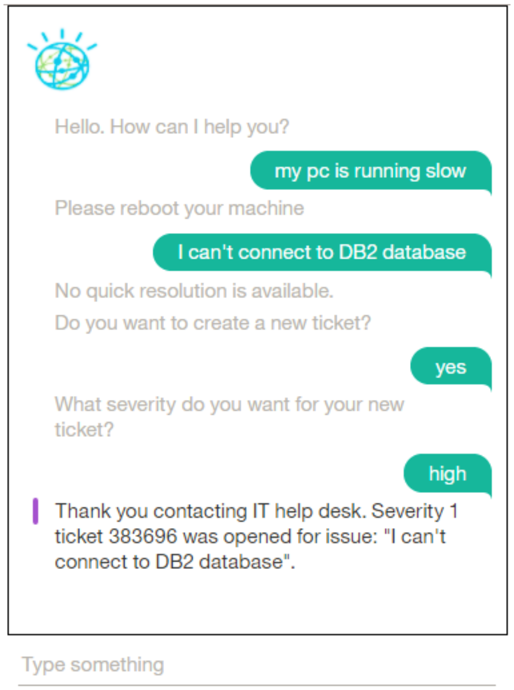
build error

IBM Cloud Deployments 2167

Virtual HelpDesk using IBM Watson Assistant, Discovery service & Maximo/IBM Control Desk

This Node.js application demonstrates how to build a `Virtual HelpDesk`, and use the Watson Assistant(Conversation) and Discovery services to interact with end users for simple Q/A. With proper training, Assistant(Conversation) service can cover most of common questions/requests. When it is not been trained to address end users' specific question, the virtual agent searches in the knowledge base through Watson Discovery service and presents relevant entries to the end user. If the end user is still not satisfied, a new ticket is created in a back-office ticketing system, such as Maximo/IBM Control Desk(ICD) system.



Hello. How can I help you?

my pc is running slow

Please reboot your machine

I can't connect to DB2 database

No quick resolution is available.
Do you want to create a new ticket?

yes

What severity do you want for your new ticket?

high

Thank you contacting IT help desk. Severity 1 ticket 383696 was opened for issue: "I can't connect to DB2 database".

Type something

Flow

1. The knowledge base documents are added to the Discovery collection.
2. The user interacts with the virtual agent via the app UI. User input is processed by the virtual agent.
3. When the virtual agent is trained to cover the discussion subject, it quickly provide feedback.
4. When the virtual agent is not trained to cover the discussion subject, it searches in the knowledge base for suggestion(s). If found, relevant suggestions are displayed via the app UI.
5. If no relevant information is found in the knowledge base or the relevant information from knowledge base does not satify end users, the Virtual HelpDesk opens a new ticket in back-office ticketing system.
6. Back-office ticketing system takes over the support task.

