

# **B2B SaaS Platform Bot**

*A*

*Project Report*

*submitted in partial fulfillment of the  
requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING**

By

**1NT17CS043**

**Bhavya V**

**1NT17CS168**

**Sangeetha S Chegaraddi**

*Under the guidance of*

**Mr. Manjit Sodhi**

**IT Architect**

**IBM**





## **CANDIDATE'S DECLARATION**

We here by certify that the project work entitled “**B2B SaaS Platform Bot**” in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING, Nitte Meenakshi Institute of Technology, Bangalore, is an authentic record of our work carried out during a period from August,2020 to September,2020 under the supervision of Mr. Manjit Sodhi

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

**1NT17CS043**

**Bhavya V**

**1NT17CS168**

**Sangeetha S Chegaraddi**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: November 3,2020

**Project Guide- Mr. Manjit Sodhi**



## Nitte Meenakshi Institute of Technology, Bangalore

PROJECT TITLE: B2B SaaS Platform Bot

---

### ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Manjit Sodhi**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected HoD of the Department, **Dr. Thippeswamy M.N.**

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

## **ABSTRACT**

Chatbots, which are often called virtual assistants, are built to conduct specific tasks or provide information. These softwares act as a human agent by responding to users, informing, collecting information, etc. and hence providing better customer satisfaction. The functionality includes both internal, employee-facing requests, and external, customer-facing requests. Chatbots allow users to interact with an application conversationally, whether textually or audibly, to perform certain tasks. Chabots acting as a virtual assistant have to think and act intelligently as a human and also with an understanding of human language to interact and hence in this modern technological world, there is a need for an inbuilt Artificial Intelligence and Natural Language Processing Chatbot making the bots more efficient.

# TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>1 Introduction</b>  | <b>6</b>  |
| <b>2 Related Work</b>  | <b>7</b>  |
| (List pros and cons of various algorithm studied in the project)                                   |           |
| <b>3 Problem Statement</b>   | <b>9</b>  |
| <b>4 Objective</b>   | <b>10</b> |
| <b>5 Design</b>  | <b>11</b> |
| (Include Methodology)  |           |
| (Steps to Create Watson Assistant)   |           |
| (How it Works ?)   |           |
| <ul style="list-style-type: none"><li>• Implementation Steps</li><li>• Cloud Functions</li></ul>   |           |
| <b>6 Implementation</b>  | <b>25</b> |
| (Include GitHub link having following)   |           |
| <ul style="list-style-type: none"><li>• Project Source Code</li><li>• Project Demo Video</li></ul> |           |
| <b>7 Conclusion and Future Scope</b>   | <b>26</b> |
| <br>   |           |
| <b>References</b>  |           |
| <ul style="list-style-type: none"><li>• Website URLs, PDF's, Books etc.</li></ul>                  |           |

## **INTRODUCTION**

Chatbots were developed to create a lively chat conversation in text or speech in place of a human agent. These chatbots require continuous tuning and testing to stand up to the company standard conversation by a software. Chatbots are used in dialog systems for various purposes including customer service, information gathering, request routing, etc. Chatbots have been a successful key in many of the industries including marketing, finance, medicine, business, etc. Chatbots were primarily functioning with the assistance of the scripted conversations by identifying the keywords, which has been changed as the technology evolved. There is continuous development happening in order to create these bots to provide the maximum customer satisfaction by making them think and answer like a human agent. Hence various technologies like Artificial Intelligence, Machine learning, etc. are implemented in these bots and also provide platforms to create, build, train and deploy our own chat bots and one such platform is IBM cloud.

IBM cloud offers Software as a service (SaaS) which includes IBM's AI assistant for business also known as Watson Assistant to build, train and deploy easily into any application, device or a channel. Watson assistant can be deployed in any cloud or on-premises environment. The Watson Assistant is developed to make the customer experience easy and joyable in every interaction of the chat interface, powered by NLU (Natural Language Understanding ) with the help of Artificial Intelligence Technology, and connect to any application or channels, etc. The IBM Watson Assistant allows you to integrate with digital tools like Zendesk and Intercom and connect to various voice platforms for speech recognition and text to speech translation.

With all the features available with IBM Watson Assistant, it's easy for any developer to create an Assistant easily when compared to other platforms and integrate it with any application. These assistants can also be useful for any industries including Business to Business and making the job easier.

### **How Chatbots Work ?**

The Chatbots usually start their conversation by introducing themselves as a virtual assistant and would ask the customers if they need any help to be dealt with.

The Customer or the user will either type the query or choose the options provided by the assistant. The Virtual Assistant will respond with respect to the query or the option chosen and if the query does not fit in any of the queries as trained the Virtual Assistant would request the user to mail to the corresponding company or would connect to the human agent if it is necessary as trained. So, the chatbots respond as they have trained and would keep learning continuously to achieve the maximum accuracy possible.

## **Related Work**

### **Pros and cons in using Chatbot**

#### **Pros-**

1. **Faster Customer Service:** There's no question that bots are able to save your team valuable time. Since bots can be programmed to answer simple questions, your staff can focus on the more complex issues that your customers may have.
2. **Constantly Available:** One of the advantages of bots is that, unlike humans, bots can respond to customer inquiries around the clock without costing you extra. With 24/7 chat capabilities, your chatbot can answer the questions of the customer instantly, without requiring them to call your service team between working hours.
3. **Promotes Conversational Marketing:** Most consumers prefer to receive an answer to their questions immediately. Rather than having to scroll through the website and find the answer themselves, or fill out a contact form to receive a call or email at a later date, a chatbot allows customers to engage with the site without requiring an actual human to monitor the conversations.
4. **Helps Manage Customer Requests:** If you sometimes have many thousands of customer inquiries that you simply find are generally simple to reply to, a chatbot can help. There's no ceiling to what percentage of customers a chatbot can handle, which allows your service team to specialize in the more complex requests which will come their way.

