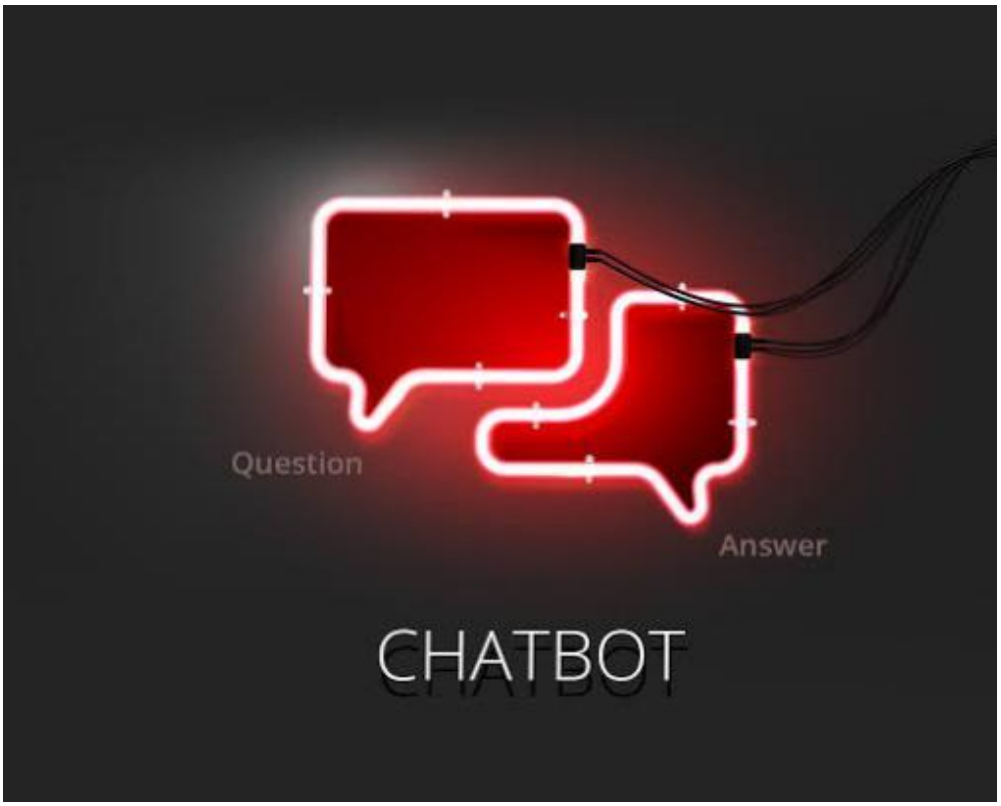
CLOUD APPLICATION DEVELOPMENT

CHATBOT DEPLOYMENT USING IBM CLOUD

BY
M.Ashwathy
V.Lakshmi Priya
V.abisha
M.nandhini
M.chandru



DEVELOPMENT PHASE - 3

Build a chatbot project by using Watson Assistant

Chatbot projects that use Watson Assistant involve three phases: scope, design, and integrate.

In the scope phase, you gather requirements for the conversation and how customers support the use case today. They might have a script, coded procedures, or other artifacts.

You define personas, create an empathy map, and build a system context diagram. Then, you extract the potential list of intents. Intents are the purposes or goals that are expressed in a user's input, such as answering a question or processing a payment. After you define intents, you assess the sentences that lead to those intents.

In the design phase, you create an instance of Watson Assistant and use its builder tool to define the intents and the entities. An entity represents a class of object or data type that is relevant to a user's purpose. At the end of the design phase, you start the dialog flow and unit-test it.