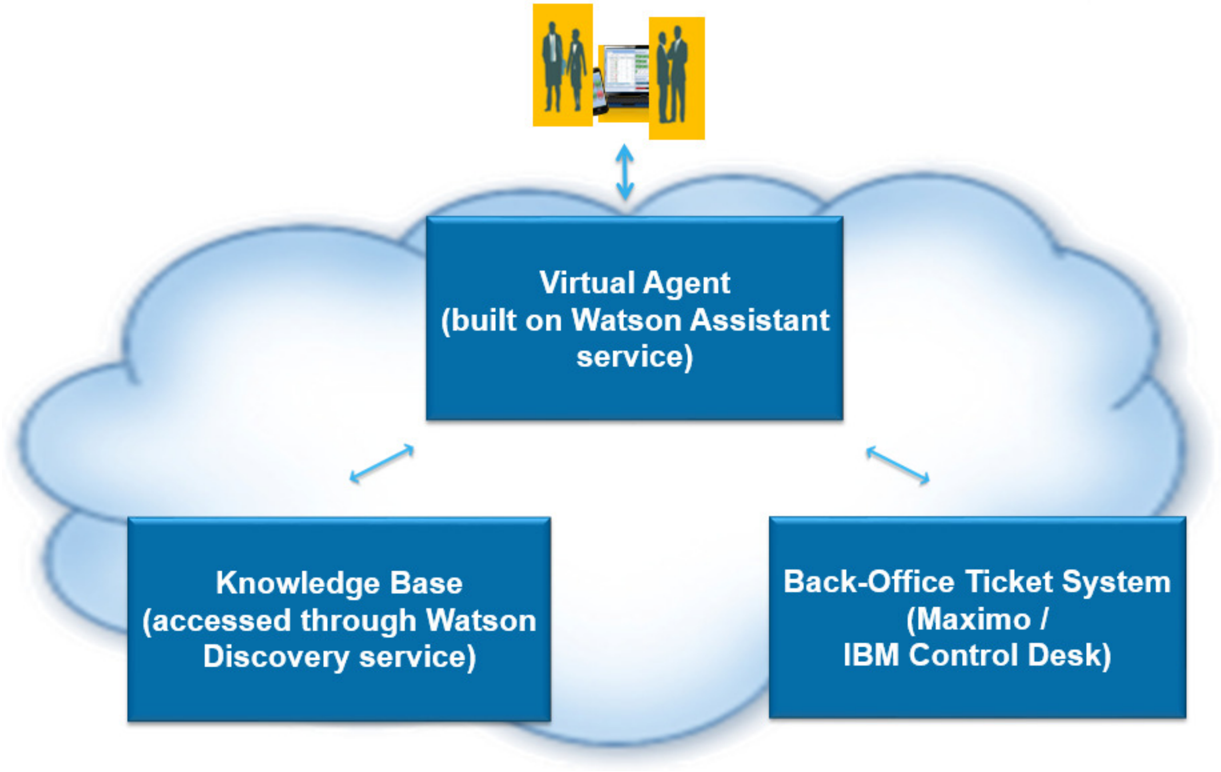
With Watson

Want to take your Watson app to the next level? Looking to leverage Watson Brand assets? Join the With Watson program which provides exclusive brand, marketing, and tech resources to amplify and accelerate your Watson embedded commercial solution.

Included components

- IBM Watson Assistant: Build, test and deploy a bot or virtual agent across mobile devices, messaging platforms, or even on a physical robot.

- IBM Watson Discovery: A cognitive search and content analytics engine for applications to identify patterns, trends, and actionable insights.
- Maximo/IBM Control Desk(ICD): Back-office ticketing system. Other ticketing system can be used alternatively.


Virtual Agent (built on Watson Assistant service)

Knowledge Base (accessed through Watson Discovery service)

Back-Office Ticket System (Maximo / IBM Control Desk)

Featured technologies

- Node.js: An asynchronous event driven JavaScript runtime, designed to build scalable applications.

Watch the Video

Steps

Before you begin

- Create a Bluemix account

- Sign up in Bluemix, or use an existing account. Your account must have available space for at least 1 app and 1 service.
- Make sure that you have the following prerequisites installed:
    - The Node.js runtime, including the npm package manager
    - The Cloud Foundry command-line client

      Note: Ensure that you Cloud Foundry version is up to date

Setting up Assistant(Conversation) service

Watson Assistant(Conversation) service is used to provide underline infrastructure for the virtual agent in this code pattern.

Creating an Assistant(Conversation) service

Watson Assistant(Conversation) service is to be setup to simulate help desk level 1 activities. For topics that the virtual agent has been trained, it can help end users interactively. For subjects that the virtual agent does not understand, it searches the knowledge base through Discovery service, collects information from end user and creates a new ticket in back-oofice ticketing system, for example Maximo/ICD, if necessary.

Slots are configured in the Assistant(Conversation) service to collect additional information from end users.

1. At the command line, go to the local project directory (`vaticketbot`).
2. Connect to Bluemix with the Cloud Foundry command-line tool. For more information, see the Watson Developer Cloud documentation.

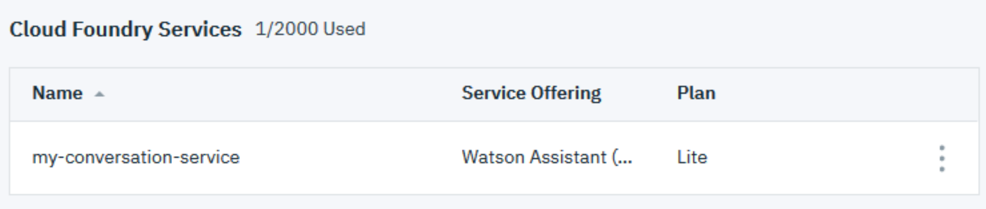
```
cf login
```

3. Create an instance of the Assistant(Conversation) service in Bluemix. For example:

```
cf create-service conversation free my-conversation-service
```

Importing the Assistant(Conversation) workspace

1. In your browser, navigate to [your Bluemix console] (https://console.ng.bluemix.net/dashboard/services).
2. From the **All Items** tab, click the newly created Assistant(Conversation) service in the `Cloud Foundar Services` list.



Cloud Foundry Services 1/2000 Used

Name	Service Offering	Plan
my-conversation-service	Watson Assistant (...	Lite

3. On the next page, click `Launch tool` .
4. In the Watson Assistant page, navigate to `Workspace` tab.
5. Click the `Import workspace` icon on the top of the Assistant(Conversation) Workspaces.
6. Specify the location of the workspace JSON file in your local copy of the app project:

   `<project_root>/training/ITSM_workspace.json`
7. Select `Everything (Intents, Entities, and Dialog)` option and then click `Import` .
8. The sample ITSM workspace is created.

Setting up Discovery service

Watson Discovery service is used to provide underline infrastructure in this code pattern when searching in knowledge base.

Creating a Discovery service

Watson Discovery service is to be setup to search in the knowledge base when the virtual agent is not trained to cover specific topics.