#### **Cons-**

1. **Language limitations.** The natural language and conversation features are not multi-lingual to date, which limits your target market.
2. **Slow integration.** It takes time to integrate Watson Commerce technology, which makes it slow to market if this is your first ecommerce platform. It's also difficult and expensive to switch to Watson Commerce.
3. **Steep learning curve.** This applies more to the machine than to your staff. To utilize Watson technology to its fullest potential, it takes time for the platform to gather sufficient data. This also takes effort on behalf of your IT team.
4. **Site maintenance.** As you can imagine, a platform that can handle the volume of data required for AI has an increased need for maintenance and upgrades. This will require an IT team intimately familiar with the IBM Watson technology, who can quickly identify and fix issues, as well as make platform enhancements.
5. **Increasing rate of data, with limited resources.**

6. **Higher Capacity for Misunderstanding:** When a customer's question isn't clear or is just too specific, a bot may have a tough time helping, which is one of the most important disadvantages of chatbots. Chatbots are programmed to answer general questions with answers which will be found in its database, so if a customer asks something outside of this narrow list of answerable questions, they're going to likely confuse the bot and can either be taken around in a circle because the bot tries to know the question being asked (often to no avail), or just be left without a solution. In either case, this isn't an excellent customer experience and may negatively impact your company's image.
7. **Need to be Maintained:** Chatbots need frequent optimization to form sure they're giving the proper information to your customers and representing your brand appropriately. As your user demand and business priorities shift, you'll get to update your bot accordingly, which needs analysis of previous bot conversations to spot common questions your bot receives. If you allow your bot to its own devices, your customers will notice and your customer satisfaction rating will suffer. Additionally, chatbots are programmed to handle a selected amount of knowledge, and as you update and edit the info, there are often disruptions to the chatbot model as an entire. This needs ongoing and careful maintenance to form sure you don't create holes within the chatbot interface.
8. **Limited Functionality:** Chatbots were created to reply to simple questions that will be answered with facts. Chatbots have limited responses, so they're rarely ready to answer multi-part questions or questions that need decisions. This often means your customers are left without an answer and need to undergo more steps to contact your support team.



## **PROBLEM STATEMENT**

To Build an AI inbuilt Chatbot for Business to Business Transactions using IBM Cloud Platform to build, train, test and deploy using web chat or in a link preview. The User can select any of the options provided by the bot and get the corresponding replies by the bot. The Bot can provide information, collect information, or if necessary provide the administrative details for the user to contact. The Chatbot must be able to collect information and do the necessary or provide instructions or just respond accordingly. The Bot must be able to understand and process human language (Natural Language Processing) and the bot should be designed to give options for the user to just select or click to avoid the unnecessary typing.

## **OBJECTIVES**

Main objectives of this project are:

- To respond with appropriate answers as if it was responded by a human agent.
- Avoiding miscommunication such as vulgarity
- Customer Satisfaction
- Data Security
- Accessibility, Maintenance, Huge number of requests, etc.

# Design

## METHODOLOGY

There are three primary actors in the system:

1. Virtual Assistant - Chatbot
2. User or Customer
3. Administration

Each of them performs following tasks:

### **1. Virtual Assistant**

- Respond to the queries or options selected by the user as it was trained during the development of it.
- To keep learning from the mistakes and correct the model and train itself again to achieve the most accuracy.
- If not sure about the answer, to respond with the administration email for the user to contact.

### **2. User/ Customer**

- User just takes help from the chatbot to resolve his/her problems.
- To provide feedback to the company regarding chatbot performance.

### **3. Administration**

- To make necessary changes and develop chatbot accordingly for Customer satisfaction.
- To get in contact with customers if necessary when chatbot couldn't resolve customer issues.
- Maintenance of the chatbot.

## Steps to Create a Virtual Assistant

1. Go to the [Watson Assistant](#) page in the IBM Cloud® catalog.

The service instance will be created in the default resource group if you do not choose a different one, and it cannot be changed later. This group is sufficient for the purposes of trying out the product.

2. Sign up for a free IBM Cloud account or log in.
3. Click Create.

### **Step 1: Open Watson Assistant**

After you create a Watson Assistant service instance, you land on the Manage page.

1. Click Launch Watson Assistant. If you're prompted to log in, provide your IBM Cloud credentials.

A new browser tab or window opens and Watson Assistant is displayed.

- An assistant named My first assistant is created for you automatically. An assistant is a chatbot. You add skills to your assistant so it can interact with your customers in useful ways.
- A dialog skill named My first skill is added to the assistant for you automatically. A dialog skill is a container for the artifacts that define the flow of conversations that your assistant has with your customers.
- The dialog skill is opened and the Intents page is displayed.

