**Managerial User Guide**

|  |  |
| --- | --- |
| UMGC Fall 2020 | Benjamin Fetterman, Benjamin Murray, Hanim Danur, James Cornelius, Robert Lee  SWEN 670 |

Project Plan Approvals

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Approved by:  Dr. Mir Assadullah |  |  |
| Approved by:  "Stakeholder" |  |  |
| Approved by:  "Project Manager" | Robert Lee | 11/3/2020 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Name | Description of Change(s) |
| 1.0 | 11/03/2020 | Robert Lee | Initial Release |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.0 Introduction 5](#_Toc55321592)

[1.1 Purpose 5](#_Toc55321593)

[1.2 Intended Audience 5](#_Toc55321594)

[1.3 Technical Project Stakeholders 5](#_Toc55321595)

[1.4 Technical Team Skills 5](#_Toc55321596)

[1.5 Hardware/Software Specifications 6](#_Toc55321597)

[1.6 Licensing Considerations 6](#_Toc55321598)

[1.7 Deployment of the Chatbot 6](#_Toc55321599)

[2.0 Software Description and Data 7](#_Toc55321600)

[2.1 Characteristics, Capabilities, and Features 7](#_Toc55321601)

[2.1.1 Characteristics 7](#_Toc55321602)

[2.1.2 Capabilities and Limitations 7](#_Toc55321603)

[2.1.3 Features 8](#_Toc55321604)

[2.2 Support 14](#_Toc55321605)

[2.3 History of The Application 15](#_Toc55321606)

[2.4 Connections to The Application 15](#_Toc55321607)

[3.0 Functional Interface 15](#_Toc55321608)

[3.1 Login to IBM cloud Account 16](#_Toc55321609)

[3.2 Create Intents 17](#_Toc55321610)

[3.3 Create Entities 19](#_Toc55321611)

[3.4 Create Dialog 21](#_Toc55321612)

[4.0 Troubleshooting 24](#_Toc55321613)

[4.1 IBM Watson Assistant 24](#_Toc55321614)

[4.2 Spring IO REST API 24](#_Toc55321615)

[4.3 MySQL Database 24](#_Toc55321616)

[5.0 Maintenance 24](#_Toc55321617)

[5.1 IBM Watson Assistant 24](#_Toc55321618)

[5.2 Spring IO REST API and MySQL Database 25](#_Toc55321619)

[6.0 Technical Support 25](#_Toc55321620)

[Appendix A: Acronyms and Abbreviations 26](#_Toc55321621)

[Appendix B: References 27](#_Toc55321622)

List of Tables

[Table 1 Stakeholders 5](#_Toc55321623)

List of Figures

[Figure 1 Accessing the City Chatbot agent 8](#_Toc55321624)

[Figure 2 City Chatbot Service Page 9](#_Toc55321625)

[Figure 3 Assistants 9](#_Toc55321626)

[Figure 4 Assistant Management 10](#_Toc55321627)

[Figure 5 Intents 10](#_Toc55321628)

[Figure 6 Entities 11](#_Toc55321629)

[Figure 7 Dialog 11](#_Toc55321630)

[Figure 8 Options 11](#_Toc55321631)

[Figure 9 Analytics 12](#_Toc55321632)

[Figure 10 Versions 12](#_Toc55321633)

[Figure 11 Content Catalog 13](#_Toc55321634)

[Figure 12 Dialog Nodes 13](#_Toc55321635)

[Figure 13 Webhook Results 14](#_Toc55321636)

[Figure 14 Webhook Parameters 14](#_Toc55321637)

[Figure 15 Dialog Action 14](#_Toc55321638)

[Figure 16 Watson Logon 16](#_Toc55321639)

[Figure 17 Start Watson Service 16](#_Toc55321640)

[Figure 18 Select Watson Assistant 17](#_Toc55321641)

[Figure 19 Start Assistant 17](#_Toc55321642)

[Figure 20 Create Intent 18](#_Toc55321643)

[Figure 21 Setup an Intent 18](#_Toc55321644)

[Figure 22 Complete Intent 19](#_Toc55321645)

[Figure 23 Create Entities 19](#_Toc55321646)

[Figure 24 New Entities 20](#_Toc55321647)

[Figure 25 Synonyms for Entities 20](#_Toc55321648)

[Figure 26 Entities Annotations 21](#_Toc55321649)

[Figure 27 Try it out Bot 21](#_Toc55321650)

[Figure 28 Dialog Tree 22](#_Toc55321651)

[Figure 29 Dialog Options 22](#_Toc55321652)

[Figure 30 Dialog Node Options 23](#_Toc55321653)

[Figure 31 Dialog Conditions 23](#_Toc55321654)

[Figure 32 Add Webhook to Dialog Node 24](#_Toc55321655)

1. Introduction

The UMGC chatbot application was developed in order to aid the City of Pasadena residents in their requests for regulations information and permit applications. This document aids in the management of the UMGC chatbot application.

Purpose

This managerial user’s document purpose is to describe the UMGC chatbot application software’s description, software’s data, functional interface, troubleshooting and maintenance. This document is written to aid in the management of the UMGC chatbot application.

Intended Audience

The chatbot system was designed for the City of Pasadena. As such this document intended audience is City of Pasadena employee who is tasked with the management of the UMGC chatbot application. As such this document is written for a non-technical audience, those with more technical questions should refer to the programmer’s guide.

Technical Project Stakeholders

Table 1 Stakeholders

|  |  |  |
| --- | --- | --- |
| Name | E-mail address | Role |
| Professor Assadullah | mir.assadullah@faculty.umgc.edu | Stakeholder |
| Robert Lee | rlee97@student.umgc.edu | Project Manager |
| Benjamin Fetterman | bfetterman2@student.umgc.edu | Developer |
| Benjamin Murray | bmurray19@student.umgc.edu | Developer |
| Hanim Danur | hdanur@student.umgc.edu | Developer |
| James Cornelius | jcornelius10@student.umgc.edu | Developer |
| Glenn Goodlett | ggoodlett1@student.umgc.edu | DevOps |
| Dustin Emerson | demerson2@student.umgc.edu | DevOps |

Technical Team Skills

The technical team will need skills sets in order to integrate, expand, and maintain the city chatbot.

Familiarity with IBM Watson

IBM Watson is the AI platform from which the chatbot is based on. Specifically experience in creating intents, entities and dialogs will be of paramount importance. Also it is important to have familiarity with the configuration and use of webhooks with IBM Watson, as the chatbot make use of webhooks.

Familiarity with REST API

REST API is used to send and receive HTTP Requests and HTTP responses. Knowledge of how this works will improve understanding of how the chatbot works.

Familiarity with MySQL Database

MySQL is the relational database management system that is being used to house all the data the team has recorded. It is organized into various tables that need to be updated and maintained. New permits, regulations, or developments mean that the database needs to be updated.

Research, Analysis, and software engineering skills

In order to maintain and update the code provided by the team various software engineering skills are needed. Specifically, the code was developed in Java 8, on eclipse, using Spring IO. Experience connecting to a API services is needed as the chatbot makes use of MapQuest API in order to verify addresses.

Hardware/Software Specifications

Hardware and software specifications for IBM Watson is limited. The following are required:

* Desktop/Laptop Computer/Smart Phone/Tablet/Devices – Any common user devices able to access the internet.
* Software – Operating System and internet browser-based applications are the bare minimum. This includes normal and commonly used operating systems and browser applications.
* Telecommunication – Internet access is required. As IBM Watson is maintained in IBM Cloud’s infrastructure.

Licensing Considerations

Software developed for the chatbot is deemed to be open source educational work and available for use and modification. Please refer to the License Information section within the Programmers guide for more information.

Deployment of the Chatbot

REST API and Watson will need to be installed and configured. The deployment of the chatbot is discussed in far greater detail in the Deployment guide, please refer to it for questions.

1. Software Description and Data

Characteristics, Capabilities, and Features

This section discusses the characteristics, capabilities and features of the IBM Watson Chatbot agent. The tool’s characteristics detail major aspects of the tool; which are functionality, portability, efficiency, reliability, maintainability and usability.

* + 1. Characteristics

IBM AI Chatbot is a user experience tool. Developers can configure the tool to provide user fulfillment as to the scope and requirements of the application. The agent is highly portable, as it is effectuated on end user devices and maintained by a third party within the IBM Cloud infrastructure. The agent is reliable, as it is maintained within IBM’s Cloud infrastructure. It’s performance efficiency and capacity is limited by the licensing structure. Where higher grade license tiers have dedicated resources and higher data communication limits. Usability of the agent is high as configuration is GUI based; as a tool, the agent can be configured to meet the needs of most business scopes and use cases; and learning is relatively easy as there is documentation available. IBM Watson documentation can be found here: <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>.

* + 1. Capabilities and Limitations

As a stand in intermediary tool between end users and organizations. IBM Watson Chatbot capabilities are designed to fullfill a variety of business needs. Organizations will be able to:

* Process unstructured data
* Fills human limitations
* Act as a decision support system
* Improves performance and abilities by being able to provide accurate data
* Improve the customer service experience
* Handle large quantities of data for fulfillment
* Sustainable competitive advantage

IBM Watson Chatbot agent contains the following limitations:

* Limited to using the English language
* Maintenance
* Does not process structured data directly
* Increasing rate of data, with limited resources; impacting licensures
* High cost to implement and reconfigure organization IT infrastructure and business process
* Time and effort to teach Watson, as when the variey and amount of data increases. The time to teach Watson increases.

More information can be found here, <http://ibmwatson237.weebly.com/advantages--disadvantages.html>.

* + 1. Features

The features of the IBM Chatbot agent allows developers to manage and configure their application to fit their needs.

* **Accessing the City Chatbot agent** – The City Chatbot is accessible from IBM Cloud’s Resource List.