1. At the command line, go to the local project directory (`vaticketbot`).
2. Connect to Bluemix with the Cloud Foundry command-line tool. For more information, see the Watson Developer Cloud documentation.

```
cf login
```

3. Create an instance of the Discovery service in Bluemix. For example:

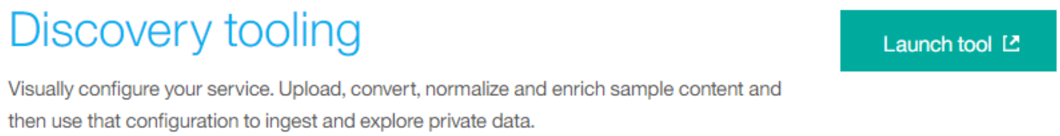
```
cf create-service discovery lite my-discovery-service
```

4. Check the status of Discovery service instance in Bluemix, if necessary

```
cf services
```

Creating a collection and ingesting documents into Discovery service

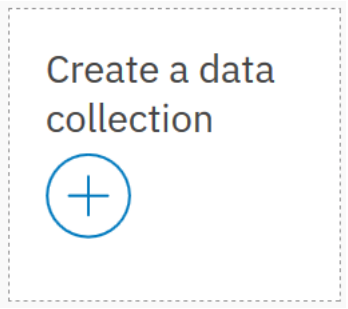
1. Download and unzip the knowledgebase.zip in this repo to reveal a set of JSON documents
2. Navigate to your Discovery instance in your Bluemix dashboard
3. Click `Launch tool`



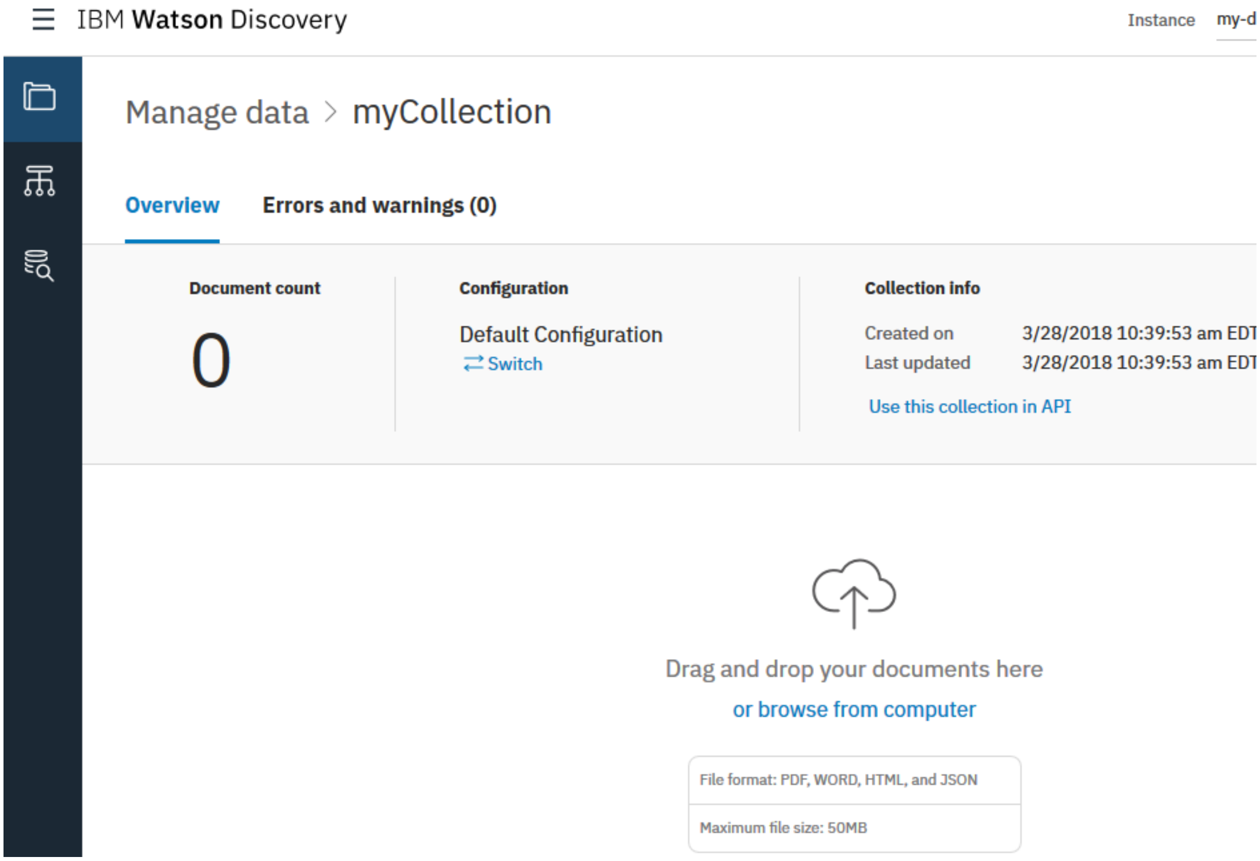
Discovery tooling

Visually configure your service. Upload, convert, normalize and enrich sample content and then use that configuration to ingest and explore private data.