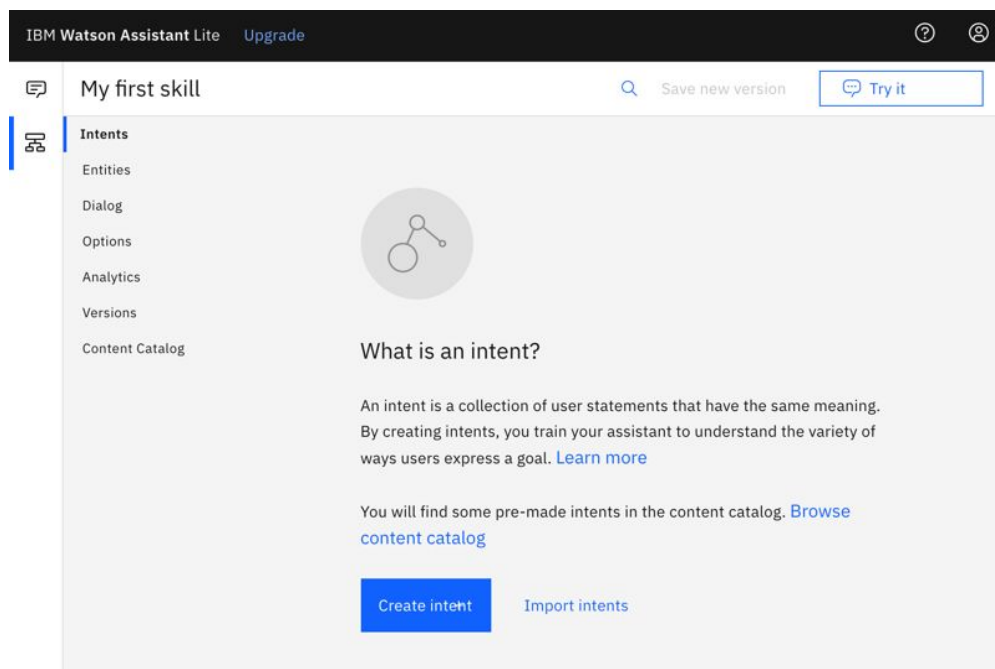


figure 1: Creating a skill in Watson Assistant

If available in your location, a tour begins that you can step through to learn about the product. Follow the tour; it provides a great overview of the product.

If an assistant and skill are not created automatically, complete Steps 2 and 3. Otherwise, skip to Step 4: Add intents from a content catalog.

## Step 2: Create an Assistant

An assistant is a cognitive bot to which you add skills that enable it to interact with your customers in useful ways.

1. Click the Assistants icon (Assistants menu icon), and then click Create assistant.

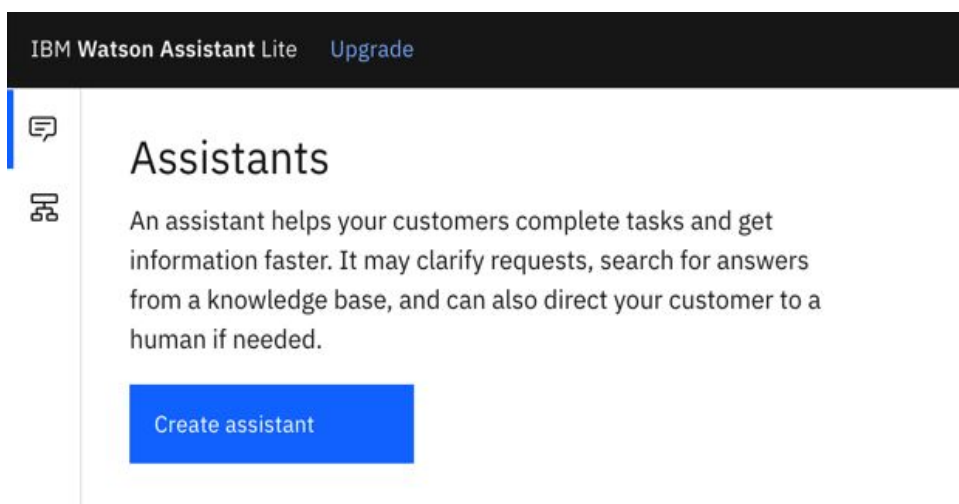


figure 2.1: Creating Assistant in Watson Assistant for the skill created

2. Name the assistant My first assistant.
3. Click Create assistant.

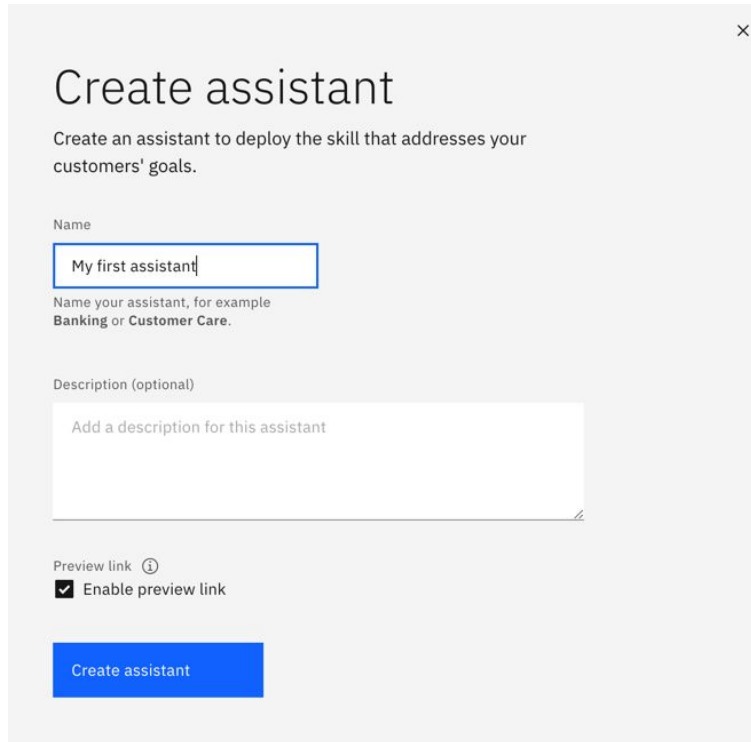


figure 2.2: Creating Assistant in Watson Assistant for the skill created

### Step 3: Create a dialog skill

A dialog skill is a container for the artifacts that define the flow of a conversation that your assistant can have with your customers.

1. Click the My first assistant tile to open the assistant.
2. Click Add dialog skill.

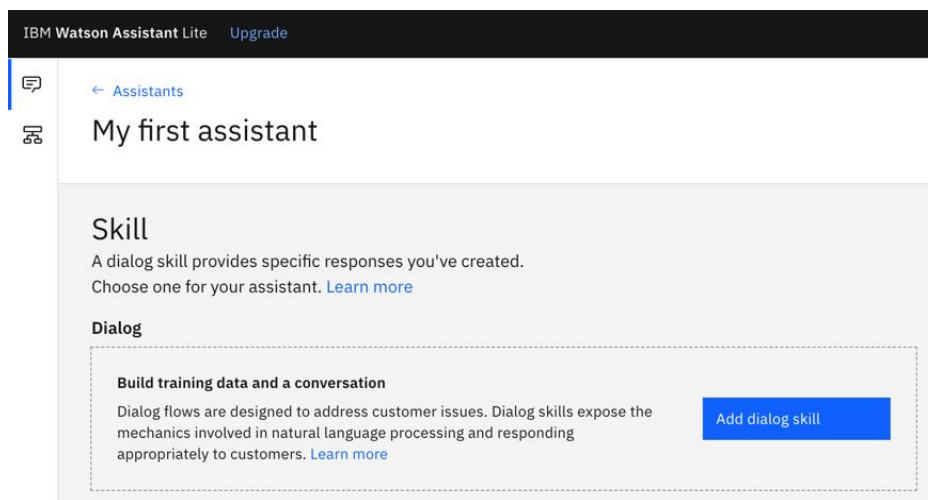


figure 3.1: Creating Dialog system

3. Give your skill the name My first skill.
4. Optional. If the dialog you plan to build will use a language other than English, then choose the appropriate language from the list.