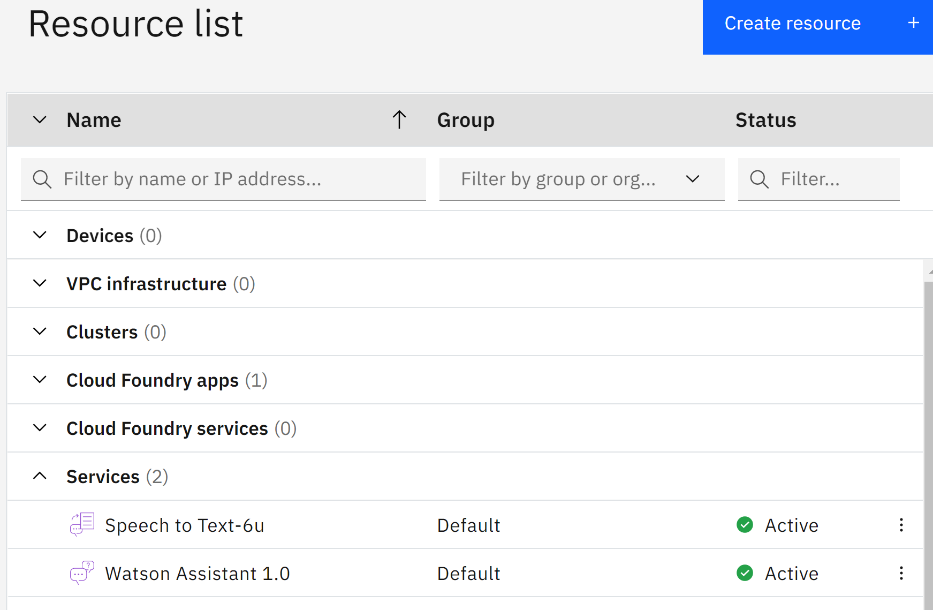


Figure Accessing the City Chatbot agent

* **City Chatbot Service Page** – The agent has an overall account management page. On this page, the user is able to do the following:
  + - Launch the assistant
    - Upgrade the plan.
    - View and access service API credentials
    - Connected Resources

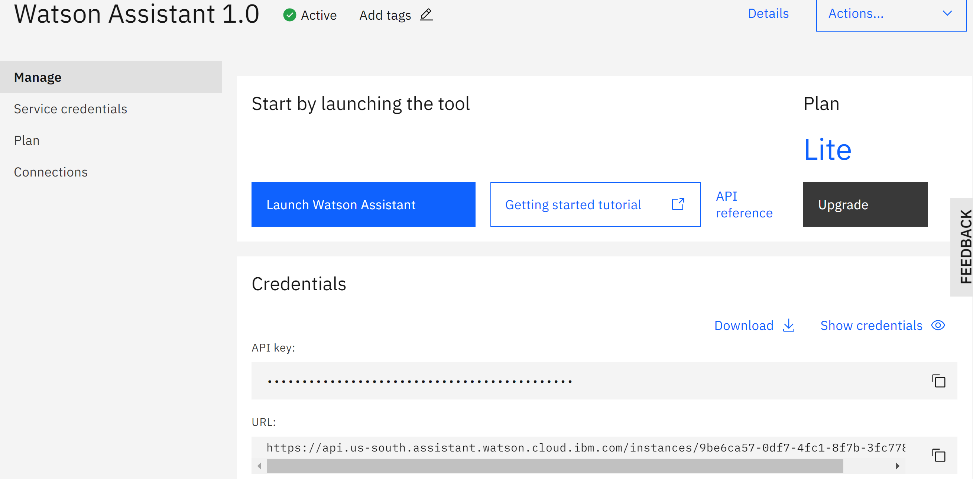


Figure City Chatbot Service Page

* **Assistants** - The City Chatbot is able to maintain multiple assistants. Each assistant can be unique to it’s functionality, purpose, and goals.

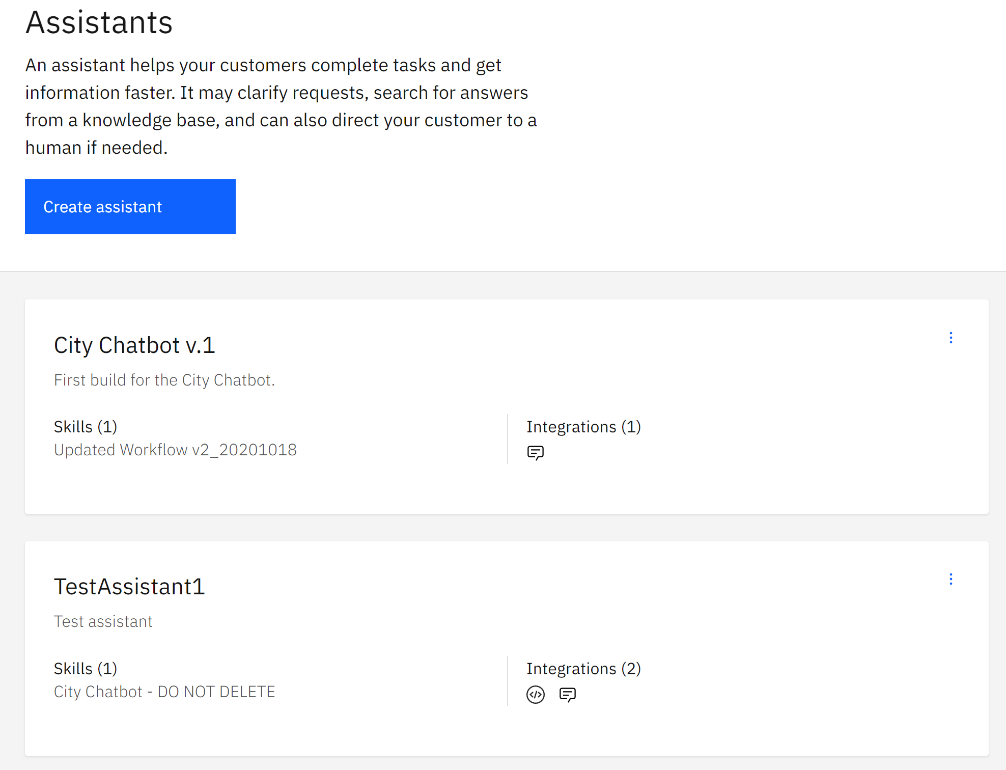


Figure Assistants

* **Assistant Management** – Once the assistant’s instance deploys, the user is able to manage the assistant. The user is able to view the connected dialog flow, add new actions, configure integrations, and view saved dialog flows.

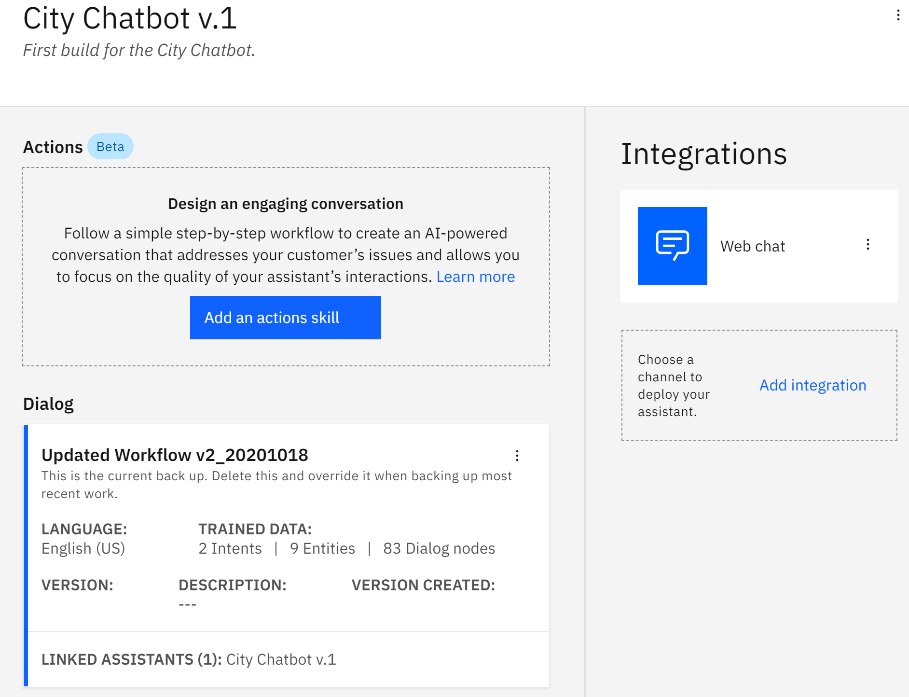


Figure Assistant Management

* **Dialog Flow Management** – The agent’s dialog flow is the primary place to configure the chatbot’s responses. The dialog flow contains the following parameters:
* **Intents** – An intent is a basic goal of the end user. The intents shape the way the AI engine categorizes the conversation. Here the user is able to create and configure intents.

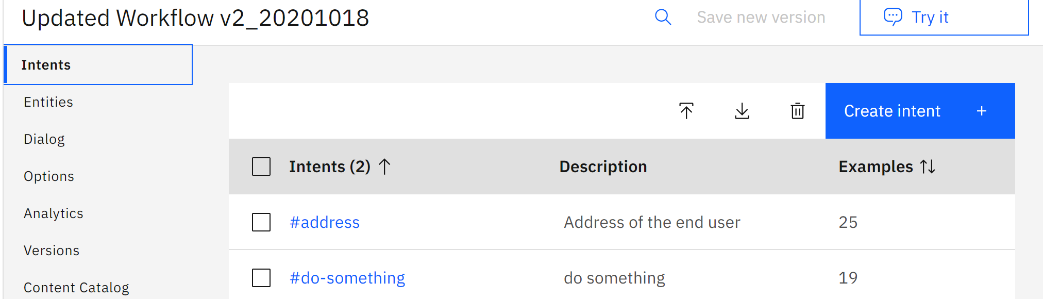


Figure Intents

* **Entities** – Entities are representations of the end user’s goal. Here, the user is able to create and configure entities.

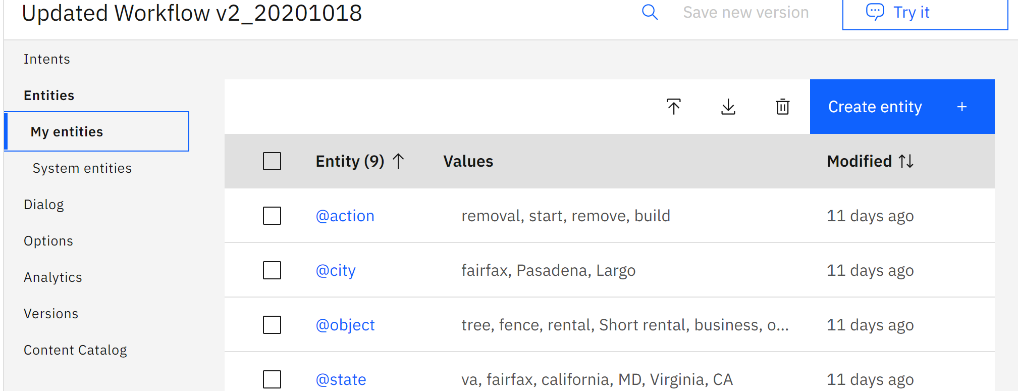


Figure Entities

* **Dialog** – Dialog is the agent’s basic conversational framework. The dialog flow is a branching network of nodes. Where each node is configured to establish an intent, entity, or another action leading to or implanting fulfillment.