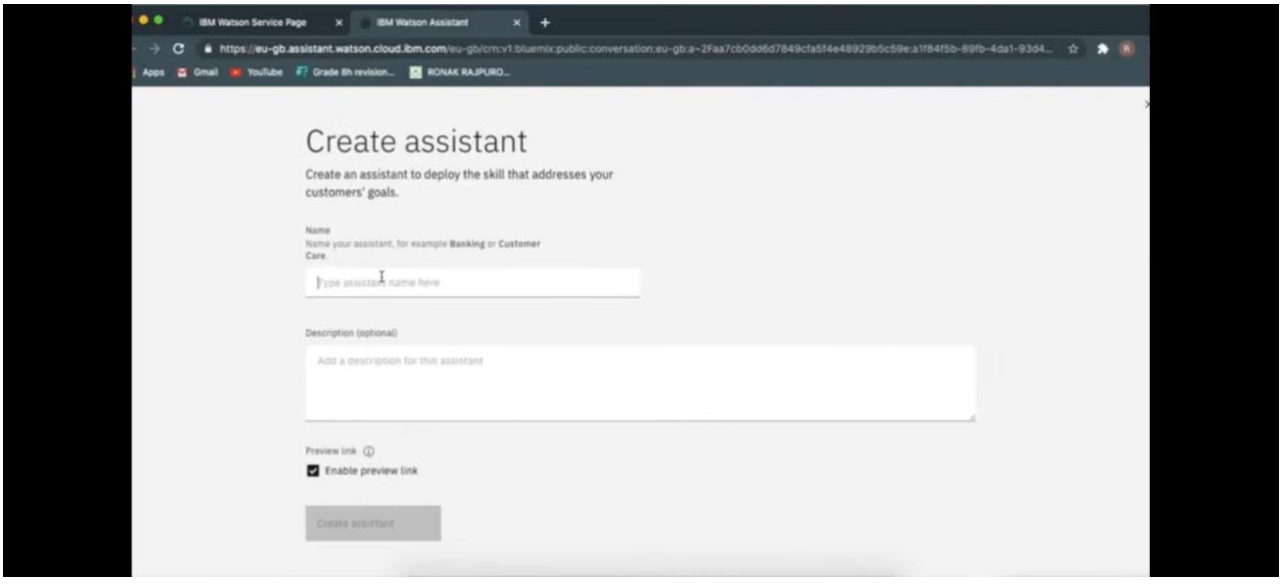
Finally, in the integrate phase, you develop the web app or microservice that interacts with Watson Assistant. You implement the business logic to handle the conversation context, and add other components to complement the business requirements, such as the IBM Watson Retrieve and Rank Service, ODM business rules, or IBM BPM process.

Scope	Design	Integrate
• Gather requirements for the conversation and how the customer is supporting the use case today • Define Persona • Build Empathy Map • Build System Context Diagram • Define channels • Define Intent and source of utterances	• Create a Watson Conversation service in Bluemix • Define intents • Define entities • Start the dialog flow, unit test it • Playback the conversation to end user	• Add more intents and Dialog flow • Develop the webapp or micro-service responsible to interact with Watson Conversation and implement some of the business logic to handle the conversation context • Add any other component to complement the business requirements, it could be Watson Discovery, ODM business rules, BPM process

Task 1: Create the Assistant service

The first task is to create an instance of Watson Assistant on IBM Cloud.