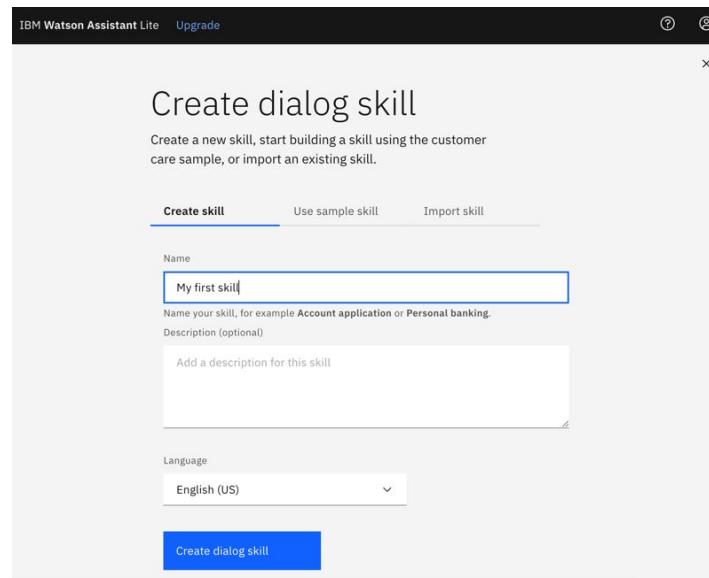


figure 3.2: Creating Dialog system

5. Click Create dialog skill.

The skill is created and opens to the Intents page.

#### Step 4: Add intents from a content catalog

The Intents page is where you start to train your assistant. In this tutorial, you will add training data that was built by IBM to your skill. Prebuilt intents are available from the content catalog. You will give your assistant access to the General content catalog so your dialog can greet users, and end conversations with them.

1. Click Content Catalog from the Skills menu.
2. Find General in the list, and then click Add to skill.

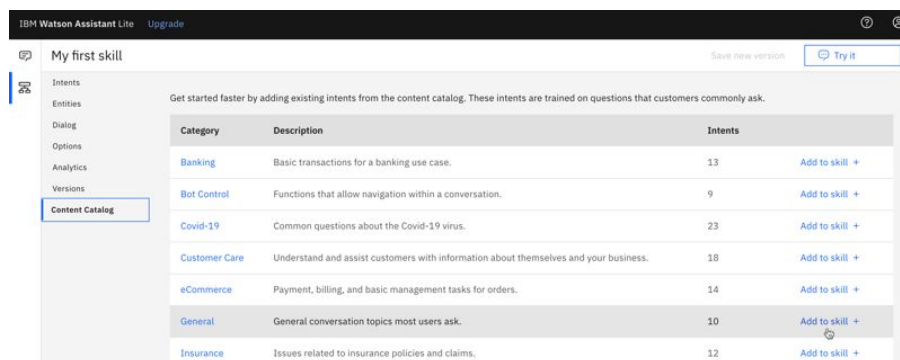
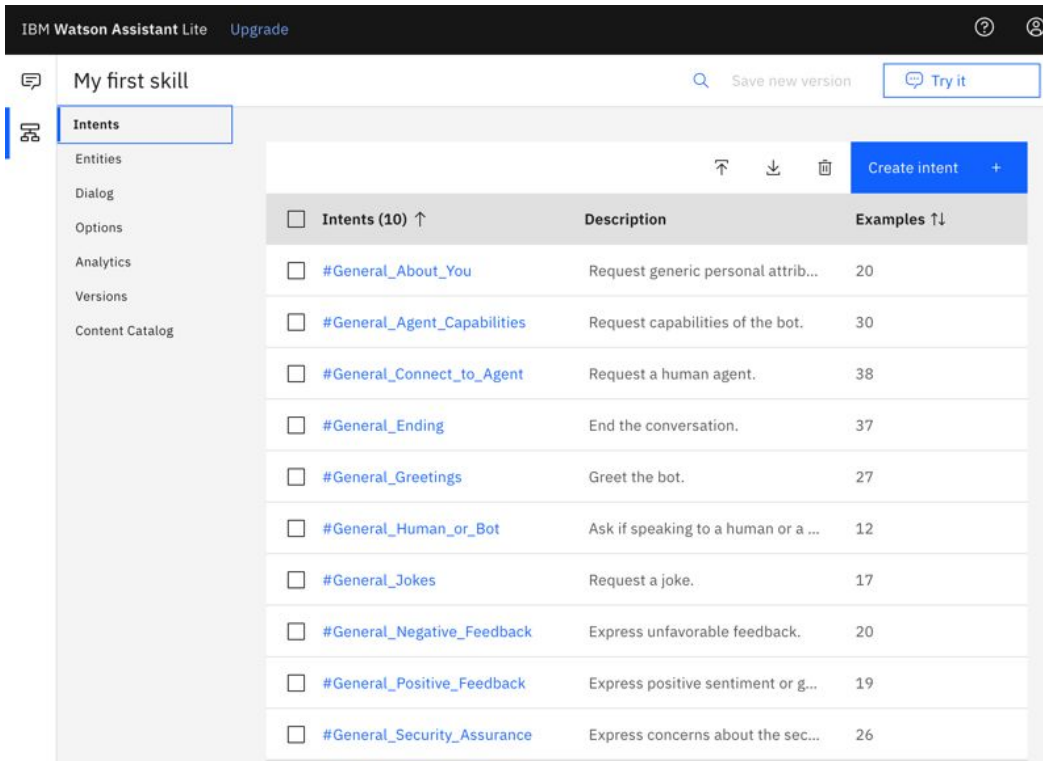


figure 4: Adding Required intents into our skill

Open the Intents tab to review the intents and associated example utterances that were added to your training data. You can recognize them because each intent name begins with the prefix #General\_. You will add the #General\_Greetings and #General\_Ending intents to your dialog in the next step.



