we have created the service, click on **Manage** (left menu) and '**launch**' the tooling, wherein we can add our documents required for cognitive searching.

[Watson](#) /

 Discovery-codeday

Location: US South **Org:** sudharshan.govindan@gmail.com **Space:** dev



Discovery

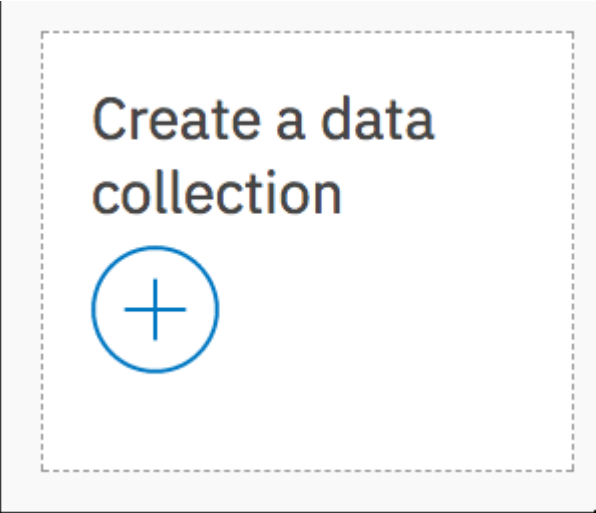
Add a cognitive search and content analytics engine to applications to identify patterns, trends and actionable insights that drive better decision-making

[Launch tool](#)

Developer resources:

- [Getting started tutorial](#)
- [Demo](#)

Lets create our own discovery '**Collection**', which will contain information related to banking domain.
Click on 'create a data collection -> Collection Name -> Create



Name your new collection

Collection name

Enter name

Select a configuration to apply

Default Configuration

Select the language of your documents

English

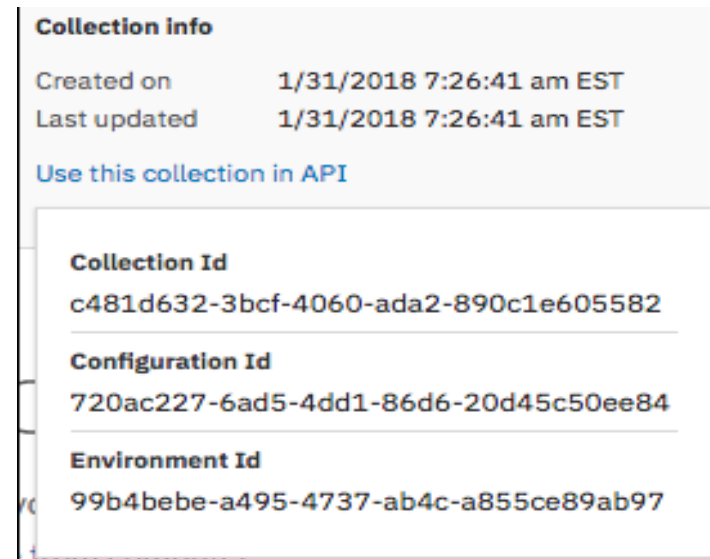
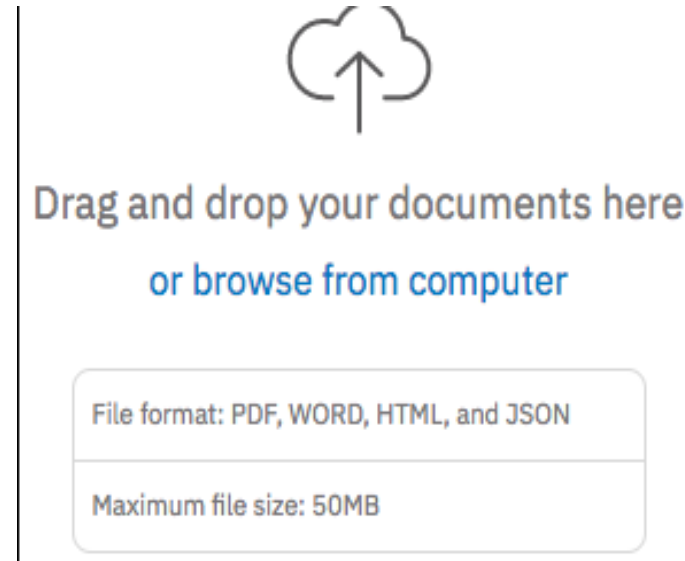
Cancel

Create

- Use the downloaded documents (<https://github.com/IBMDevConnect/IBMCodeDay-2018/tree/master/AIBankingWorkshop/discovery-docs>) as a sample content to **upload** documents into your collection.

Note: You will see default analysis of your contents such as sentiment, entity-relationships, context extraction etc.

- Click on 'Use this collection in API' to gather the credentials, and make a note of the variables
Collection Id & Environment Id



2. We will now create our IBM Function ***action*** which will implement the discovery REST APIs to query our sample collection.