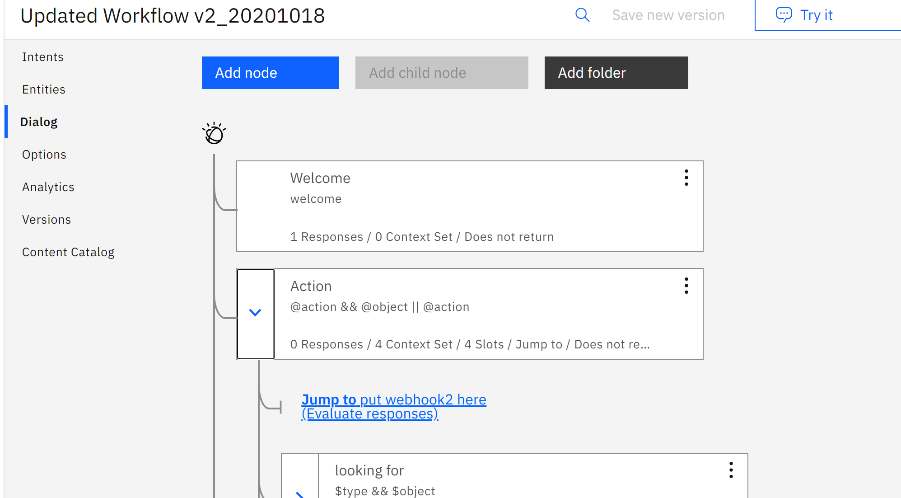


Figure Dialog

* **Options** – Options allow the user to establish webhooks, configure disambiguation, AI autocorrection, detect irrelevant inputs, and view system entities.

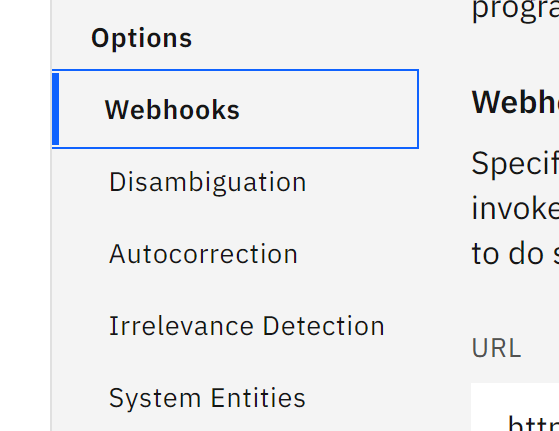


Figure Options

* **Analytics** – A dashboard displaying metrics related to total conversations, max conversations, average messages per conversation, and AI weak understanding counts of end user inputs.

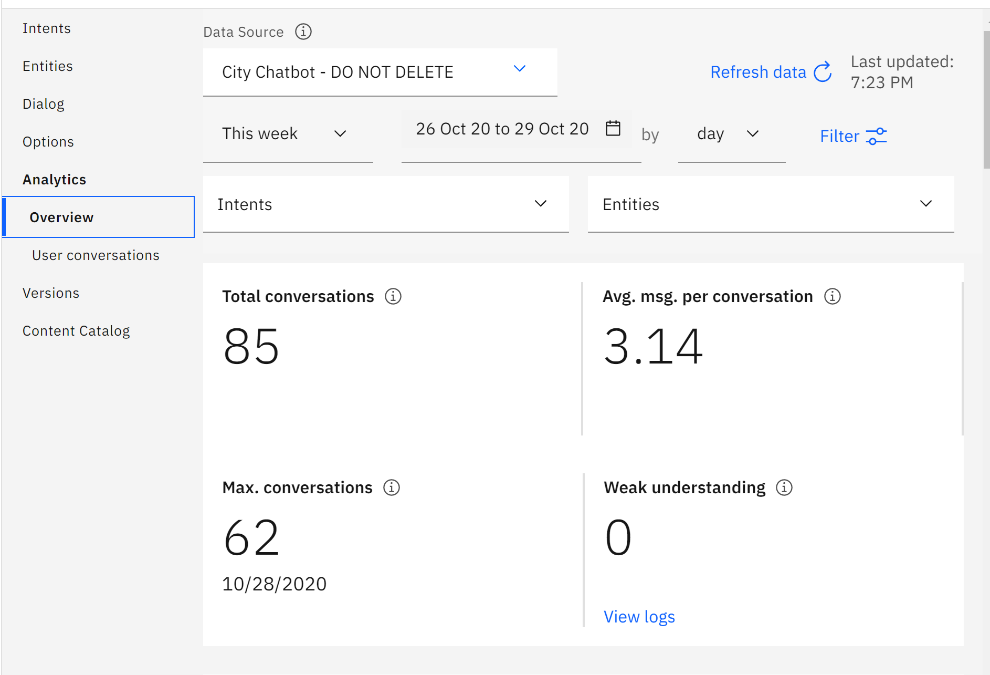


Figure Analytics

* **Versions** – Assistant versioning information.

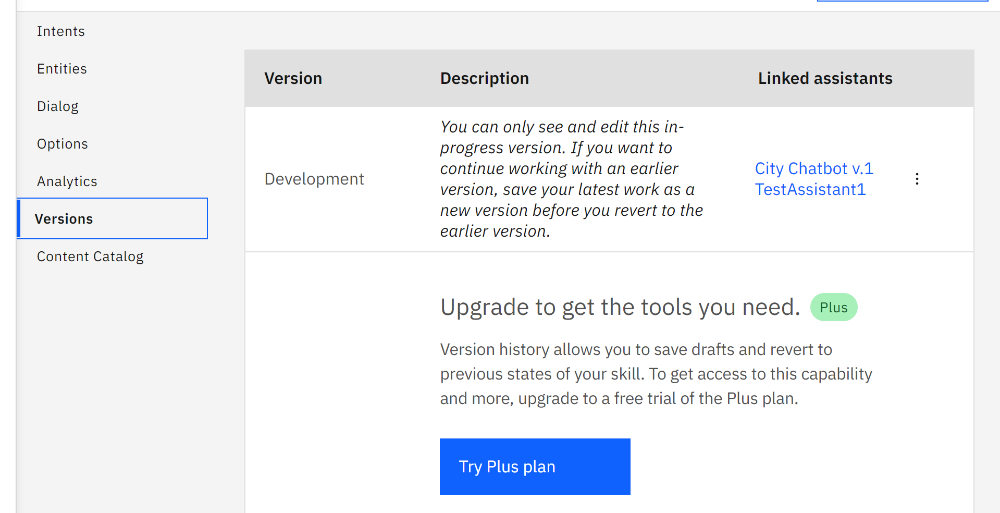


Figure Versions

* **Content Catalog** – Basic preloaded dialog flows.

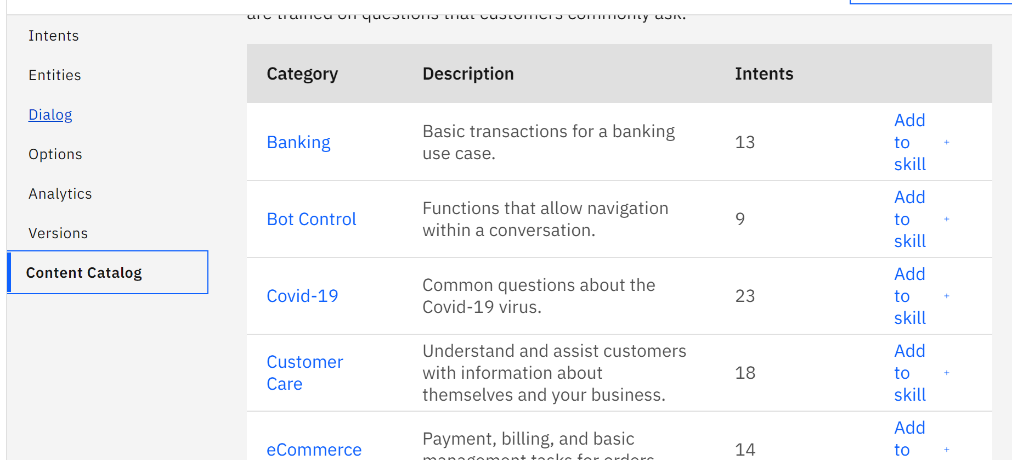


Figure Content Catalog

* **Dialog Nodes** - Dialog nodes are highly configurable and allows the user to configure the following actions.
* Establishing what the node recognizes from the input. This could be an intent or an entity.

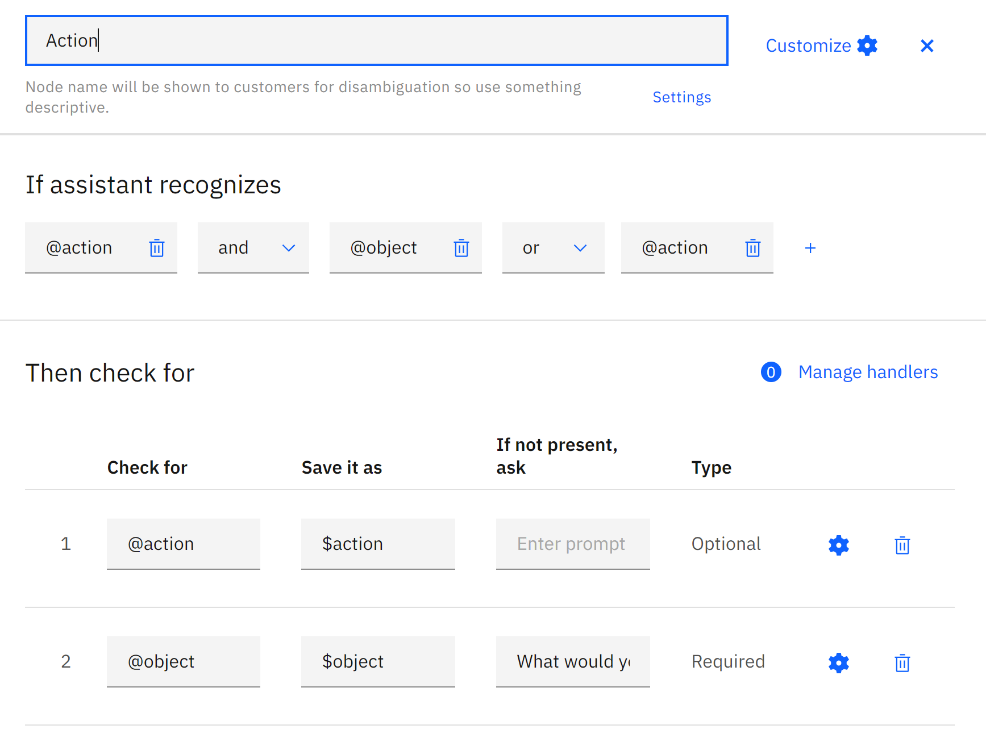


Figure Dialog Nodes

* The assistant’s response to recognized input.