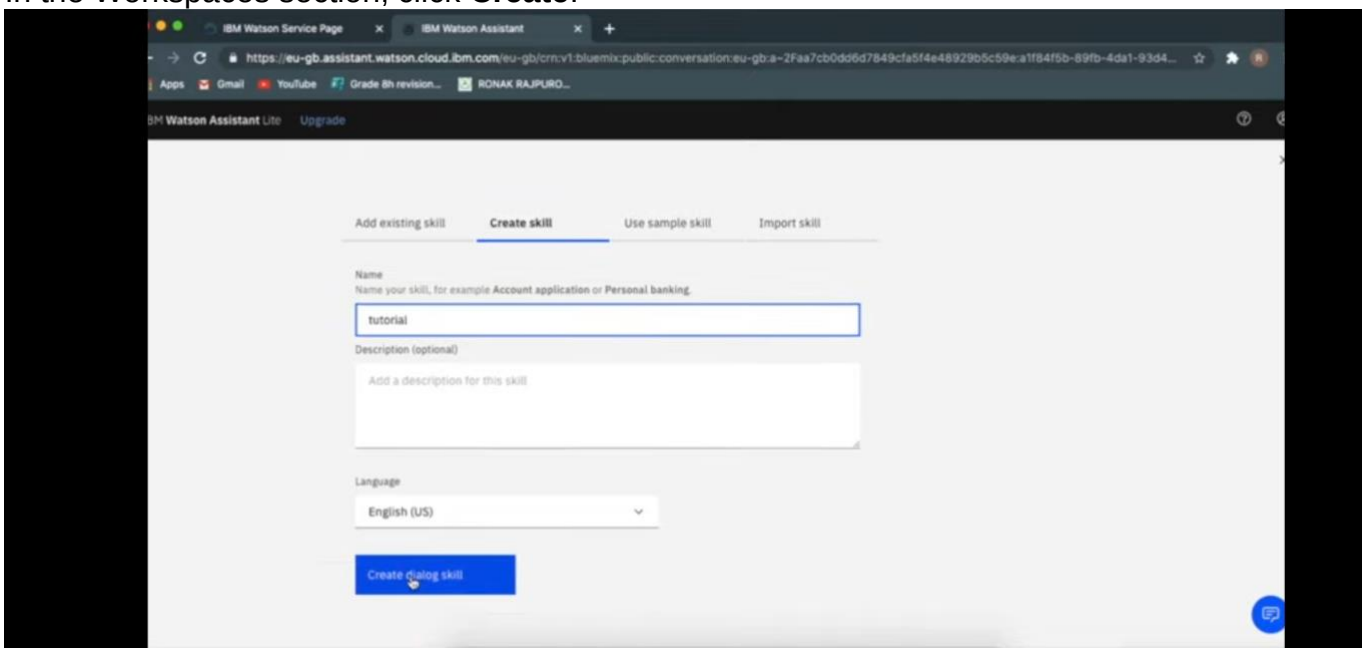
1. Make sure that you are logged in to your IBM Cloud account. Click **Catalog** and then click **Services > Watson > Assistant**.



Task 2: Create a workspace

You must use workspaces to maintain separate intents, user examples, entities, and dialog flows for each application. Watson Assistant uses a step-by-step approach to guide you to create workspace, intents, and so forth.

1. In the Workspaces section, click **Create**.



2. Type a name for the workspace. In the examples throughout this tutorial, the workspace name is ***SupportHelpDesk***.

Task 3: Create intents

Add intents. An ***intent*** is a group of examples of things that a user might say to communicate a specific goal or idea. To identify intents, start with something that a user might want and then list the ways that the user might describe it. For each intent, think of the various ways that a user might express his or her desire—those are the examples. Examples can be developed by using a crowdsourcing approach.
For example, in a discussion with the support team, you might gather this set of standard questions that support received from users:

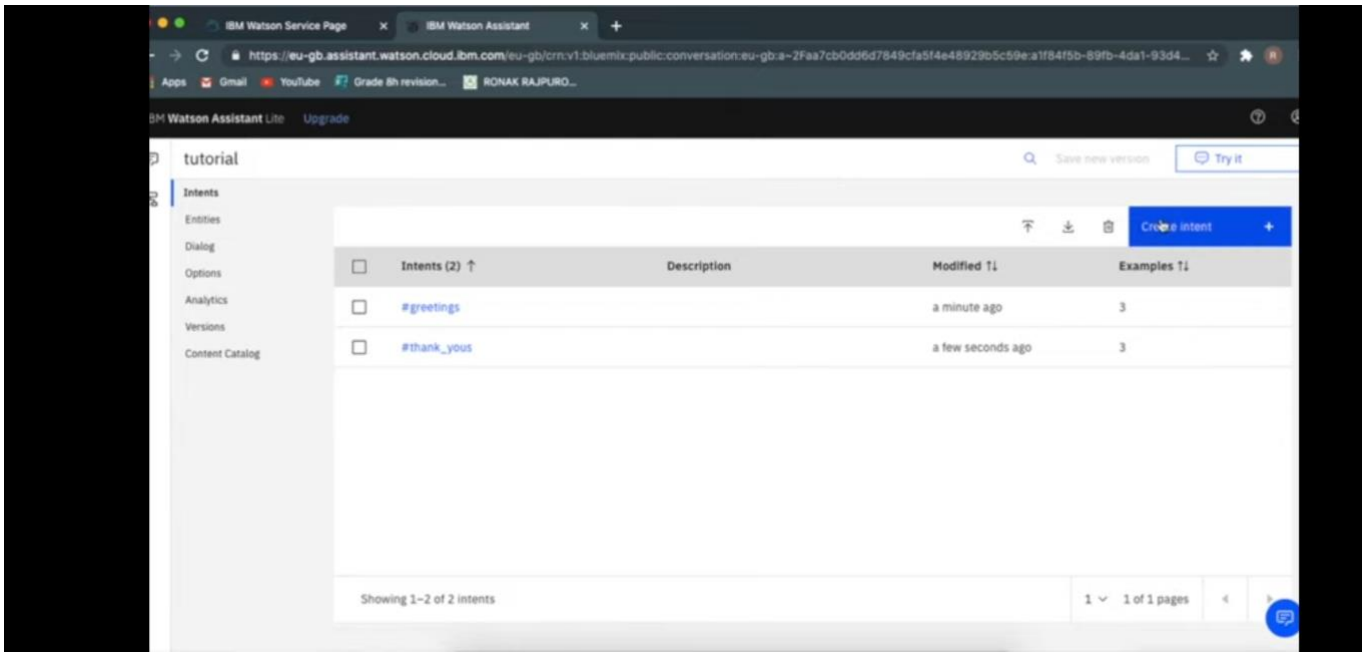
- What is the status of the business application? I could not access it.
- How to get access to a business application?
- How to reset my password for a specific application?
- When to renew my workstation?
- How to bring my own device and connect it to enterprise network?

Each of those questions is documented as a frequently asked question in the support team's document repository. Some solutions persist in a relational database in the form of application > problem > solution.

Based on the questions, you can extract these intents:

- Access to a business applications like expense report, AbC.
- Reset password
- Access to supplier on boarding business process
- Bring your own device