- Go to IBM Cloud -> Catalog -> Functions

Getting Started with IBM Cloud Functions

IBM Cloud Functions (based on Apache OpenWhisk) is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events and **costs nothing** when not in use. [Learn More](#)

Start Creating

Download CLI



Create Action

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

lets give our action name as - ***mydiscoverycall***
enclosing package as - default
Runtime - NodeJS 6.0 distribution -> Create

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

Mydiscoverycall

Enclosing Package

(Default Package)

Create Package

Runtime

Node.js 6

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

Cancel

Previous

Create

Now lets replace our hello world () nodejs code with the code logic that can query the discovery service and return us the values.

Copy the below code from the **FunctionsActionScript.txt** file downloaded earlier (available at <https://github.com/IBMDevConnect/IBMCodeDay-2018/tree/master/AIBankingWorkshop/>)


```
function main(params) {
  const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');
  return new Promise(function (resolve, reject) {

    var discovery = new DiscoveryV1({
      username: params.username,
      password: params.password,
      version_date: '2017-11-07'
    });

    discovery.query({environment_id: params.environment_id,
      collection_id: params.collection_id,
      query: params.input
    }, function(err, data) {
      if (err) {
        return reject(err);
      }
      return resolve(data);
    });

  });
}
```

- By clicking on **Endpoints**, we need to make a note of some REST API credentials such as **URL** (also is the action name). Please note the highlighted part which will be used as Action name later (replace the %40 with @)



REST API		
HTTP METHOD	AUTH	URL
POST	API-KEY	https://openwhisk.ng.bluemix.net/api/v1/namespaces/saurav.raiguru%40in.ibm.com_dev/actions/mydiscoveryCall 

- Click on **API-KEY** and copy the credentials from IBM Function into the following format. The segment before the colon (:) is your user ID. The segment after the colon is your password. {"user": "<left part of colon>", "password": "<right part of colon>"}

API Key

Start Creating

The IBM Cloud Functions **REST API** is protected with an API Key.
Each Namespace has a unique API Key assigned, which needs to be provided via HTTP Basic authentication when calling any API in that Namespace, except for Web Actions invocations.

API Key		
CURRENT NAMESPACE	HOST	KEY
saurav.raiguru@in.ibm.com_dev	openwhisk.ng.bluemix.n 	c4774bf0-006a-4  

We are in our final step to call our Action from the conversation workspace being used previously.

- Lets go back to **Dialog** and search and click on a node called '**Others**'
- Open the JSON editor and paste the below code which is available in the file **WCSNodeScript.txt** downloaded earlier (available at <https://github.com/IBMDevConnect/IBMCodeDay-2018/tree/master/AIBankingWorkshop/>)
- The value for the "name" parameter under "actions" is the one which you saved in previous step
- The username, password, collection_id and environment_id are as per saved in earlier step.

```
{
  "output": {
    "text": {
      "values": [
        "$data"
      ],
      "selection_policy": "sequential"
    }
  },
  "actions": [
    {
      "name": "<yourIBMCLOUDId>_dev/actions/mydiscoveryCall",
      "type": "server",
      "parameters": {
        "url": "https://gateway.watsonplatform.net",
        "input": "<? input.text ?>",
        "password": "<Discovery password saved earlier>",
        "username": "<Discovery username saved earlier>",
        "collection_id": "<replace-collectionid>",
        "environment_id": "<replace-environmentid>"
      },
      "credentials": "$mycreds",
      "result_variable": "$data"
    }
  ]
}
```

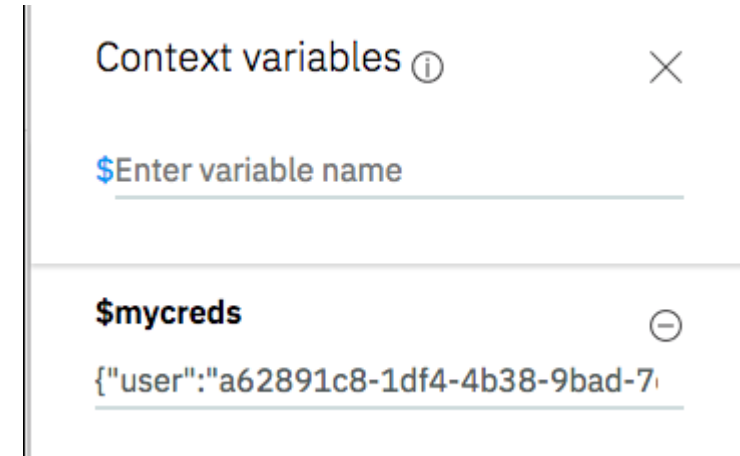
- Now, we need to set the context of WCS workspace.

From the **Try it out** pane, click **Manage context**, and then (temporarily) set the context variables to your Cloud Functions username and password saved earlier.

Create context variable name – mycreds

Copy the credentials you saved earlier using API key into the value field.

```
{"user": "<left part of colon>", "password": "<right part of colon>"}
```



Try out your first serverless programmatic call to fetch data from Discovery service (cognitive search) from Try out Pane:

Sample questions:

- Can I enter a mobile number other than the one used for downloading the app.
- how to change my aadhaar card address
- My beneficiary's account is not yet credited with the amount I have transferred
- From the list of beneficiaries, I am not able to see the name of beneficiary I have added earlier
- Why I am not able to select loan amount below 25000?