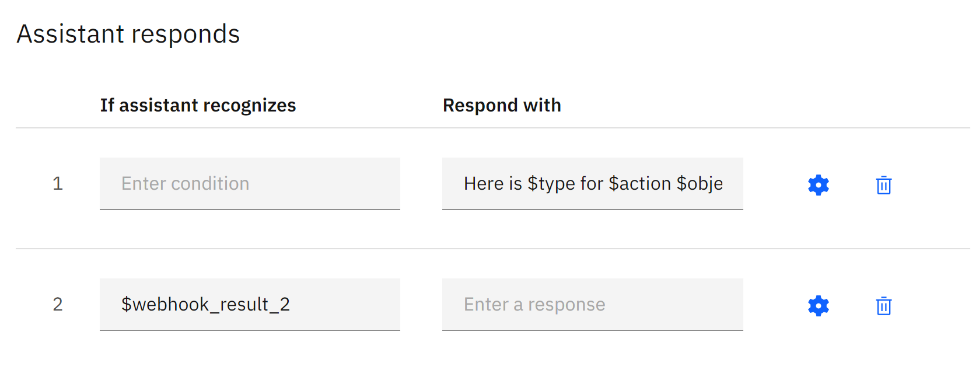


Figure Webhook Results

* If a webhook is required for that node, then an area display to configure the webhook callout and return variables.

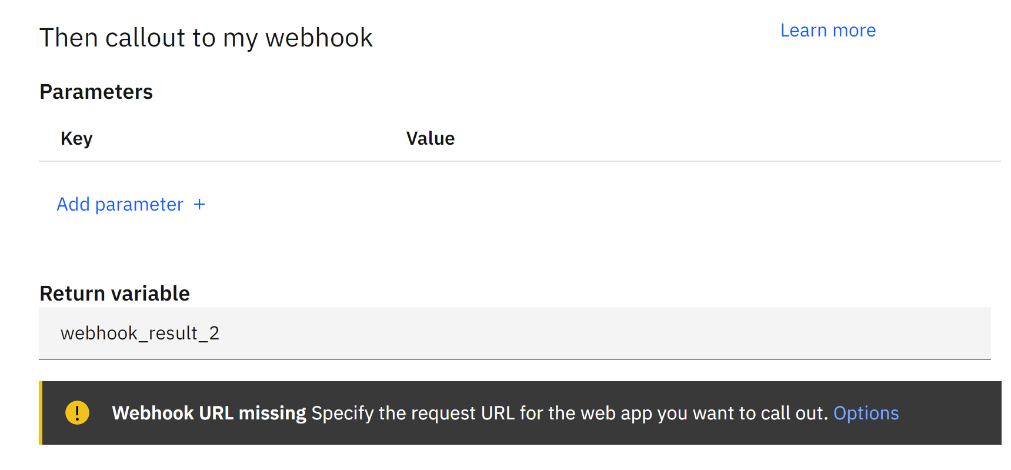


Figure Webhook Parameters

* Establishing what the node will do when all of it’s operations are complete.

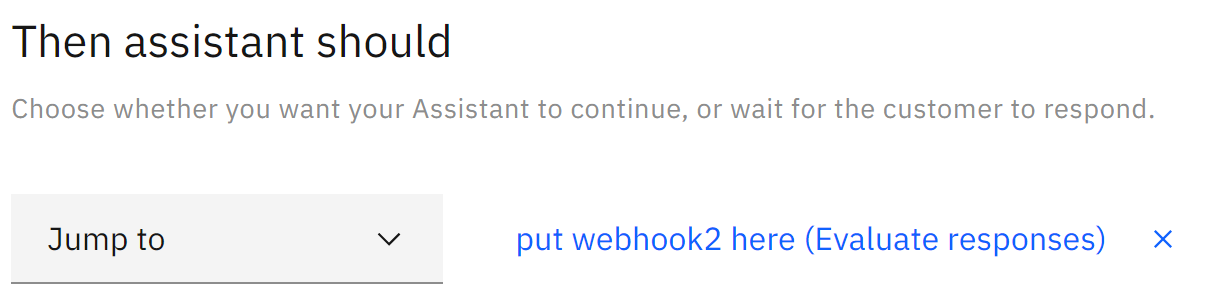


Figure Dialog Action

Support

Application support services are generally technical in nature. With limitations due to licencing. The application support services depends primarily on in house research, analysis and engineering. IBM provides the following documentation and referential information repositories.

* IBM Watson General Information – Source providing links to all documentation and API references for all of IBM Watson services. <https://cloud.ibm.com/developer/watson/documentation>
* IBM Watson Assistant Documentation – Documentation explaining the various capabilities, features, and configuration information. <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>
* IBM Watson Assistant API Reference – Documentation explaining the agent’s capabilities and configuration information regarding connecting with external APIs. <https://cloud.ibm.com/apidocs/assistant/assistant-v2>

History of The Application

IBM Watson was the next generation technological iteration of IBM Deep Blue. IBM Watson started off as a super computer designed to maintain an AI framework. The system’s early scope was to beat humans on the television game show Jeopardy. Where the system ingested 200 million pages of unstructured/structured data, amounting to roughly four terabytes of information. The system is an IBM effort and included development from various faculty and graduate students from

* Rensselaer Polytechnic Institute
* Carnegie Mellon University
* University of Massachusetts Amherst
* University of Southern California’s Information Sciences Institute
* University of Texas at Austin
* Massachusetts Institute of Technology
* University of Trento
* New York Medical College

The current plan for IBM Watson is to have computers interact with humans in a natural way, with the capabilities to process a broad range of information. As well as using the AI engine to process other information intensive fields, outside of customer fulfillment, such as telecommunications, financial services, and government.

Connections to The Application

IBM Watson is able to connect to a variety of APIs. Each connection is maintained by webhooks configurable within IBM Watson’s management webpage. For the purposes of the City Chatbot. The agent connects to a custom API service layer, scripted in Java Spring.io.

1. Functional Interface

This section includes a step by steps descriptions and images to demonstrate to login to the chatbot and create intent, entities and dialogs. It also includes detail illustration on how the IBM chatbot dialog skills works. A chatbot uses three main components to respond to a user’s query. Intent is the most important component of the chatbot because it controls what the user wants. The chatbot intent will allow the bot to determine the response. Entity is a phrase or object that provides context for an intent. The intent and the entities are the two-component work together to respond to the user. Entities search for and identifies specific values in a user’s question. The third most important part of the component is the dialog. Dialog determines the response the chatbot will give based on the user’s intent and specific entity during input.

### Login to IBM cloud Account

Log into IBM cloud account using the chatbot email address.

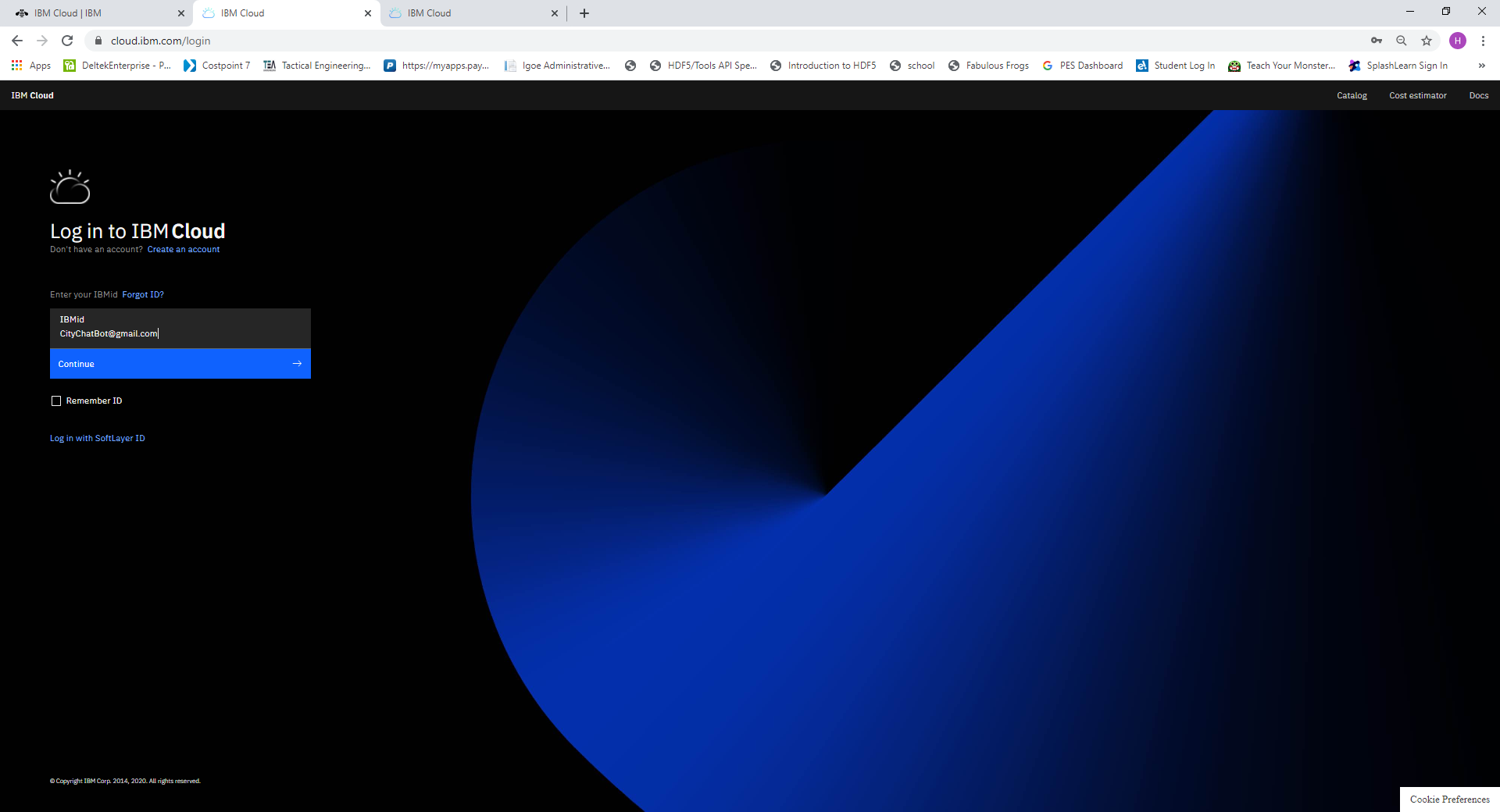


Figure Watson Logon

Starting service: - on Dashboard, click on the Service.