1. Add those intents to the workspace: From the Build page, click **Intents** and click **Create new**.



2. For the intent name, type greetings after the number sign (#).

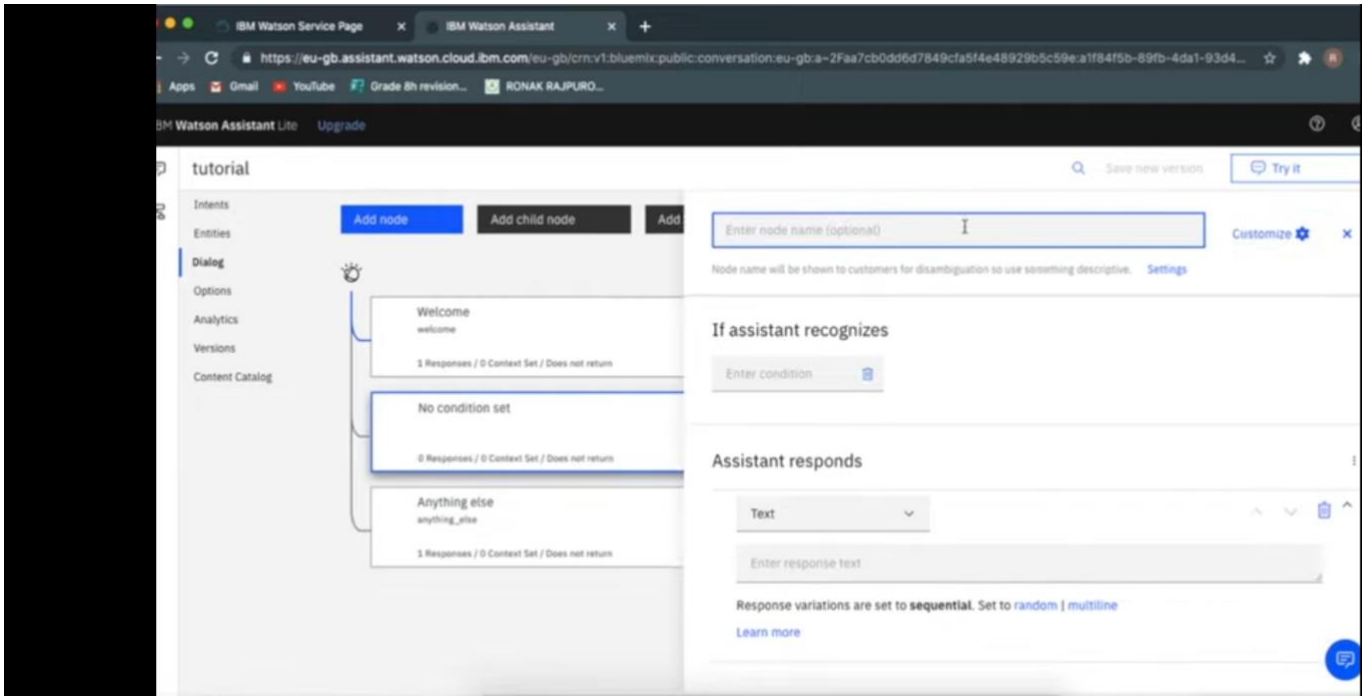
3. For each intent, add examples to train the conversation for intent recognition. You can enter the same examples as shown in the previous image.
4. Create the Goodbyes intent and add examples for it.

   Because many intents can be reused from conversation to conversation implementations, you can define .csv files and import them in the Conversation Tool Intents. The .csv format is shown in this example with one intent per line:

Task 4: Test the intents

Next, test your conversations.

1. As soon as you create an intent, you can test it by clicking **Ask Watson** icon in the top, right-hand side of the conversation editor.
2. Enter one of the examples. You should get the #greetings intent identified by Watson. Enter other greetings to test the #greetings intent.