4. Create a new data collection, name it whatever you like, and select the default configuration.



5. After you're done, a new private collection is displayed in the UI



IBM Watson Discovery

Instance my-d

Manage data > myCollection

Overview | Errors and warnings (0)

Document count	Configuration	Collection info	
0	Default Configuration ⇄ Switch	Created on	3/28/2018 10:39:53 am EDT
		Last updated	3/28/2018 10:39:53 am EDT
		Use this collection in API	

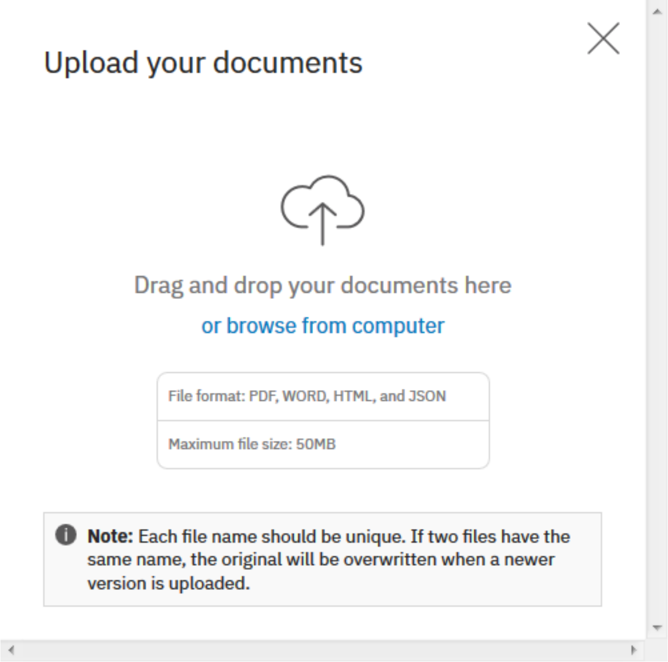
Drag and drop your documents here

or browse from computer

File format: PDF, WORD, HTML, and JSON

Maximum file size: 50MB

6. Click `Drag and drop your documents here or browse from computer` section



Upload your documents

Drag and drop your documents here

or browse from computer

File format: PDF, WORD, HTML, and JSON

Maximum file size: 50MB

Note: Each file name should be unique. If two files have the same name, the original will be overwritten when a newer version is uploaded.

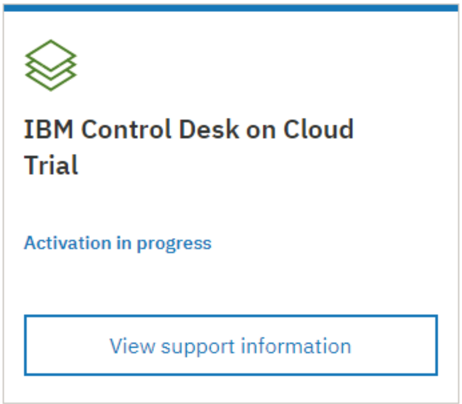
7. Select three JSON files from local file system where you downloaded and upzipped knowledgebase.zip file

- This may take a few seconds – you will see a notification when the process is finished

Setting up trial IBM Control Desk SaaS system

If you don't have an available in-house Maximo/ICD system to integrate with Watson services in this code pattern, you may request a trial ICD SaaS system.

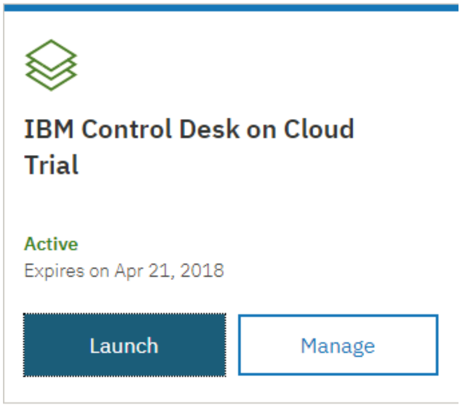
You may request a trial ICD SaaS system at no cost. Click the `Free 30-day trial` link and follow the procedure. It may take a while for the system orchestration to complete.



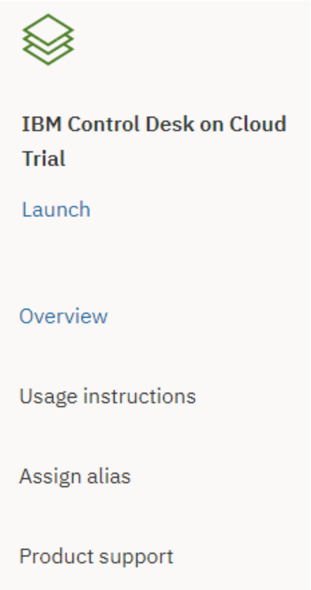
After the trial ICD SaaS system is active, you should receive an email for your trial ICD system.

```
Thank you for requesting IBM Control Desk on Cloud. Your trial is now ready for you to start

The Products and Services page provides access to all of your trials and subscriptions, and i

Your trial is valid through Sun, 22 Apr 2018 02:09 UTC.

Enjoy your trial to IBM Control Desk on Cloud!

Sincerely,
IBM Marketplace Team
```

Click Products and Services link in the email to navigate to your Products and service home page. One of trial offers is `IBM Control Desk on Cloud Trial`.