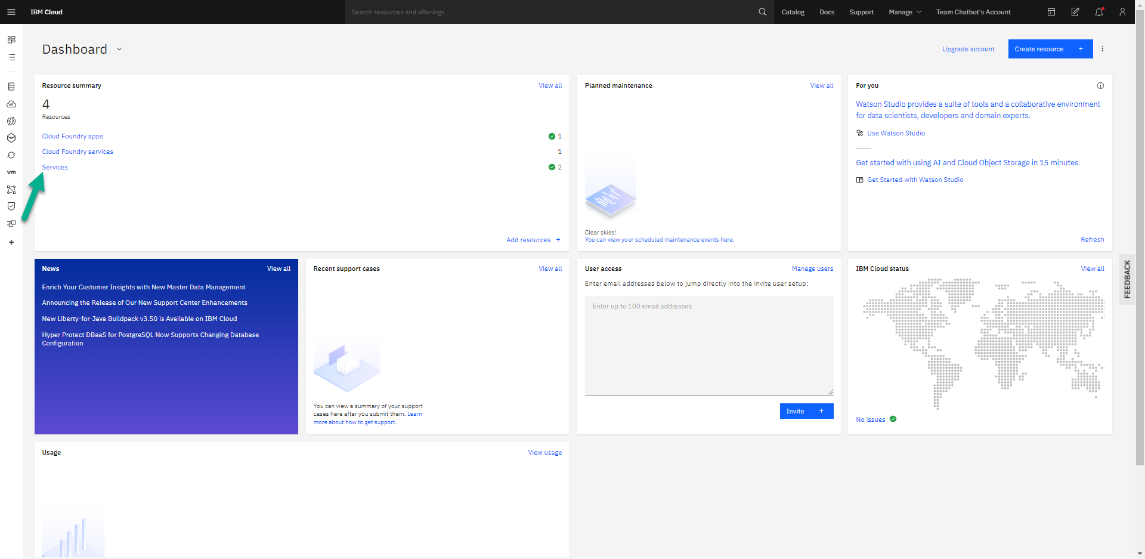


Figure Start Watson Service

Select services: -Watson Assistant1.0

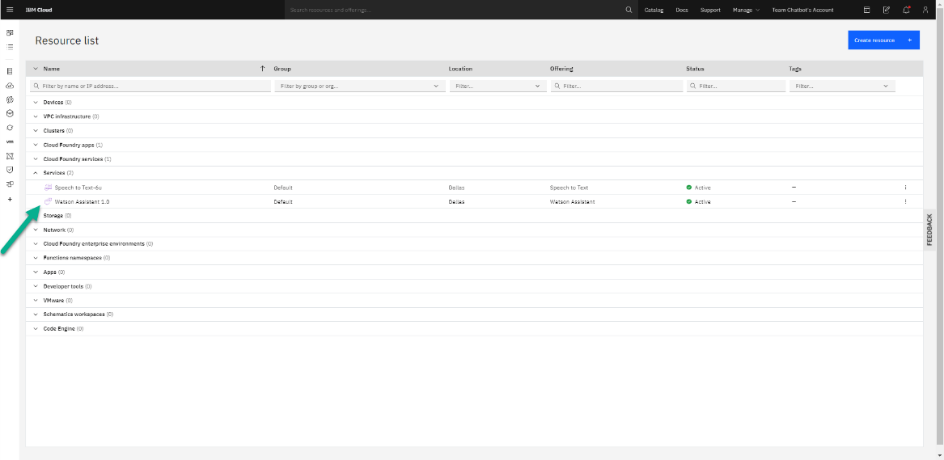


Figure Select Watson Assistant

Launch Watson Assistant: - Start by launching the Watson assistant

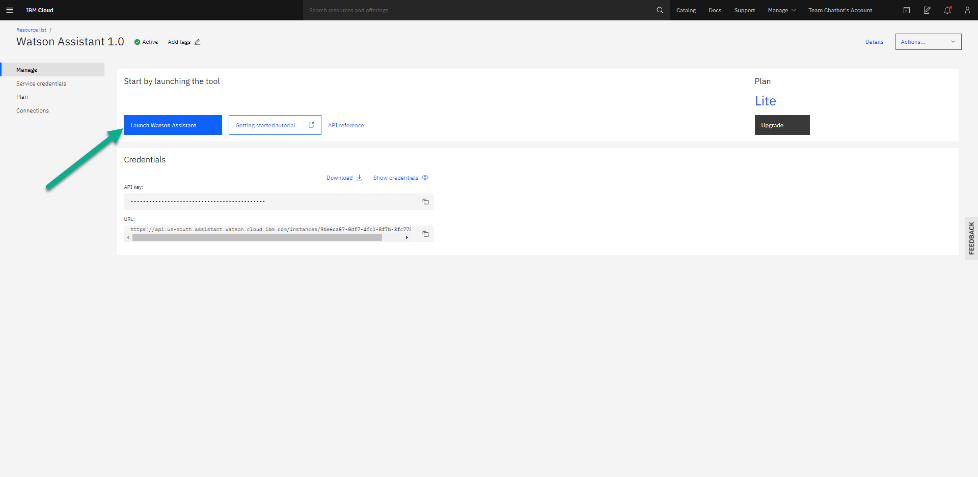


Figure Start Assistant

### Create Intents

A user purpose or an intention of user request when interacting with the application. It could be something like finding basic information including greetings.

To set intents, click “create intent”

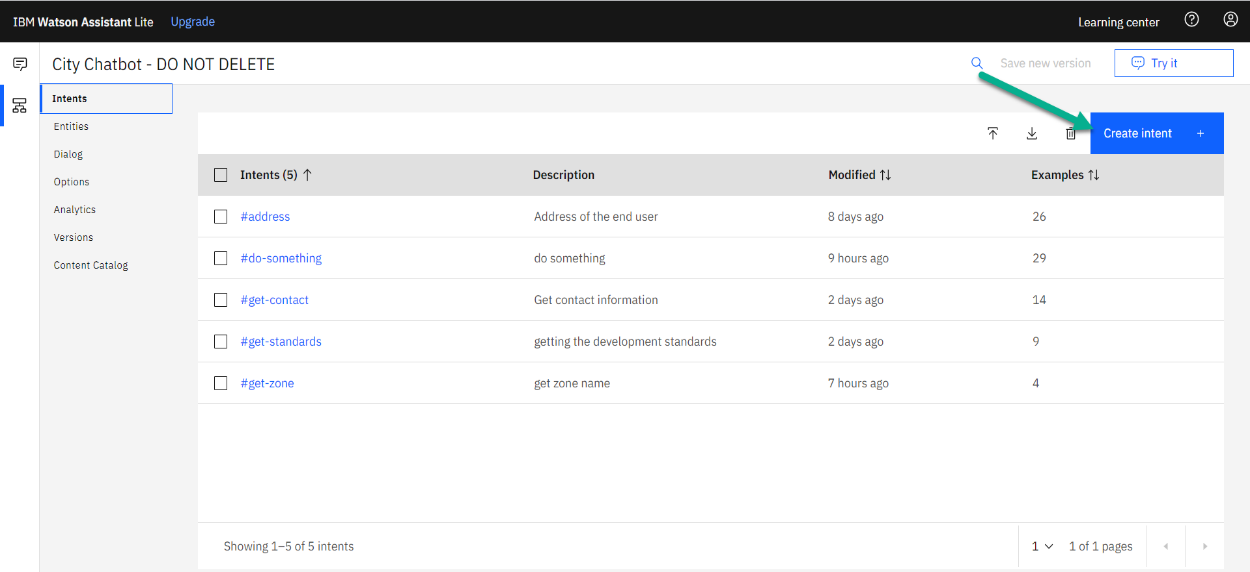


Figure Create Intent

Sequence of steps to add an intent

* Add an intent name which reflect the intent. Ex- #get-zone.
* Add description of an intent: - this is optional however adding it will give meaning to the intent name given.
* Add user examples: - the type of examples that a user expected to say.
  + Example: -what is my zone
* Repeat these steps to create multiple intents your users may have

