SMART RESTUARANT CHATBOT

Creating a Chatbot Using IBM Watson Assistant

Student:

Sairam Mandapaka

Teacher:

Smartbridge

Course:

Smart restaurant Bot

### **Build a chatbot project by using Watson Assistant**

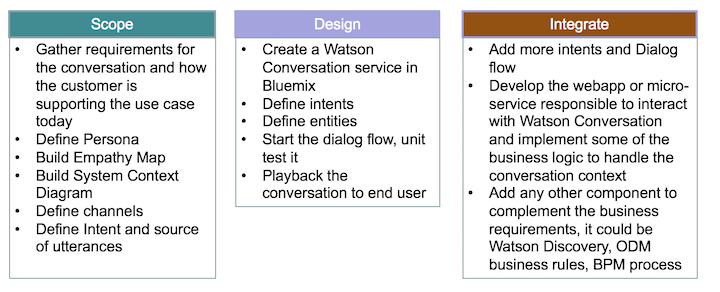
Chatbot projects that use Watson Assistant involve three phases: scope, design, and integrate.In the scope phase, you gather requirements for the conversation and how customers support the use case today. They might have a script, coded procedures, or other artifacts.

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.

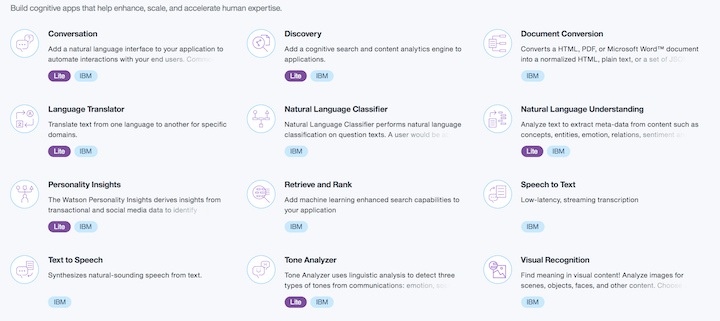


When you implement a cognitive solution, be sure to apply to develop an innovative business-impact application.

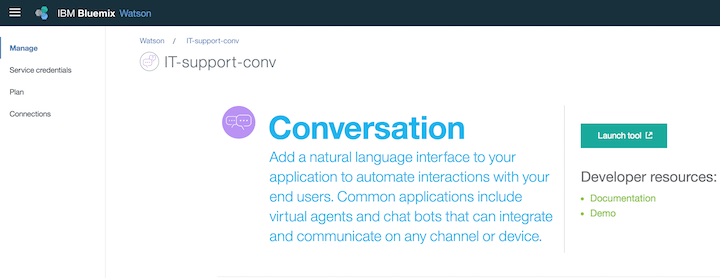
### **[Task 1: Create the Assistant service](https://www.ibm.com/cloud/architecture/tutorials/watson_conversation_support?task=1/)**

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**.

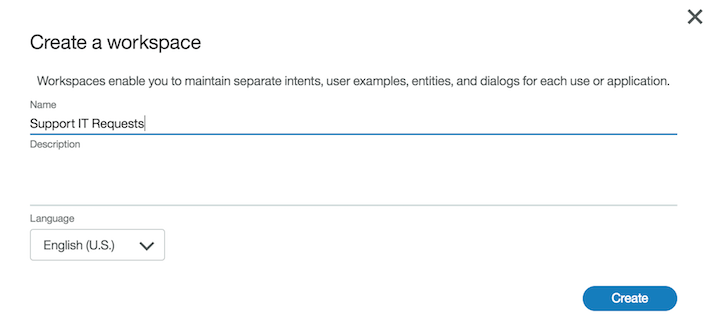


1. For the service name, type ITSupportConversation. Click **Create**. This page is displayed:



1. Click **Launch tool** to open the Watson Assistant workspace.
2. 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.

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



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

1. 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.
2. Add those intents to the workspace: From the Build page, click **Intents** and click **Create new**.
3. For the intent name, type applicationAccess after the number sign (#).
4. For each intent, add examples to train the conversation for intent recognition. You can enter the same examples as shown in the previous image.
5. Create the Goodbyes intent and add examples for it.

**[Task 2: Create a workspace And Create intents](https://www.ibm.com/cloud/architecture/tutorials/watson_conversation_support?task=2/)**

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.

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

1. Click **Entities**. On the Entities page, click **Create new**.
2. Create entities to represent to the application what the user wants to access.
3. 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.
4. After you specify your intents and entities, you can construct the dialog flow

The first node addresses greetings in a response to a query such as "hello." Click the welcome node and click **Add node below**: A new node is added between the welcome and "Anything else" nodes.

At each node level, you can expand the conversation by adding nodes. If you add nodes at the same level, the flows are parallel. Adding a child node creates a dependent track of conversation, and the conversation branches out into a tree structure.

1. Name the new node Handle Greetings. In the **If bot recognizes** field, change the value to #Greetings. The number sign (#) represents a prefix for intent. The condition is triggered when the Watson natural language classifier classifies the query as a greeting intent.
2. Unit-test your dialog by clicking the **Ask Watson** icon.

**[Task 7: Complete advanced dialog work](https://www.ibm.com/cloud/architecture/tutorials/watson_conversation_support?task=7/)**

Each time the dialog returns a response and waits for user input, the dialog stores the ID of the node where the conversation must resume. This node is called the contextual node. Its ID is added to the context.system.dialog pack property, which contains a JSON array of dialog node IDs that are on the dialog stack.

### **The context object**

State information for your conversation is maintained by using the context object. The context object is a JSON object that is passed between your application and the Watson Assistant service.

### **Use slots**

Add a dialog flow to address when a user wants to bring his or her own device. Only certain brands and devices are supported, so you must determine the brand and type of device.

**[Use the API](https://www.ibm.com/cloud/architecture/tutorials/watson_conversation_support?task=8/)**

To use the API, you need the service credentials and the tool to perform an HTTP request. For detailed instructions, see Watson assistant API.