Click `Manage` button to review the Overview page of your trial ICD system. In the navigation pane on the left, select `Usage instructions`.



Default account information is displayed on `Usage instructions` page. Take a note of password for maxadmin account for further code pattern configuration.

```
Your trial is predefined with demo data and several sign in IDs that you can use to see how IB

    bob - End user (originates tickets, service requests, and catalog requests)
    scott - Service Desk Agent (handles service requests, incidents, and problems)
    franklin - Change Manager (works with changes, releases, and the configuration management
    jake - Asset Manager (handles hardware and software assets)
    maxadmin - Super user (has full administrative rights)
```

click `Launch` button to bring up ICD login screen. Note down the login page URL for late configuration. For example,

```
https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/maximo_t4hj/webclient/login/login.jsp?welcom
```

Login to your trail ICD SaaS system and verify it's working.

Installing locally

If you want to modify the app or use it as a basis for building your own app, install it locally. You can then deploy your modified version of the app to the Bluemix cloud.

Getting the files

Use GitHub to clone the repository locally,

Configuring the Assistant(Conversation) service environment

1. Copy the `.env.example` file and create a new `.env` file.
2. In the Bluemix with the Cloud Foundry command-line tool, create a service key for the Assistant(Conversation) service in the format `cf create-service-key <service_instance> <service_key>`. For example:

   ```
   cf create-service-key my-conversation-service myKey
   ```

3. Retrieve the credentials from the service key of the Assistant(Conversation) service using the command `cf service-key <service_instance> <service_key>`. For example:

   ```
   cf service-key my-conversation-service myKey
   ```

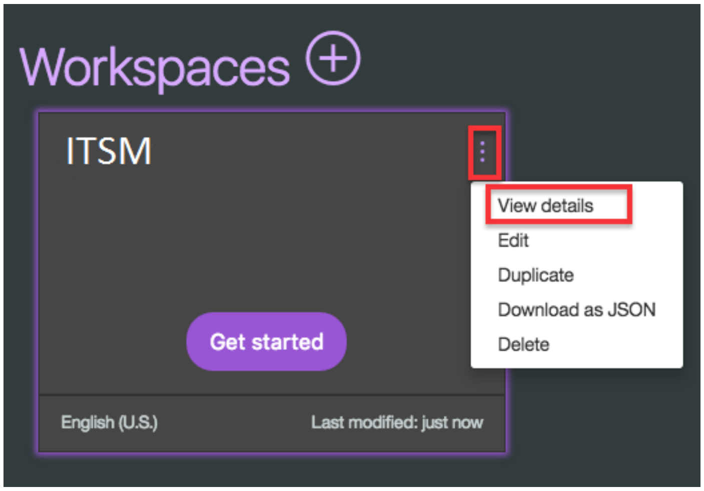
   The output from this command is a JSON object, as in this example:

   ```
   {
     "password": "87iT7aqpvU7l",
     "url": "https://gateway.watsonplatform.net/conversation/api",
     "username": "ca2905e6-7b5d-4408-9192-e4d54d83e604"
   }
   ```

4. Copy and paste the `password` and `username` values (without quotation marks) from the JSON into the `CONVERSATION_PASSWORD` and `CONVERSATION_USERNAME` variables in the `.env` file. For example:

   ```
   CONVERSATION_USERNAME=ca2905e6-7b5d-4408-9192-e4d54d83e604
   CONVERSATION_PASSWORD=87iT7aqpvU7l
   ```

5. In your Bluemix console, open the Assistant(Conversation) service instance where you imported the workspace.
6. Click the menu icon in the upper-right corner of the workspace tile, and then select **View details**.



7. Click the 📋 icon to copy the workspace ID to the clipboard.
8. On the local system, paste the workspace ID into the WORKSPACE_ID variable in the `.env` file.
9. Save the file.

Configuring the Discovery service environment

1. In the Bluemix with the Cloud Foundry command-line tool, create a service key for the Discovery service in the format `cf create-service-key <service_instance> <service_key>`. For example:

```
cf create-service-key my-discovery-service myKey
```

2. Retrieve the credentials from the service key of the Assistant(Conversation) service using the command `cf service-key <service_instance> <service_key>`. For example:

```
cf service-key my-discovery-service myKey
```

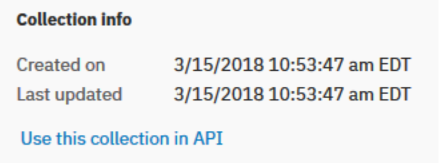
The output from this command is a JSON object, as in this example:

```
JSON
{
    "password": "E8CCHs37pUwj",
    "url": "https://gateway.watsonplatform.net/discovery/api",
    "username": "07629c30-a460-436d-ae54-97a3b6e71902"
}
```