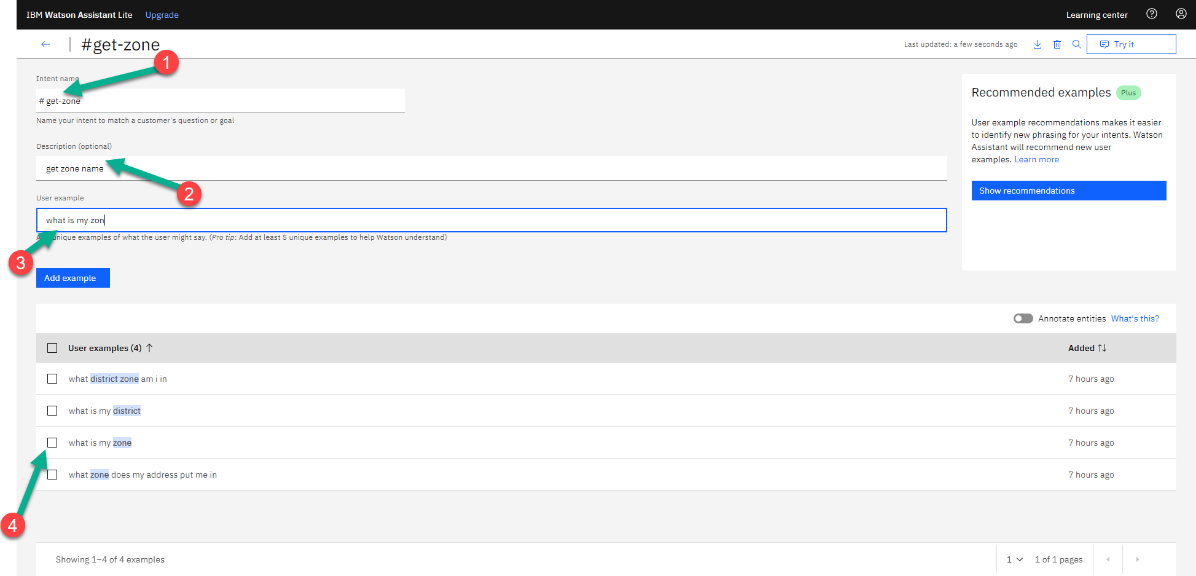


Figure Setup an Intent

Image shows created intent, description and status

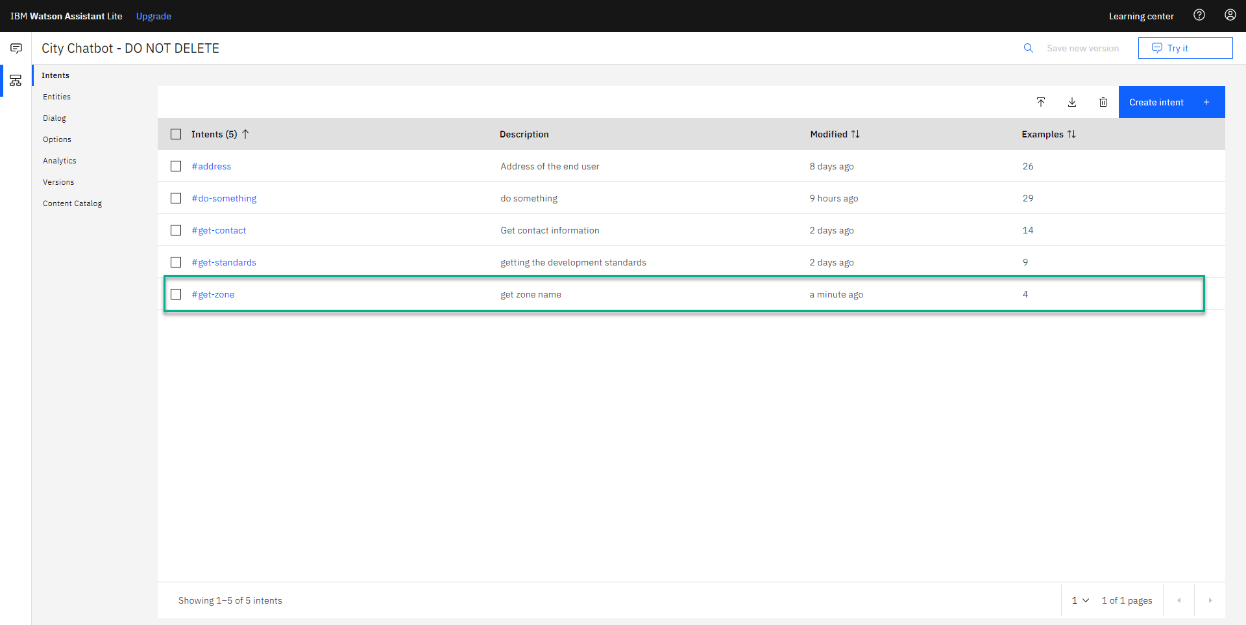


Figure Complete Intent

### Create Entities

Create entities is phrase or object that provide context for an intent. This is some entities related to the chatbot such as @zone so that the chatbot knows how to start answering questions. Values are used by the chatbot to find an entity based on synonyms a customer may use in their question- ex- remove. Add multiple synonyms word for the value added. User may use different word to ask the same question so having more synonyms increase functionality of the chatbot.

To set entities, click “create entities”

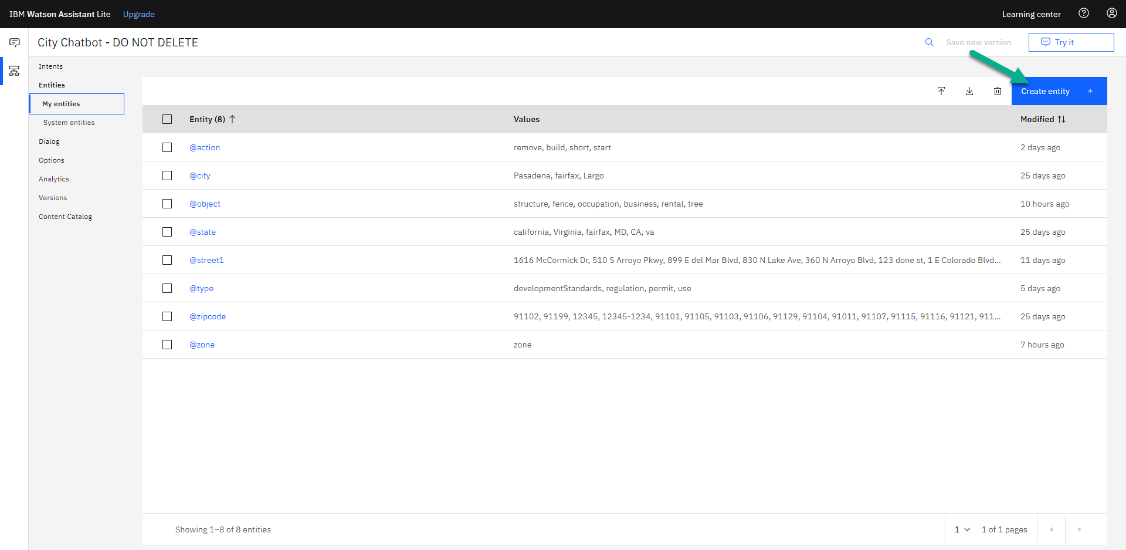


Figure Create Entities

Sequence of steps to add an entity

* Add value
* Add synonyms word for the value added.
* Repeat these steps to create multiple intents your users may have