| <input type="checkbox"/> | Intents (10) ↑              | Description                         | Examples ↓ |
|--------------------------|-----------------------------|-------------------------------------|------------|
| <input type="checkbox"/> | #General_About_You          | Request generic personal attrib...  | 20         |
| <input type="checkbox"/> | #General_Agent_Capabilities | Request capabilities of the bot.    | 30         |
| <input type="checkbox"/> | #General_Connect_to_Agent   | Request a human agent.              | 38         |
| <input type="checkbox"/> | #General_Ending             | End the conversation.               | 37         |
| <input type="checkbox"/> | #General_Greetings          | Greet the bot.                      | 27         |
| <input type="checkbox"/> | #General_Human_or_Bot       | Ask if speaking to a human or a ... | 12         |
| <input type="checkbox"/> | #General_Jokes              | Request a joke.                     | 17         |
| <input type="checkbox"/> | #General_Negative_Feedback  | Express unfavorable feedback.       | 20         |
| <input type="checkbox"/> | #General_Positive_Feedback  | Express positive sentiment or g...  | 19         |
| <input type="checkbox"/> | #General_Security_Assurance | Express concerns about the sec...   | 26         |

Figure 5: List of Intents in the skill

### Step 5: Build a dialog

A dialog defines the flow of your conversation in the form of a logic tree. It matches intents (what users say) to responses (what your virtual assistant says back). Each node of the tree has a condition that triggers it, based on user input.

We'll create a simple dialog that handles greeting and ending intents, each with a single node.

#### Adding a start node

1. From the Skills menu, click Dialog.

The following two dialog nodes are created for you automatically:

Welcome: Contains a greeting that is displayed to your users when they first engage with the assistant.

Anything else: Contains phrases that are used to reply to users when their input is not recognized.



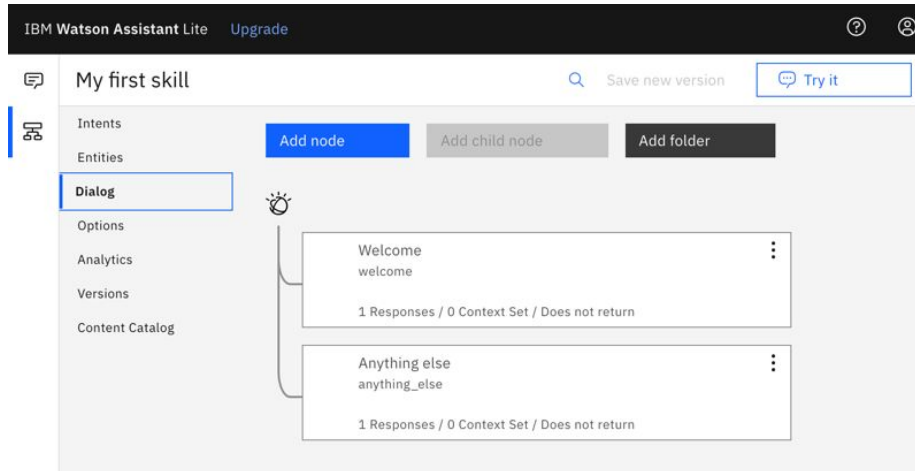


Figure 6.1: Creating Dialog System

2. Click the Welcome node to open it in the edit view.
3. Replace the default response with the text, Welcome to the Watson Assistant tutorial!.

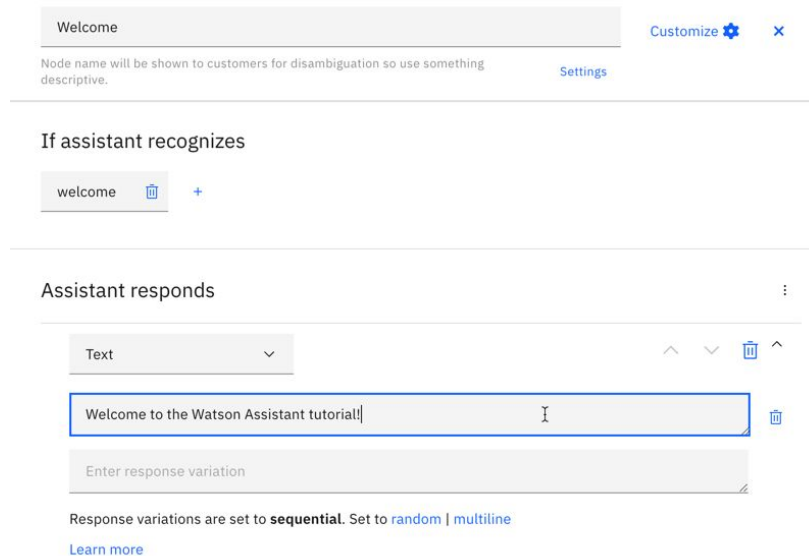


Figure 6.2: Creating Dialog System

### Testing the start node

You can test your dialog at any time to verify the dialog. Let's test it now.

Click the Try it icon to open the "Try it out" pane. You should see your welcome message.

## Adding nodes to handle intents

Now let's add nodes between the Welcome node and the Anything else node that handle our intents.

1. Click Add node.
2. In the node name field, type Greet customers.
3. In the If assistant recognizes field of this node, start to type #General\_Greetings. Then, select the #General\_Greetings option.
4. Add the response text, Good day to you!
5. Add the dialog boxes and create the conversation as required and train the model.