3. Copy and paste the `password`, `username` and `username` values (without quotation marks) from the JSON into the `.env` file. For example:

```
DISCOVERY_USERNAME=07629c30-a460-436d-ae54-97a3b6e71902
DISCOVERY_PASSWORD=E8CCHs37pUwj
DISCOVERY_URL=https://gateway.watsonplatform.net/discovery/api/v1
```

4. In your Bluemix console, open your Discovery service instance.
5. Open the Collection in your Discovery service.
6. Locate the `collection info` section,



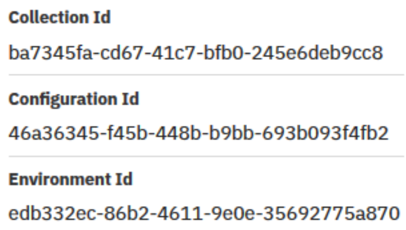
Collection info

Created on 3/15/2018 10:53:47 am EDT

Last updated 3/15/2018 10:53:47 am EDT

Use this collection in API

7. click `Use this collection in API` link to display the collection information.



Collection Id

ba7345fa-cd67-41c7-bfb0-245e6deb9cc8

Configuration Id

46a36345-f45b-448b-b9bb-693b093f4fb2

Environment Id

edb332ec-86b2-4611-9e0e-35692775a870

8. Copy and paste Collection ID and Environment ID to the corresponding variable in the `.env` file.
9. Save the file.

Configuring the Maximo/ICD environment

1. Set MAXIMO_AUTH environment variable in file .env. This variable setting depends on how Maximo/ICD authentication is configured.
    - `Application Server Authentication (LDAP)` - In this case, the variable has two parts separated by a blank space. The first part is the value “Basic”. The second part is `user:password` base64 encoded. You can get its value through any online base64 encoder based on your ICD/Maximo user:password.
    - `Native Maximo Authentication` - In this case, the variable has one part only. It is `user:password` base64 encoded. You can get its value through any online base64 encoder based on your ICD/Maximo user:password. Note, the trial ICD SaaS system has navive Maximo authentication.
2. Keep “application/json” as the value of MAXIMO_CONTEXT_TYPE environment variable.
3. Modify the hostname portion of MAXIMO_REST_URL environment variable to point to your ICD/Maximo system. If you are connecting to trial ICD system, you may have to modify its context root as well. For example, if the URL used to login to the trial ICD system is https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/maximo_t4hj, the URL in the .env file will be https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/meaweb_t4hj/os/MXSR.
4. Set MAXIMO_PERSONID environment variable to a valid person ID in your ICD/Maximo system. For example, MAXADMIN. Note, the person ID is typically case sensitive.
5. Set MAXIMO_UI_URL environment variable in the similar way as you have done for MAXIMO_REST_URL environment variable. Change its hostname and context root.

```
# For Application Server Authentication (LDAP)
MAXIMO_AUTH=Basic bWF4YWRtaW46c056dXhYN1M=
MAXIMO_CONTEXT_TYPE=application/json
MAXIMO_REST_URL=https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/meaweb_t4hj/os/MXSR
MAXIMO_PERSONID=MAXADMIN
MAXIMO_UI_URL=https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/maximo_t4hj/ui/?event=loada
```

```
# For Native Maximo Authentication
MAXIMO_AUTH=bWF4YWRtaW46c056dXhYN1M=
MAXIMO_CONTEXT_TYPE=application/json
MAXIMO_REST_URL=https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/meaweb_t4hj/os/MXSR
MAXIMO_PERSONID=MAXADMIN
MAXIMO_UI_URL=https://siwr35cdwsa-tr3.sccd.ibmserviceengage.com/maximo_t4hj/ui/?event=loada
```

6. Save the file.

Installing and starting the app

1. Navigate to the folder where your local ticketbot application locates.
2. Install required Node.js modules to the local runtime environment:

```
npm install
```

3. Start the Virtual HelpDesk:

```
npm start
```

4. Point your browser to http://localhost:3000 to try out the app.

Running the use cases

When pointing your browser to http://localhost:3000, you are starting a Q/A session.

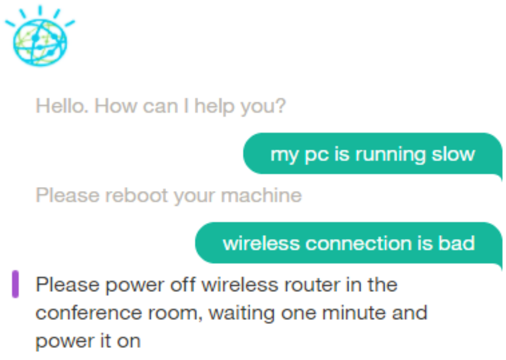
Watson Assistance(Conversation) delivers