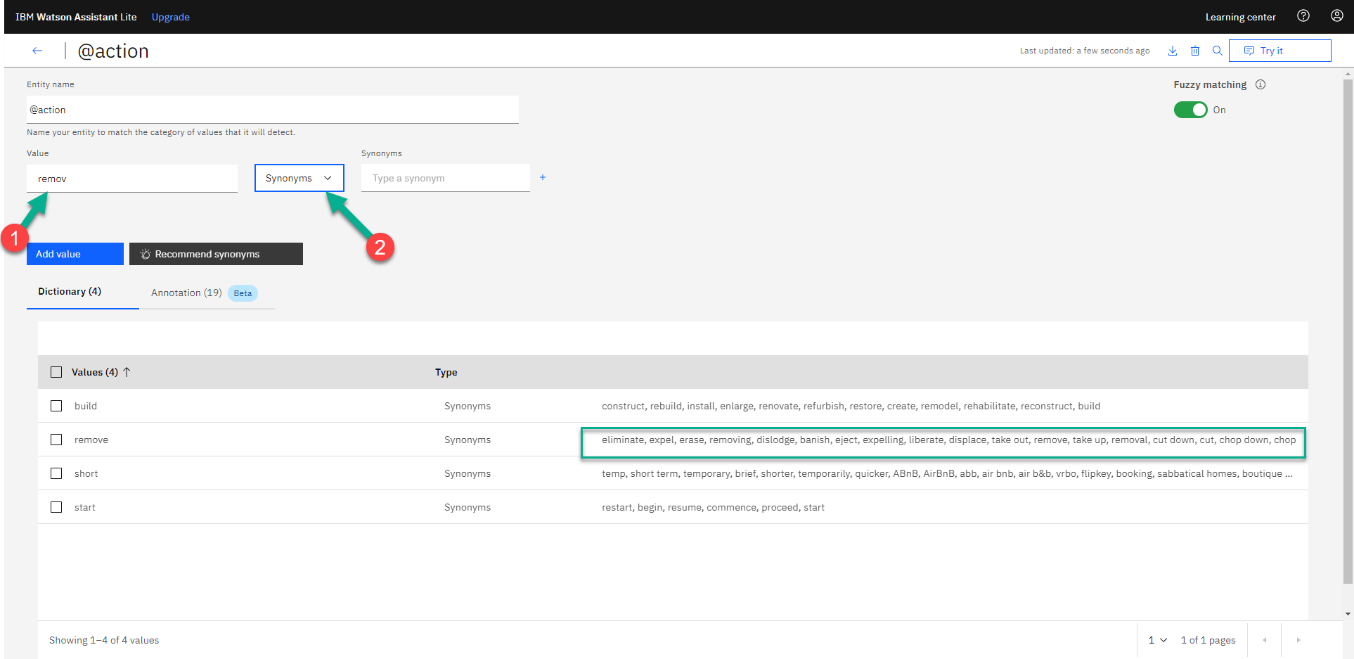


Figure New Entities

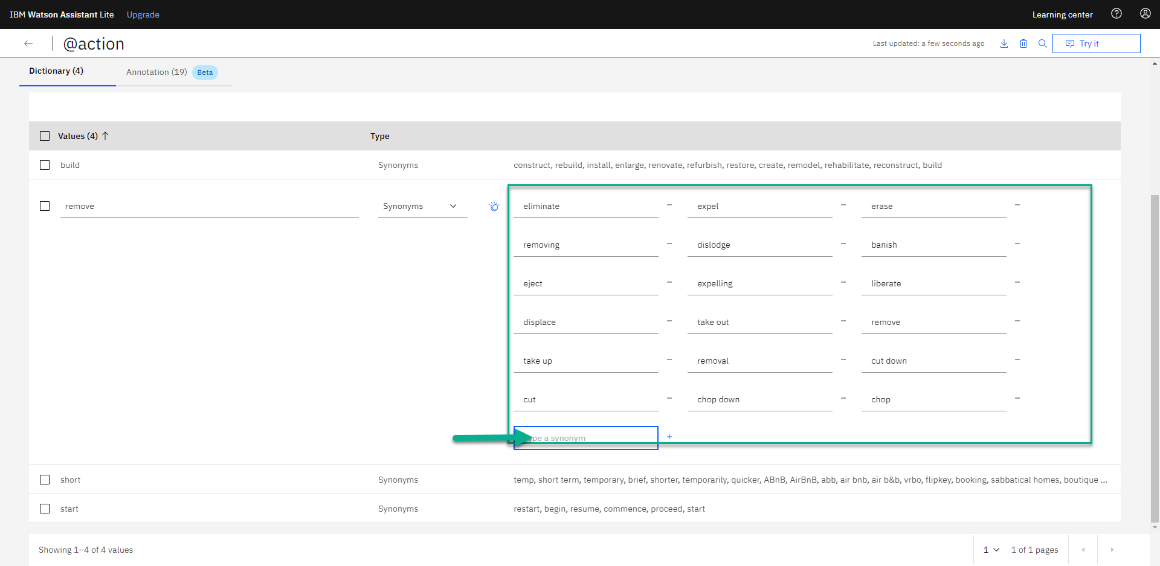


Figure Synonyms for Entities

Annotations: - statements added what the user might say when creating an intent

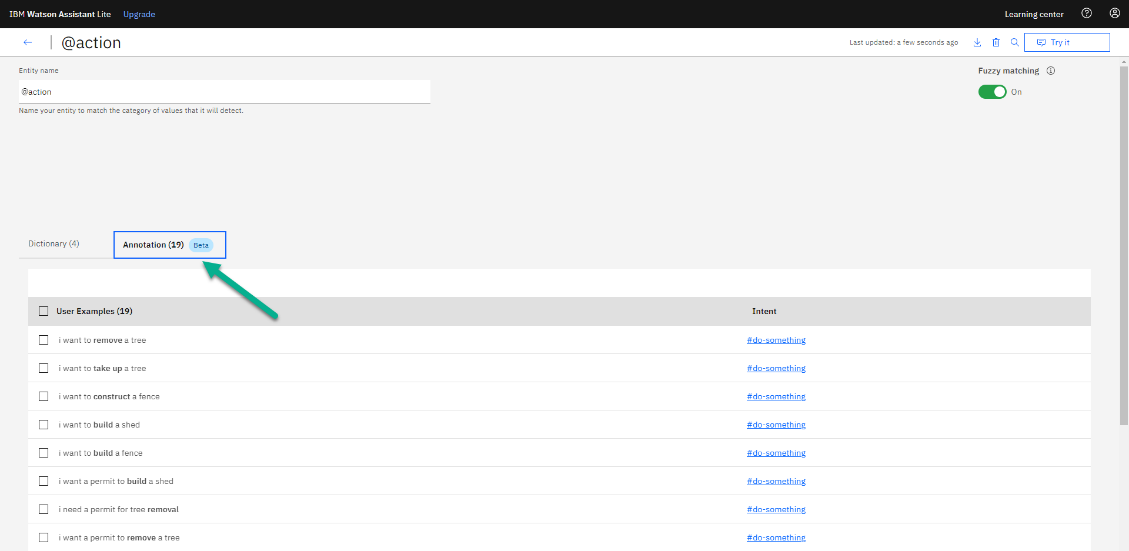


Figure Entities Annotations

After Adding your value

* Click “Try it” to preview an instance of your chatbot.
* Add an instant but at this point the chatbot don’t understand until dialog is created

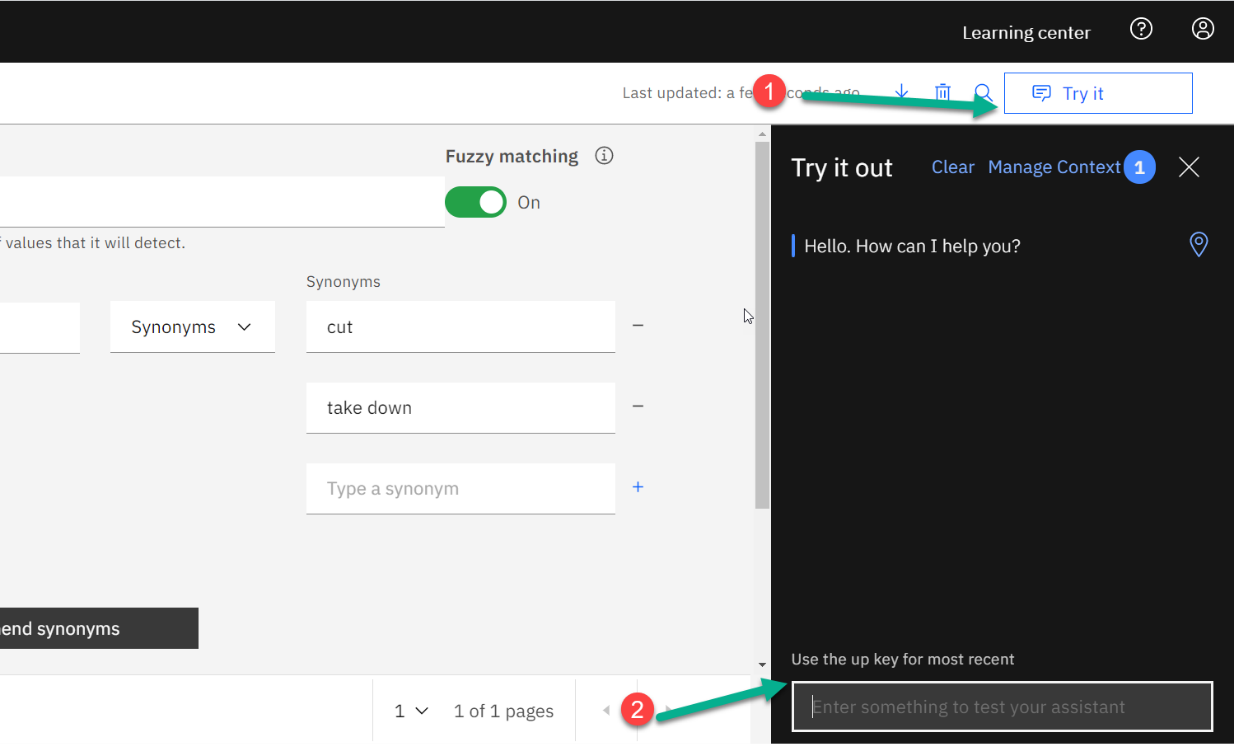


Figure Try it out Bot

### Create Dialog

**A** dialog is how the chatbot responds to users and moves to new topics based on prior responses using dialog trees. A node deals with responses to specific entities or intents. A node name should be detailed and relate to the issue the chatbot will be dealing with.

Dialog tree refers users to diverse topics using nodes

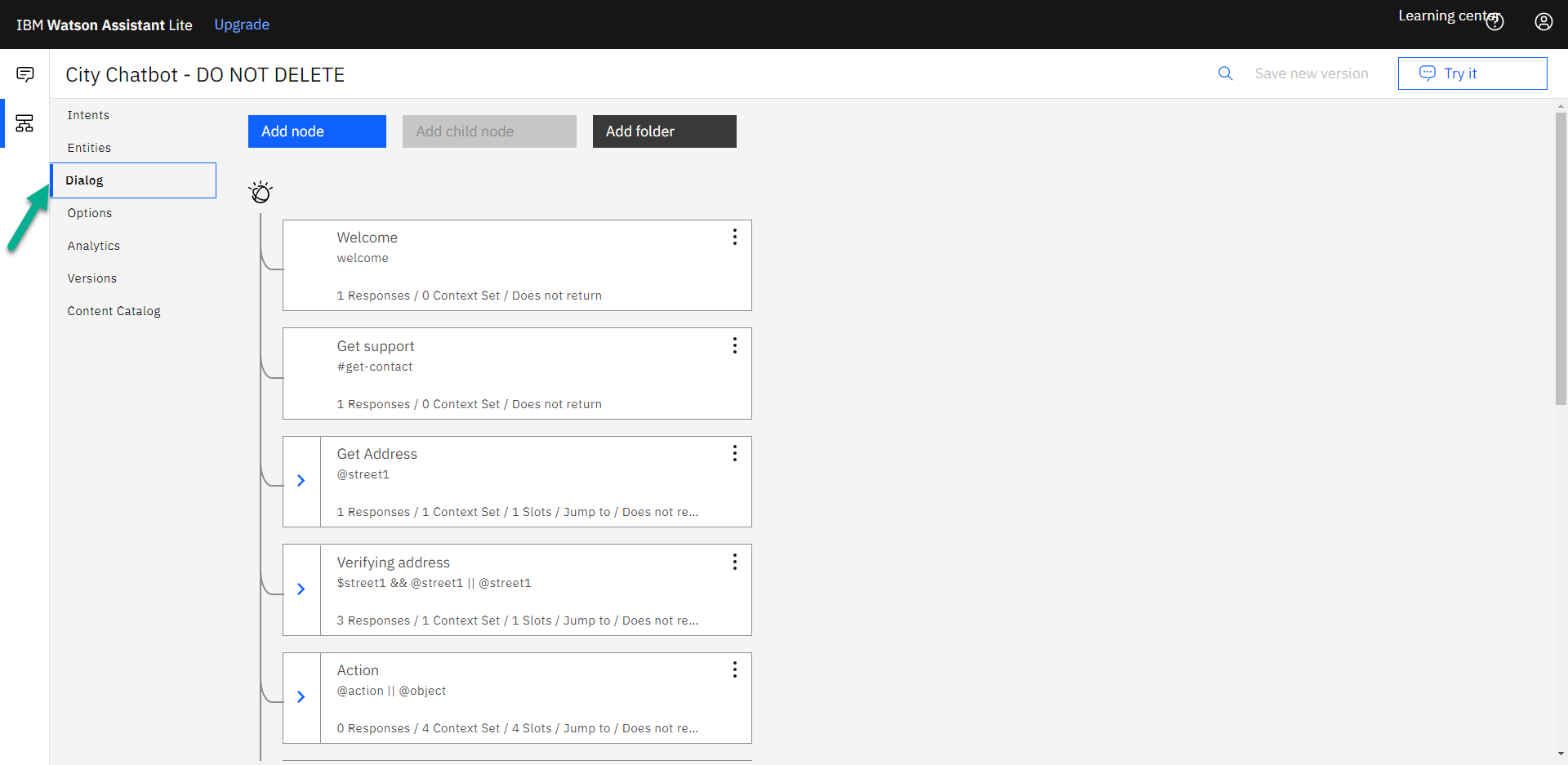


Figure Dialog Tree

Options on the dialog tree to add node up or down to the tree, remove, move, duplicate