## Testing intent recognition

You built a simple dialog to recognize and respond to both greeting and ending inputs. Let's see how well it works.

1. Click the Try it icon to open the "Try it out" pane. There's that reassuring welcome message.
2. In the text field, type Hello and then press Enter. The output indicates that the #General\_Greetings intent was recognized, and the appropriate response (Good day to you.) is displayed.

## Step 6: Integrate the assistant

Now that you have an assistant that can participate in a simple conversational exchange, test it.

1. Click the Assistants icon Assistants menu icon to open a list of your assistants.
2. Find the My first assistant assistant, and open it.
3. Test your assistant with a Preview link integration.

The Preview link integration is created for your automatically. It builds your assistant into a chat widget that is hosted by an IBM-branded web page. You can open the web page and chat with your assistant to test it out.

4. From the Integrations section, click the Preview link tile.
5. Click the URL that is displayed on the page.

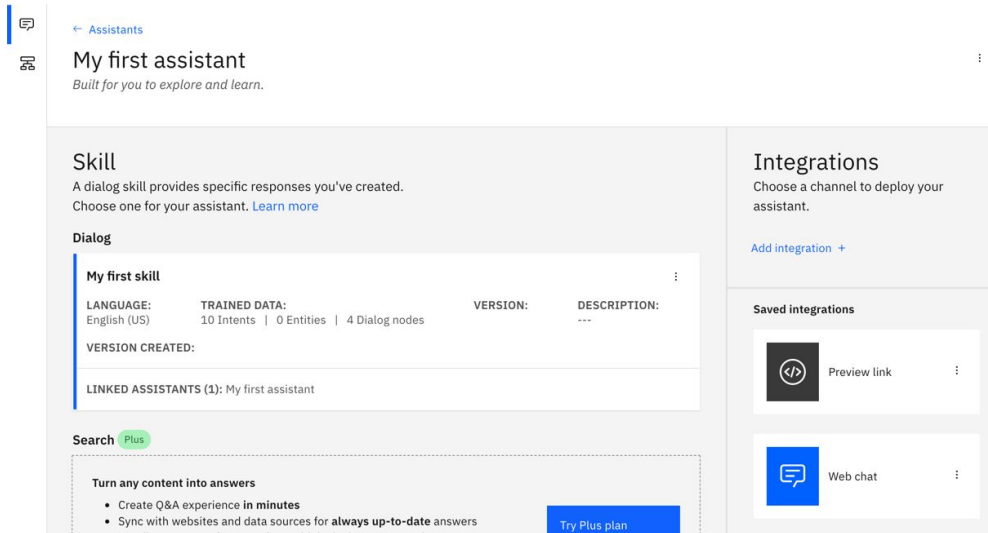


Figure 7: Integrating the assistant

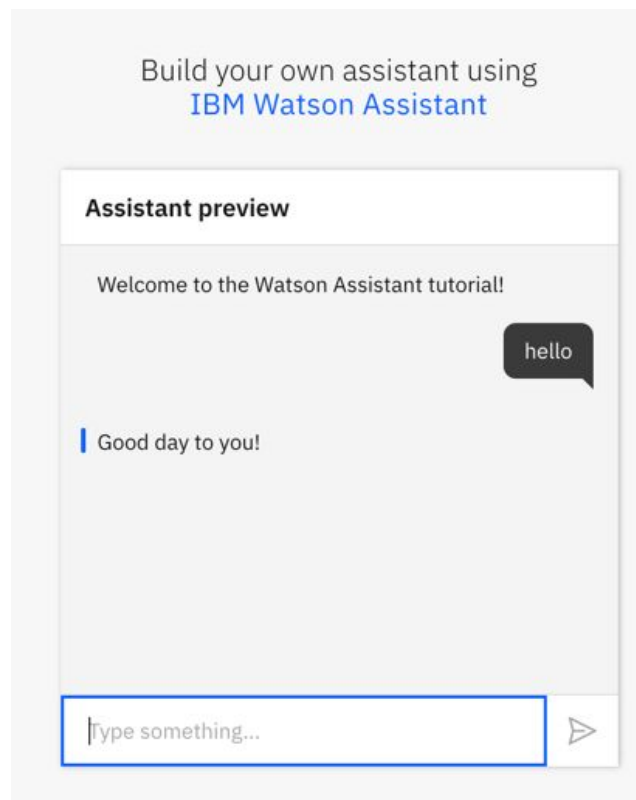


Figure 8: Sample Web Chat Integration

You can share the URL with others who might want to try out your assistant.

6. After testing, close the web page. Click the X to close the preview link integration page.

## How it Works ?

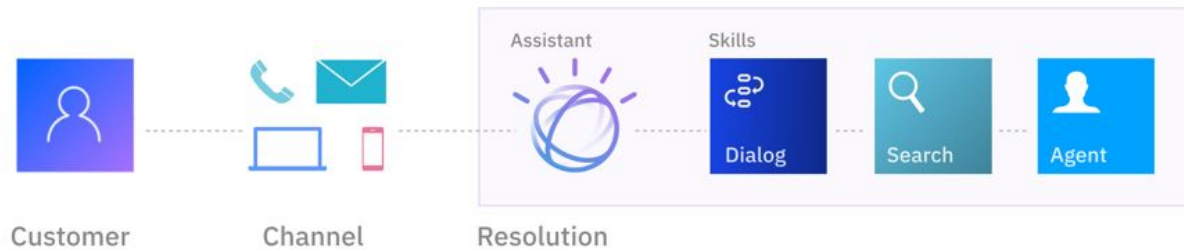


Figure 9: product delivers an omnichannel customer experience

1. Users interact with the assistant through one or more of these integration points:
  - A virtual assistant that you publish directly to an existing social media messaging platform, such as Slack or Facebook Messenger.
  - A web chat that you embed in your company website that can answer customer questions directly and can transfer complex requests to a customer support representative.
  - A custom application that you develop, such as a mobile app or a robot with a voice interface.
2. The assistant receives user input and routes it to the dialog skill.
3. The dialog skill interprets the user input further, then directs the flow of the conversation. The dialog gathers any information it needs to respond or perform a transaction on the user's behalf.
4. Any questions that cannot be answered by the dialog skill are sent to the search skill, which finds relevant answers by searching the company knowledge bases that you configure for the purpose.