CODE :

```
{
  "intents": [
    {
      "intent": "book-flight",
      "examples": [
        {
          "text": "book a flight"
        },
        {
          "text": "I need to buy a ticket"
        },
        {
          "text": "I want to take an airplane ride"
        },
        {
          "text": "I want to take a trip"
        }
      ],
      "description": ""
    },
    {
```

```
"intent": "check-flight-status",
"examples": [
 {
  "text": "Can I check the status of AA 123"
 },
 {
  "text": "Is my flight delayed"
 },
 {
  "text": "What is the status of AA 456"
 },
 {
  "text": "What is the status of my flight"
 },
 {
  "text": "What time does my flight depart"
 },
 {
  "text": "When does AA 123 arrive"
 },
 {
  "text": "when does my flight land"
 },
 {
  "text": "Where is my flight"
```

```
      }
    ],
    "description": ""
  }
 ],
 "entities": [
  {
    "entity": "airline",
    "values": [
     {
       "type": "synonyms",
       "value": "AA",
       "synonyms": [
         "American Airlines",
         "American Eagle"
       ]
     },
     {
       "type": "synonyms",
       "value": "DL",
       "synonyms": [
         "Delta"
       ]
     },
     {
```

```
  "type": "synonyms",
  "value": "NK",
  "synonyms": [
    "Spirit"
  ]
},
{
  "type": "synonyms",
  "value": "UA",
  "synonyms": [
    "United",
    "United Airlines"
  ]
},
{
  "type": "synonyms",
  "value": "VS",
  "synonyms": [
    "Virgin",
    "Virgin Atlantic"
  ]
},
{
  "type": "synonyms",
  "value": "WN",
```

```
      "synonyms": [
        "Southwest"
      ]
    }
  ],
  "fuzzy_match": true
},
{
  "entity": "airport",
  "values": [
    {
      "type": "synonyms",
      "value": "ATL",
      "synonyms": [
        "Atlanta",
        "Hartsfield-Jackson"
      ]
    },
    {
      "type": "synonyms",
      "value": "AUS",
      "synonyms": [
        "Austin",
        "Austin Bergstrom"
      ]
```

```
},
{
 "type": "synonyms",
 "value": "DAL",
 "synonyms": [
  "Dallas Love Field",
  "Love Field"
 ]
},
{
 "type": "synonyms",
 "value": "DFW",
 "synonyms": [
  "Dallas Fort Worth",
  "DFW"
 ]
},
{
 "type": "synonyms",
 "value": "LAX",
 "synonyms": [
  "LAX",
  "Los Angeles"
 ]
}
```

```
    ],
    "fuzzy_match": true
   },
   {
    "entity": "sys-number",
    "values": [],
    "fuzzy_match": true
   }
  ],
  "metadata": {
   "api_version": {
    "major_version": "v2",
    "minor_version": "2018-11-08"
   }
  },
  "webhooks": [
   {
    "url": "https://us-south.functions.appdomain.cloud/api/v1/web/your-url-here.json",
    "name": "main_webhook",
    "headers": []
   }
  ],
  "dialog_nodes": [
   {
    "type": "standard",
```

```
  "title": "Anything else",
  "output": {
    "generic": [
      {
        "values": [
          {
            "text": "I didn't understand. You can try rephrasing."
          },
          {
            "text": "Can you reword your statement? I'm not understanding."
          },
          {
            "text": "I didn't get your meaning."
          }
        ],
        "response_type": "text",
        "selection_policy": "sequential"
      }
    ]
  },
  "conditions": "anything_else",
  "dialog_node": "Anything else",
  "previous_sibling": "node_2_1578347066358",
  "disambiguation_opt_out": true
},
```

```
{
  "type": "event_handler",
  "output": {},
  "parent": "slot_1_1578503813667",
  "context": {
    "number": "@sys-number"
  },
  "conditions": "@sys-number",
  "event_name": "input",
  "dialog_node": "handler_3_1578503813672"
},
{
  "type": "event_handler",
  "output": {},
  "parent": "slot_2_1578503800481",
  "context": {
    "airline": "@airline"
  },
  "conditions": "@airline",
  "event_name": "input",
  "dialog_node": "handler_4_1578503800493"
},
{
  "type": "event_handler",
  "output": {
```

```
    "generic": [
     {
      "values": [
       {
        "text": "What is the flight number?"
       }
      ],
      "response_type": "text",
      "selection_policy": "sequential"
     }
    ]
   },
   "parent": "slot_1_1578503813667",
   "event_name": "focus",
   "dialog_node": "handler_5_1578503813672",
   "previous_sibling": "handler_3_1578503813672"
  },
  {
   "type": "event_handler",
   "output": {
    "generic": [
     {
      "values": [
       {
        "text": "What airline?"
```

```
        }
       ],
       "response_type": "text",
       "selection_policy": "sequential"
     }
   ]
  },
  "parent": "slot_2_1578503800481",
  "event_name": "focus",
  "dialog_node": "handler_9_1578503800493",
  "previous_sibling": "handler_4_1578503800493"
},
{
  "type": "standard",
  "title": "book-flight",
  "output": {
    "generic": [
     {
       "values": [
        {
          "text": "Which airline would you prefer?"
        }
       ],
       "response_type": "text",
       "selection_policy": "sequential"
```

```
      }
    ]
  },
  "conditions": "#book-flight",
  "dialog_node": "node_2_1578347066358",
  "previous_sibling": "node_6_1578503784131"
},
{
  "type": "frame",
  "title": "check-flight-status",
  "actions": [
    {
      "name": "main_webhook",
      "type": "webhook",
      "parameters": {
        "airline": "$airline",
        "flightNumber": "$number"
      },
      "result_variable": "webhook_result_1"
    }
  ],
  "metadata": {
    "_customization": {
      "mcr": true
    }
```

```
  },
  "conditions": "#check-flight-status",
  "dialog_node": "node_6_1578503784131",
  "previous_sibling": "Welcome"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "$webhook_result_1.message"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_6_1578503784131",
  "conditions": "$webhook_result_1.message",
  "dialog_node": "response_10_1642616790032"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Sorry an error has occured "
```

```
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_6_1578503784131",
  "conditions": "anything_else",
  "dialog_node": "response_8_1642616790428",
  "previous_sibling": "response_10_1642616790032"
},
{
  "type": "slot",
  "parent": "node_6_1578503784131",
  "variable": "$number",
  "dialog_node": "slot_1_1578503813667",
  "previous_sibling": "slot_2_1578503800481"
},
{
  "type": "slot",
  "parent": "node_6_1578503784131",
  "variable": "$airline",
  "dialog_node": "slot_2_1578503800481",
  "previous_sibling": "response_8_1642616790428"
},
{
  "type": "standard",
```

```
    "title": "Welcome",
    "output": {
      "generic": [
        {
          "values": [
            {
              "text": "Hello. How can I help you?"
            }
          ],
          "response_type": "text",
          "selection_policy": "sequential"
        }
      ]
    },
    "conditions": "welcome",
    "dialog_node": "Welcome"
  }
],
"counterexamples": [],
"system_settings": {
  "off_topic": {
    "enabled": true
  },
  "disambiguation": {
    "prompt": "Did you mean:",
```

```
    "enabled": true,
    "randomize": true,
    "max_suggestions": 5,
    "suggestion_text_policy": "title"
   },
   "human_agent_assist": {
    "prompt": "Did you mean:"
   },
   "spelling_auto_correct": true
  },
  "learning_opt_out": false,
  "name": "Webhook Demo - ML",
  "language": "en",
  "description": ""
}
```