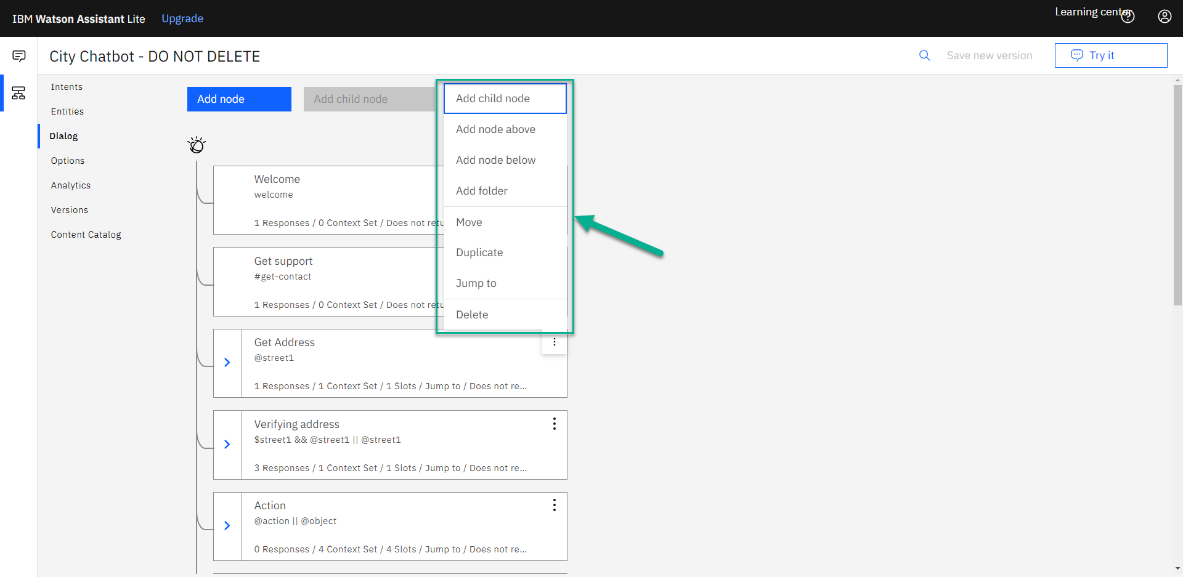


Figure Dialog Options

Sequence of steps to add dialog nodes

* Adding Node
* Add Node name
* Add response to the user

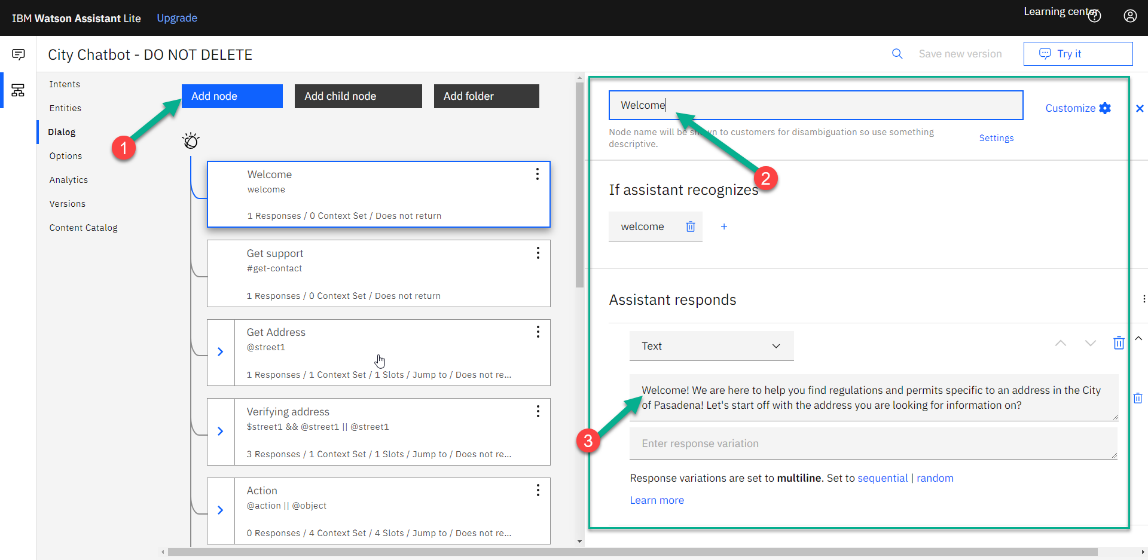


Figure Dialog Node Options

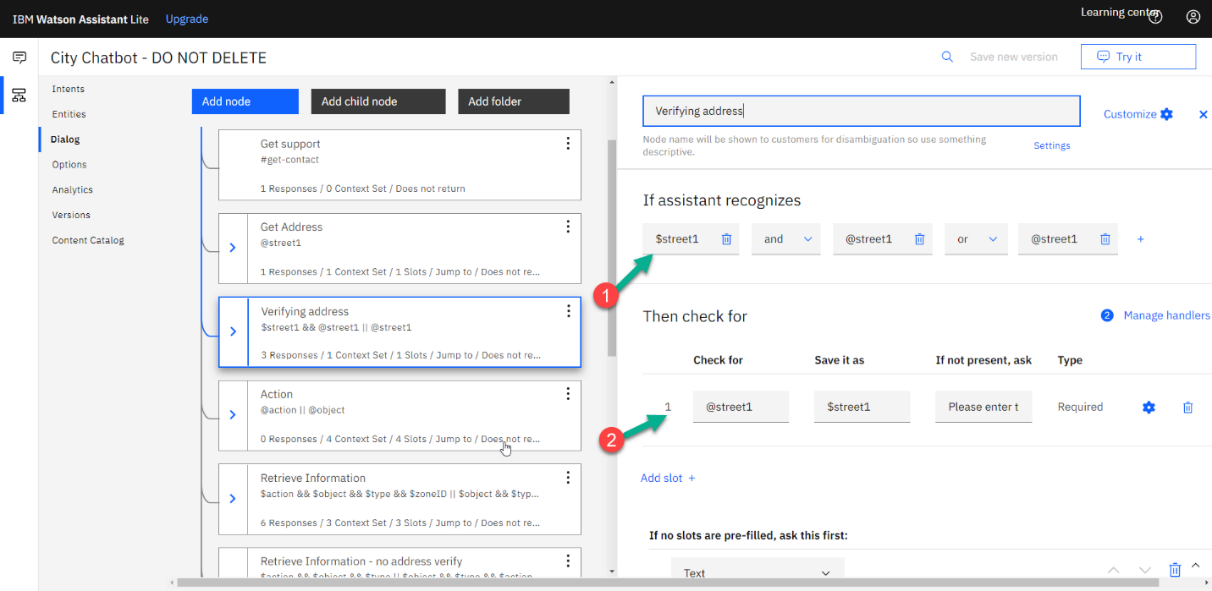


Figure Dialog Conditions

Steps for adding a webhook.

* Click on Dialog and then options
* Add/editing the url to reach out to external program.

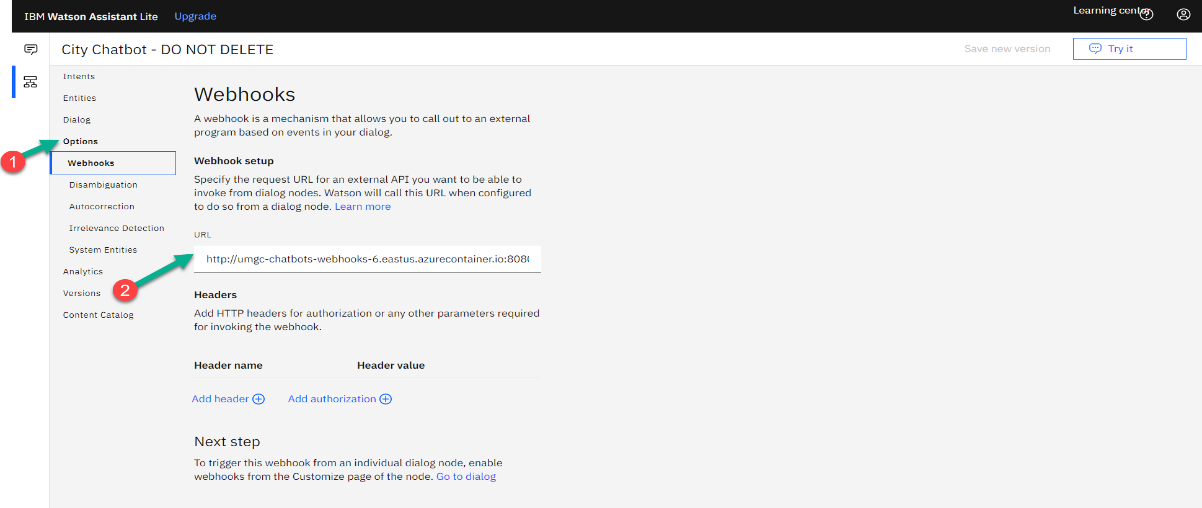


Figure Add Webhook to Dialog Node

1. Troubleshooting

### IBM Watson Assistant

IBM Watson assistant has a time limit of 8 seconds when reaching out to third-party sites for information. If a request does not complete within 8 seconds, then the chatbot assumes that the request has failed and that no information was returned. Below are some actions that can be taken to address timeouts.

* Submit request to chatbot again
* Clear user’s browser cache
* Refresh web page that user is on

For additional issues with IBM Watson Assistant, please refer to IBM’s documentation on Watson Assistant at <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started&_ga=2.21935803.1272816929.1603116070-718699761.1602794626#getting-started>.

### Spring IO REST API

Refer to the programmer’s guide for information on how the REST API is developed and refer to the deployment guide for information on how the REST API is deployed.

### MySQL Database

Refer to the programmer’s guide for information on how the MySQL database is developed and refer to the deployment guide for information on how the MySQL database is deployed.

1. Maintenance

### IBM Watson Assistant

This section will discuss basic IBM Watson Assistance maintenance and tasks. IBM Watson Assistant is deployed on IBM Cloud which handles running Watson Assistant. For any issues or assistance with maintenance, reach out to IBM Watson Assistant’s support center at <https://cloud.ibm.com/unifiedsupport/supportcenter>

### Spring IO REST API and MySQL Database

The Spring REST API and MySQL database are both deployed on Microsoft Azure. Azure is a cloud platform that handles running applications. For any issues or assistance with maintenance, reach out to Azure Support at <https://azure.microsoft.com/en-us/support/options/>

1. Technical Support

This software is free to use by anyone. It comes with no warranties and is provided solely "AS-IS". It may contain significant bugs, or may not even perform the intended tasks, or fail to be fit for any purpose. University of Maryland is not responsible for any shortcomings and the user is solely responsible for the use.

1. Acronyms and Abbreviations

|  |  |
| --- | --- |
| Acronym/Abbreviation | Definition/References |
| AI | Artificial Intelligence – An application that aims to mimic human intelligence. |
| API | Application Programming Interface |
| CH | Chatbot |
| CU | Conditional Use Permit |
| DevSecOps | Development, Security and Operations – Group of developers responsible for the deployment and security of an application. |
| ECUP | Expressive Use Permit |
| EPSP | East Pasadena Specific Plan |
| ETL | Extract, transfer, and load |
| FGSP | Fair Oaks/Orange Grove Specific Plan |
| GIS | Geographical Information System – System for working with geographical data. |
| HTTP | Hyper Text Transfer Protocol – A network protocol for specifying how servers and clients communicate with each other. |
| IDE | Integrated Development Environment |
| KML | Keyhold Markup Language – A markup language for visualizing geographical data. |
| LASP | Lincoln Avenue Specific Plan |
| MCUP | Minor Conditional Use Permit |
| MVC | Model-View-Controller |
| REST API | Representation State Transfer – An API for interacting with data. |
| SRS | Software Requirements Specification |
| UI | User Interface – The part of the application that users use to interact with the application. |
| URL | Uniform Resource Locator |
| WSL | Windows Subsystem for Linux |

1. References

* <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>