## Implementation Steps

1. Create an assistant.
2. Add a skill to your assistant.

Depending on your service plan, you can add the following types of skills:

- To create an AI-driven conversational flow, add a dialog skill.

Use the intuitive graphical product to define the training data and dialog for the conversation between your assistant and your customers. The training data consists of the following artifacts:

**Intents:** Goals that you anticipate your users have when they interact with your assistant. Define one intent for each goal that can be identified in a user's input. For example, you might define an intent that is named `store_hours` that answers questions about store hours.

**Dialog:** Use the dialog editor to build a dialog flow that incorporates your intents. The dialog flow is represented graphically as a tree. You can add a branch to process each of the intents that you want your assistant to handle.

**Entities:** An entity represents a term or object that provides context for an intent. For example, an entity might be a city name that helps your dialog to distinguish which store the user wants to know store hours for. After you add entities, update your dialog to use them. Add dialog nodes that handle the many possible permutations of a request based on the entities that are found in the user input.

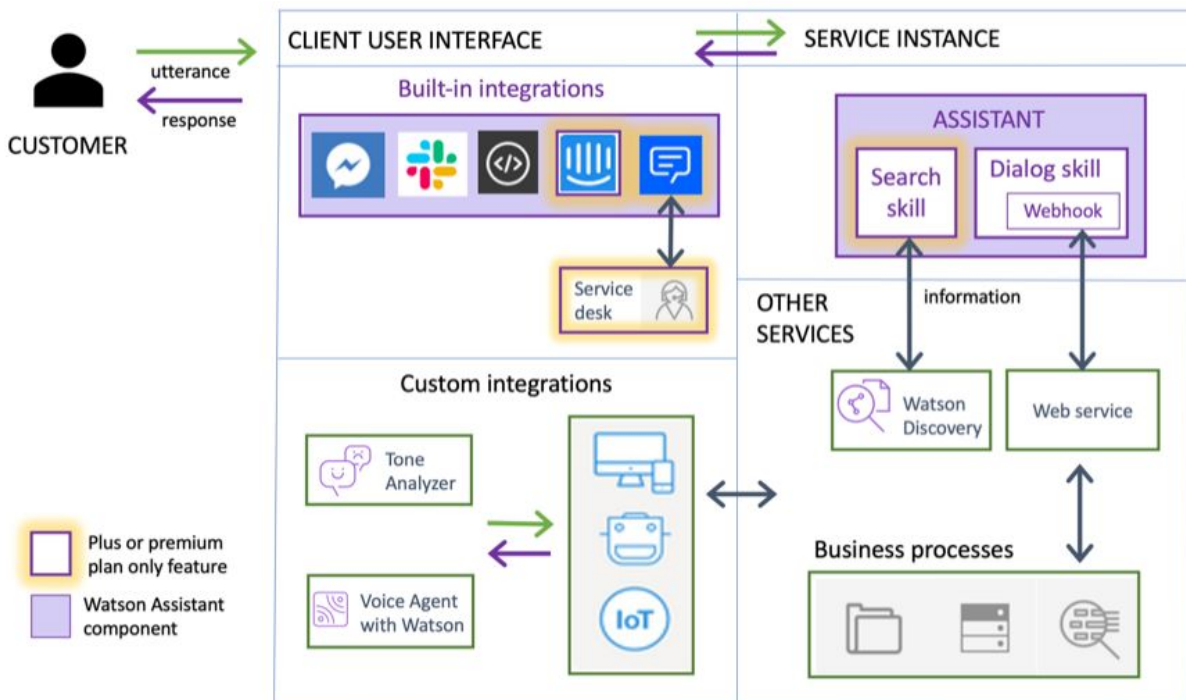


Figure 10: Implementation

As you add training data, a natural language classifier is automatically added to the skill. The classifier model is trained to understand the types of requests that you teach your assistant to listen for and respond to.

3. Bring the assistant to your customers where they are by adding integrations.

Add a built-in channel integration to deploy the configured assistant directly to a social media or messaging channel. Build your own client application as the user interface for the assistant. Or add the built-in web chat integration to your company website. From the web chat you can transfer customers who ask to speak to someone to your existing service desk personnel.

## Cloud Functions

IBM Cloud™ Functions service is an event-driven compute platform, also referred to as Serverless computing, or as Function as a Service (FaaS), that runs code in response to events or direct invocations.

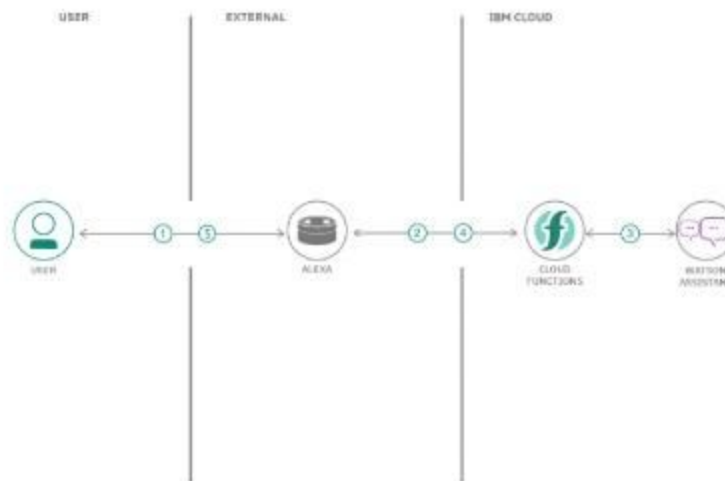


Figure 11: Implementation using cloud functions

## Cloud Functions terminology

The basic concepts of the technology behind Cloud Functions:

**Namespace:** It contains Cloud Functions entities, such as actions and triggers, and belong to a resource group. You can let users access your Cloud Functions entities by granting them access to the namespace.