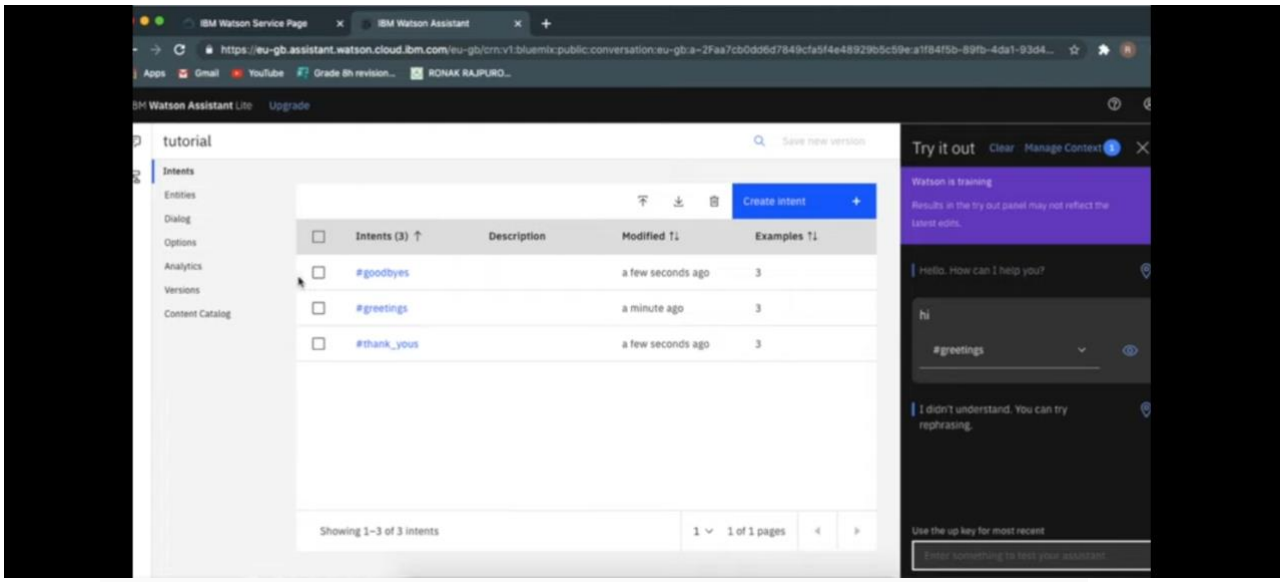

For view in github use link :

