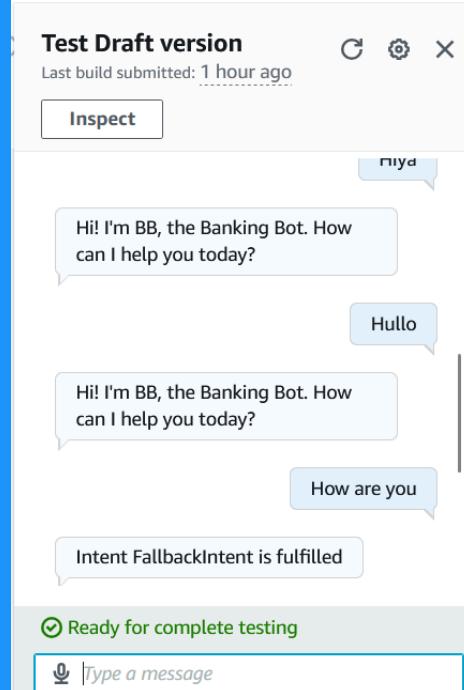




Build a Chatbot with Amazon Lex



miguelhhuerto@gmail.com



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is an AWS product that uses AI to create powerful and dynamic chatbots that can be used to enhance customer experience. This AI-powered service creates a smooth and easy experience to set-up a chatbot for applications and websites.

How I used Amazon Lex in this project

I used Amazon Lex to create a chatbot named BankerBot. This chatbot is able to answer user greetings and messages for assistance. It also displays a varied responses for error handling, enhancing overall customer experience.

One thing I didn't expect in this project was...

I didn't expect this project to be straightforward and simple. It was very enjoyable to set different configurations to personalize the chatbot.

This project took me...

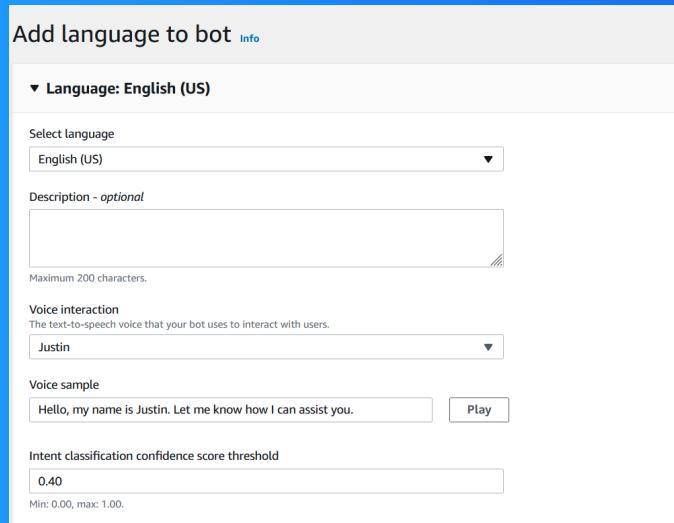
This project took me around 20 minutes with breaks.

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me around a minute or two.

While creating my chatbot, I also created a role with basic permissions because Amazon Lex needs permissions to access and use other AWS resources it will be using.

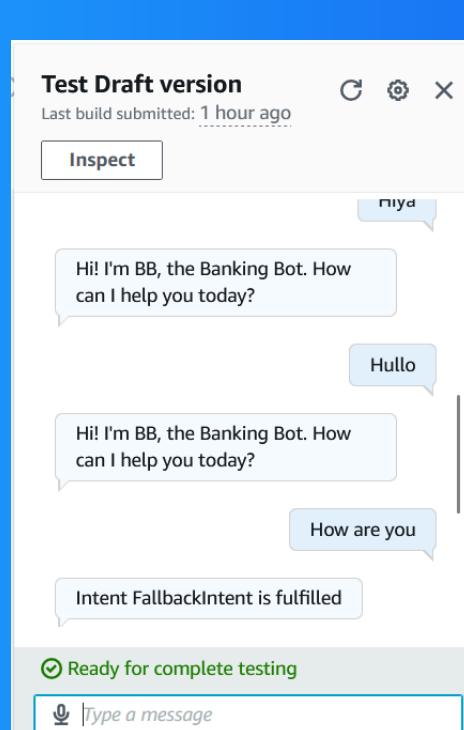
In terms of the intent classification confidence score, I kept the default value of 0.40. This means that my new chatbot needs to be at least 40% confident that it understands the user's input question to be able to provide an answer.