You may type problem statements such as

- my pc is running slow
- wireless connection is bad

The virtual agent will do its best to address the issue, for example

- Please reboot your machine
- Please power off wireless router in the conference room, waiting one minute and power it on



Hello. How can I help you?

my pc is running slow

Please reboot your machine

wireless connection is bad

Please power off wireless router in the conference room, waiting one minute and power it on

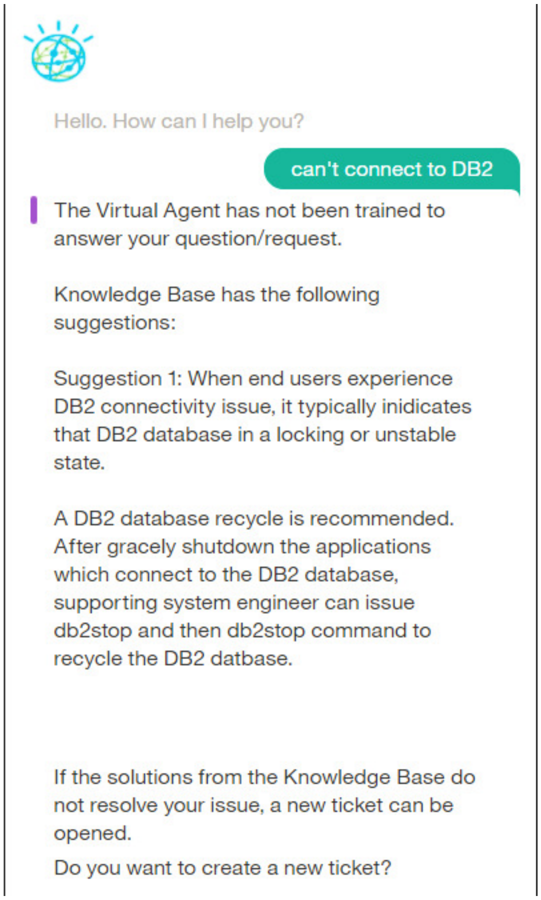
Watson Discovery comes to rescue

When end users have any question/request that the virtual agent has not been trained to understand, it searches in the knowledge base through Watson Discovery service and presents relavant entries as suggestion(s) to the end users.

For example, when you enter

- can’t connect to DB2

in the Q/A session, the virtual agent may return suggestion(s) depending on information in your knowledge base.



Hello. How can I help you?

can't connect to DB2

The Virtual Agent has not been trained to answer your question/request.

Knowledge Base has the following suggestions:

Suggestion 1: When end users experience DB2 connectivity issue, it typically inidicates that DB2 database in a locking or unstable state.

A DB2 database recycle is recommended. After gracely shutdown the applications which connect to the DB2 database, supporting system engineer can issue db2stop and then db2stop command to recycle the DB2 datbase.

If the solutions from the Knowledge Base do not resolve your issue, a new ticket can be opened.

Do you want to create a new ticket?

If the entries from the knowledge base does not provide sufficient information, end users have option to open ticket.

Opening a ticket in Maximo/ICD system

As the last resort, the virtual agent can collect information and create a new ticket on your behalf. For example, if you ask

- how to program in Java

This is an area that the virtual agent has not been trained and it founds nothing in knowledge base. It then prompts you

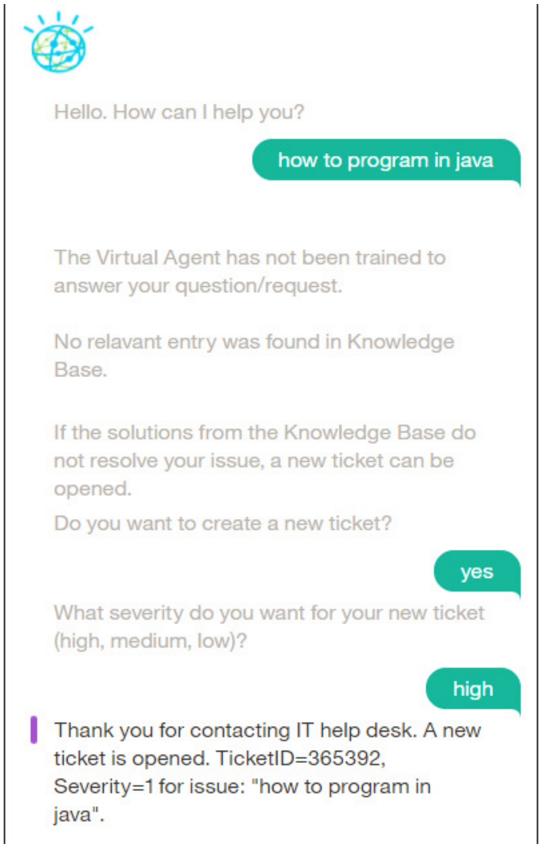
~~~
```bash
The Virtual Agent has not been trained to answer your question/request.
No relavant entry was found in Knowledge Base.

If the solutions from the Knowledge Base do not resolve your issue, a new ticket can be opened
Do you want to create a new ticket?
```
~~~

When you reply “Yes”, the virtual agent will ask

- What severity (high, medium and low) do you want for your new ticket?