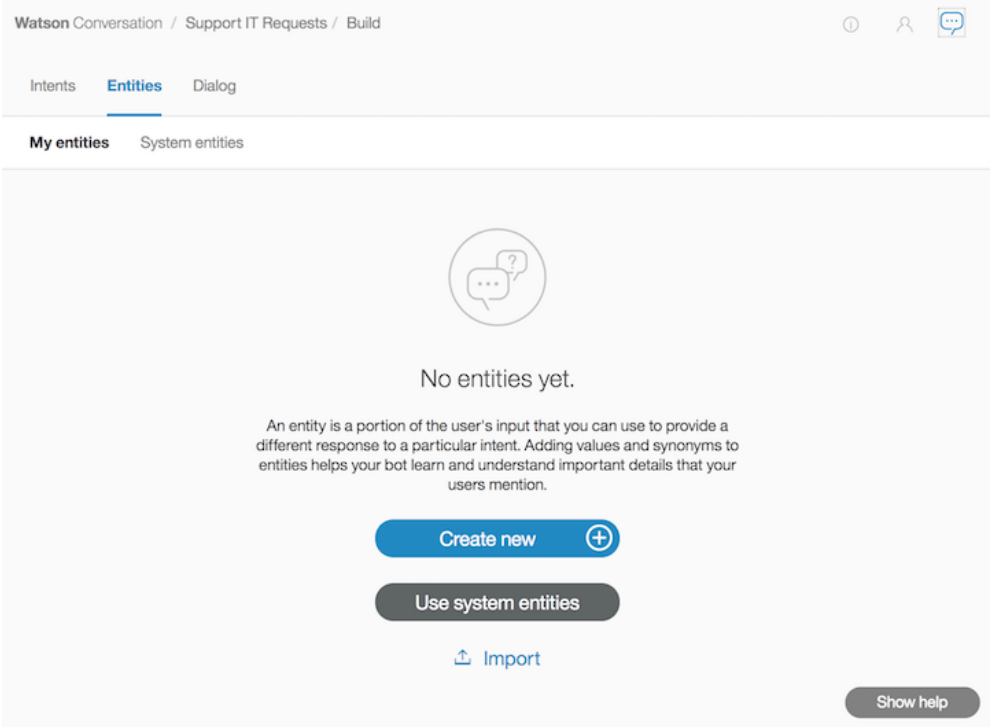
https://github.com/Ashwathy4209/Refactored-fishstick/blob/main/chatbot.json

Task 5: Add entities

An entity is a portion of the user's input that you can use to provide a different response to a particular intent.

- Click Entities. On the Entities page, click Create new.





Adding values and synonyms to entities helps your chatbot learn important details that your users might mention.
Each entity definition includes a set of specific entity values that can be used to trigger different responses. Each value can have multiple synonyms that define different ways that the same value can be specified in user input.

- Create entities to represent to the application what the user wants to access.

  Fuzzy logic is a feature that allows Watson Assistant to accept misspelled words. You can enable this feature at the entity level.
  As you did for intents, you can reuse entities' definitions through the export and import capabilities. Import the `wcs-workspace/chtatbot-Entities.csv` file.
- If you click the Ask Watson icon immediately after you import the entities, the `Watson is training` message is displayed. Watson Assistant classifies the entities. You can unit-test the entities by entering `I want to access application AbC`. The following figure shows both the intent and entity (@application:AbC) extracted by Watson Assistant:

You are now ready to create the dialog flow.

To be continued in phase 4 .