Intents

Intents are what the user aims to achieve in its conversation with the chatbot. Different intents can be defined and categorized to handle various requests related to each other.

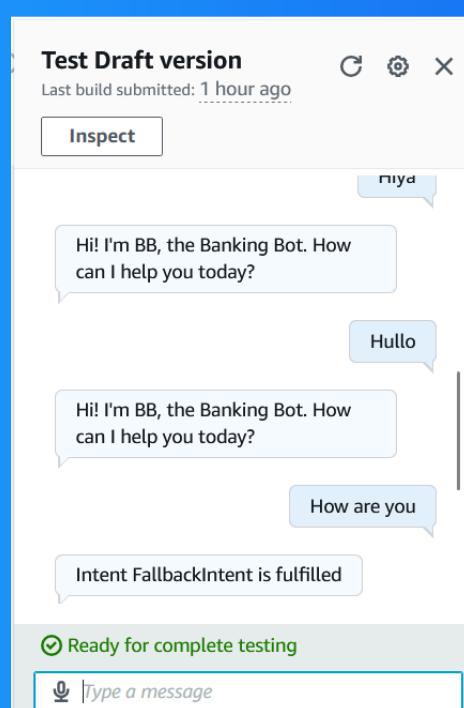
I created my first intent, WelcomeIntent, to reply to the user's greeting.



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter "Hi", "Hello", or "I Need Help"

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered "How are you" This error message occurred because Amazon Lex did not recognize the user input (utterance), and its Intent Classification was below that of 0.4



Configuring FallbackIntent

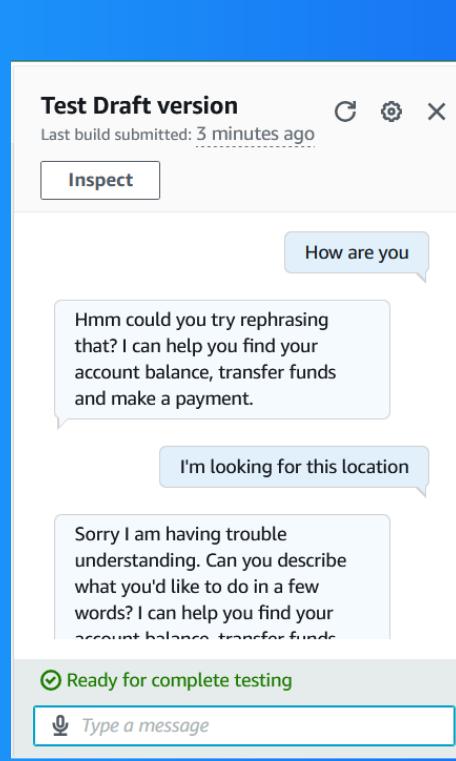
FallbackIntent is a default intent in every chatbot that gets triggered when the chatbot's Intent Classification is below 0.4 (which was set earlier) for all defined intents.

I wanted to configure FallbackIntent because it will serve as the automatic error message for when the chatbot does not understand the user input.

Variations

To configure FallbackIntent, I changed the FallbackIntent's closing message to the desired message I wanted it to display upon triggering, giving hints in the message of some keywords that the user can use for the chatbot to understand.

I also added variations! What this means for an end user is varying response for the same message, offering a dynamic user experience, and not just displaying the same error message over and over again.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