**Action:** An action is a piece of code that performs one specific task. An action can be written in the language of your choice, such as small snippets of JavaScript or Swift code or custom binary code embedded in a Docker container. You provide your action to Cloud Functions either as source code or a Docker image. An action performs work when it is directly invoked by using the Cloud Functions API, CLI, or iOS SDK. An action can also automatically respond to events from IBM Cloud services and third-party services by using a trigger.

**Sequence:** A set of actions can be chained together into a sequence without having to write any code. A sequence is a chain of actions, invoked in order, where the output of one action is passed as input to the next action. By creating a sequence, you can combine existing actions together for quick and easy reuse. A sequence can then be invoked just like an action, through a REST API or automatically in response to events.

**Event:** Examples of events include changes to database records, IoT sensor readings that exceed a certain temperature, new code commits to a GitHub repository, or simple HTTP requests from web or mobile apps. Events from external and internal event sources are channeled through a trigger, and rules allow actions to react to these events.

**Trigger:** They are a named channel for a class of events. A trigger is a declaration that you want to react to a certain type of event, whether from a user or by an event source.

**Rule:** A rule associates a trigger with an action. Every time the trigger fires, the rule uses the trigger event as input and invokes the associated action. With the appropriate set of rules, it's possible for a single trigger event to invoke multiple actions, or for a single action to be invoked as a response to events from multiple triggers.

**Feed:** A feed is a convenient way to configure an external event source to fire trigger events that can be consumed by Cloud Functions. For example, a Git feed might fire a trigger event for every commit to a Git repository.

**Package:** Integrations with services and event providers can be added with packages. A package is a bundle of feeds and actions. A feed is a piece of code that configures an external event source to fire trigger events. For example, a trigger that is created with an IBM Cloudant change feed configures a service to fire the trigger every time a document is modified or added to an IBM Cloudant database.

## Actions In IBM Watson

IBM Action feature allows users to develop dialogs in a rapid fashion. The approach taken with Actions is one of an extreme non-technical nature. The interface is intuitive and requires virtually no prior development knowledge or training.

### How To Use Actions

*Firstly, **Actions*** should be seen as another type of skill to complement the other two existing skills;

**Dialog skill:** Dialog offers all the smarts, power, and flexibility you've come to trust. Select to keep building with the tools you know and love.

**Search skills:** Create Q&A experiences in minutes. Sync with websites and data sources for always up-to-date answers. Handle even complex questions with inclusive, contextual responses.

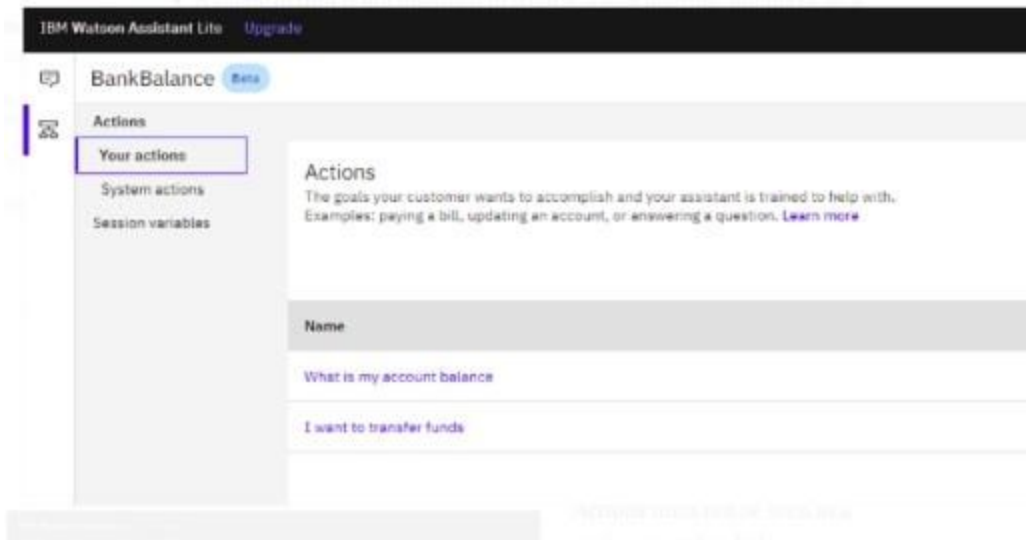


Figure 12: Creating Actions in Cloud Functions



## **Implementation**

### **Source Code**

[Github Link](#)

### **Demo Video**

[Vide Link](#)

## **Discussion and Conclusion**

In this technical report, we have discussed the way to make a chatbot using IBM Watson. By using Watson it could be easy and more efficient to make an intelligent chatbot, but the conversations with the customer still are scripted.

Furthermore, some important initial steps like training the data have been done manually. It is not possible to make a completely automated chatbot yet, but with the services of Watson, IBM makes it possible to make a chatbot that learns from earlier interactions or data and responds in other ways in different kinds of situations. If you want to make a smart chatbot, some coding skills are required. The several services do have web interfaces to train the bot with to make it smarter, but it is still not that easy to make a chatbot that can do a lot of different things.

## **REQUIREMENTS**

**Software Requirements:**

IBM Developer Account, Node.js

**Hardware Requirements:**

Operating System (Mac OS, Microsoft Windows, or any Linux distribution) PC with minimum 32-bit processor.

## **REFERENCES**

1. <https://github.com/IBM/alex-skill-watson-assistant>
2. <https://www.youtube.com/watch?v=j8TBqD2rx2o>
3. <https://eu-gb.assistant.watson.cloud.ibm.com/eu-gb/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa~2Fa4a93f80f6294d26bfac2754d64fa069%3A001d70c6-2d3f-4e4d-b3f6-0707443ee08b%3A%3A/home>