After you specify the ticket severity (high, medium and low), the virtual agent opens a new ticket in your backend ticketing system.

- Thank you for contacting IT help desk. A new ticket is opened. `TicketID=365392, Severity=1 for issue: "how to program in java"`.



Hello. How can I help you?

how to program in java

The Virtual Agent has not been trained to answer your question/request.

No relavant entry was found in Knowledge Base.

If the solutions from the Knowledge Base do not resolve your issue, a new ticket can be opened.
Do you want to create a new ticket?

yes

What severity do you want for your new ticket (high, medium, low)?

high

Thank you for contacting IT help desk. A new ticket is opened. TicketID=365392, Severity=1 for issue: "how to program in java".

As the REST API is widely available, this app can be used to integrate Waston Assistant(Conversation) and Discovery service with most of back-office ticketing systems. Integrates with IBM Control Desk/Maximo is provided as an example in the code.

```
headers: {
            'Authorization': 'Basic bWF4aW1vOnJlbW90ZTE=',
            'Content-Type': 'application/json',
          },

          url: 'https://maximo-demo75.mro.com/meaweb/os/MXSR',
          body: '<?xml version="1.0" encoding="UTF-8"?><max:CreateMXSR xmlns:max="http://www.i
          method: 'POST'
        }
```

Exploring data of Conversation service

Below is the sample response JSON object from Watson Assistant(Conversation) service. Values of its intents, entities, input, output and context can be gathered and/or manipulated in Node.js code.

At the end of the JSON object, context.newticket and context.severity are related to the slot configurations in Conversation service.

```
{
   "intents":[
      {
         "intent":"greetings",
         "confidence":0.46840930583966855
      }
   ],
   "entities":[
      {
         "entity":"severity",
         "location":[
            0,
            1
         ],
         "value":"2",
         "confidence":1
      },
      {
         "entity":"sys-number",
         "location":[
            0,
            1
         ],
         "value":"2",
         "confidence":1,
         "metadata":{
            "numeric_value":2
         }
      }
   ],
   "input":{
      "text":"2"
   },
   "output":{
      "text":[
         "Thank you for contacting IT helpdesk. A new ticket is opened."
      ],
      "nodes_visited":[
         "slot_6_1516850647245",
         "node_1_1516850017677",
         "node_13_1516852865520"
      ],
      "log_messages":[

      ]
   },
   "context":{
      "conversation_id":"40a875f1-c8ef-4b63-9c69-661777bf3d71",
      "system":{
         "dialog_stack":[
            {
               "dialog_node":"node_13_1516852865520"
            }
         ],
         "dialog_turn_counter":17,
         "dialog_request_counter":17,
         "_node_output_map":{
            "Welcome":[
               0
            ],
            "node_3_1516832266395":[
               0
            ],
            "node_6_1516832414895":[
               0
            ],
            "node_5_1516850287208":[
               0
            ],
            "node_18_1517000905140":[
               0
            ],
            "node_13_1516852865520":[
               0
            ],
            "node_4_1516832287824":[
               0
            ]
         }
      },
      "newticket":true,
      "severity":2
   }
}
```

Modifying the app

After you have the app deployed and running, you can explore the source files and make changes. Try the following:

- Modify the .js files to change the app logic.
- Modify the .html file to change the appearance of the app page.
- Use the Conversation tool to train the service for new intents, or to modify the dialog flow. For more information, see the Conversation service documentation.

Deploying to Bluemix

You can use Cloud Foundry to deploy your local version of the app to Bluemix.

1. In the project root directory, open the `manifest.yml` file:

- In the `applications` section of the `manifest.yml` file, change the `name` value to a unique name for your version of the demo app.
- In the `services` section, specify the name of the Assistant(Conversation) service instance and Discovery service instance that you created. If you do not remember the service name, use the `cf services` command to list all services you have created.

The following example shows a modified `manifest.yml` file:

```
---
declared-services:
  my-conversation-service:
    label: conversation
    plan: free
  my-discovery-service:
    label: discovery
    plan: lite
applications:
- name: vaticketbot
  command: npm start
  path: .
  memory: 512M
  instances: 1
  services:
  - my-conversation-service
  - my-discovery-service
env:
  NPM_CONFIG_PRODUCTION: false
```

When you are ready to deploy the app to Bluemix environment,

1. In the Bluemix with the Cloud Foundry command-line tool, make sure you are in the correct folder where you downloaded the code pattern.
2. Push the app to Bluemix:

```
cf push
```

Access your app on Bluemix at the URL specified in the command output.

Troubleshooting

If you encounter a problem, you can check the logs for more information. To see the logs, run the `cf logs` command:

```
cf logs <application-name> --recent
```

License

This sample code is licensed under Apache 2.0.
Full license text is available in LICENSE.

Contributing

See CONTRIBUTING.

Open Source @ IBM

Find more open source projects on the
IBM Github Page.