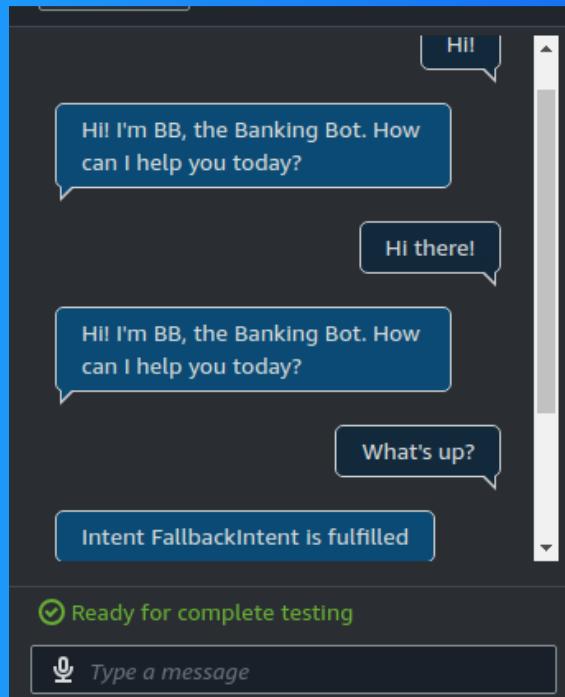




# Build a Chatbot with Amazon Lex



tskhelp@gmail.com



# Introducing Today's Project!

## What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces using voice and text. It's useful for creating chatbots, virtual assistants, and automated customer support with natural language understanding and seamless AWS integration.

## How I used Amazon Lex in this project

I created a basic chatbot with WelcomeIntent and FallbackIntent.

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

I didn't expect that you can specify not only one FallbackIntent message, but several.

## This project took me...

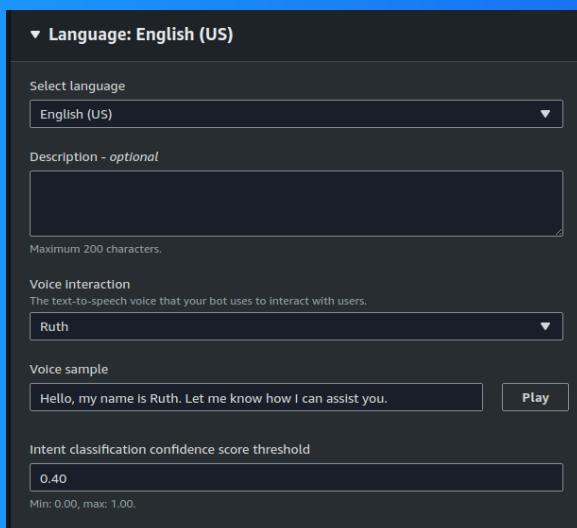
This project only took me 30 minutes to create a chatbot and learn some additional information about Amazon Lex.

# Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me just 3 minutes.

While creating my chatbot, I also created a role with basic permissions because Amazon Lex needs the permission to call other AWS services on your behalf, later I'll be integrating Lex with another service called Lambda.

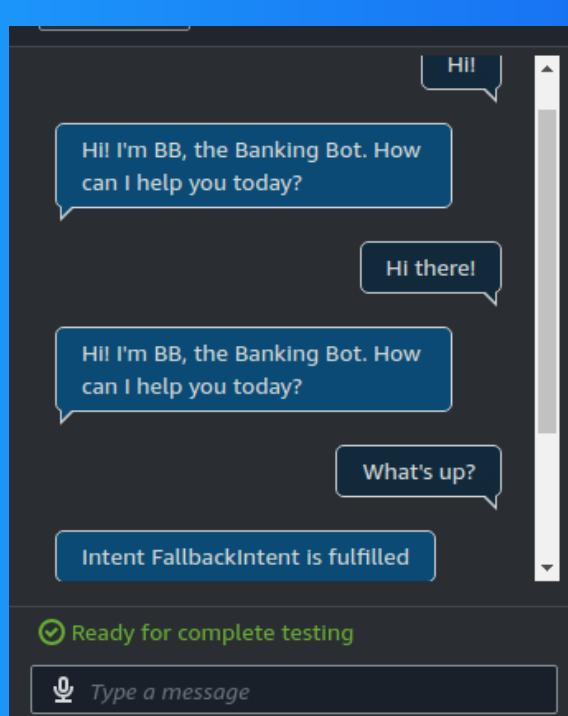
In terms of the intent classification confidence score, I kept the default value of 0.40. This means that a chatbot needs to be at least 40% confident that it understands what the user is asking to be able to give a response.



# Intents

An intent is what the user is trying to achieve in their conversation with the chatbot. For example, checking a bank account balance; booking a flight; ordering food.

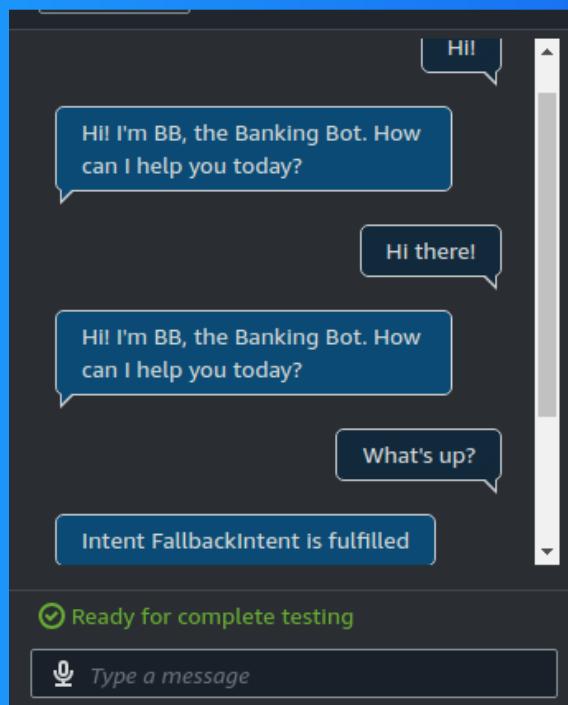
I created my first intent, WelcomeIntent, to greet a user.



# FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter 'Hi!' or 'Hi there!' Amazon Lex is able to use its ML techniques to match what I have said against my utterances.

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered 'What's up?' This error message occurred because Amazon Lex doesn't quite recognize my utterance.



# Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when a chatbot has a confidence score below 40% for all the intents you've defined.

I wanted to configure FallbackIntent because I wanted it to be clearer to the user that my chatbot doesn't understand their request.

TS

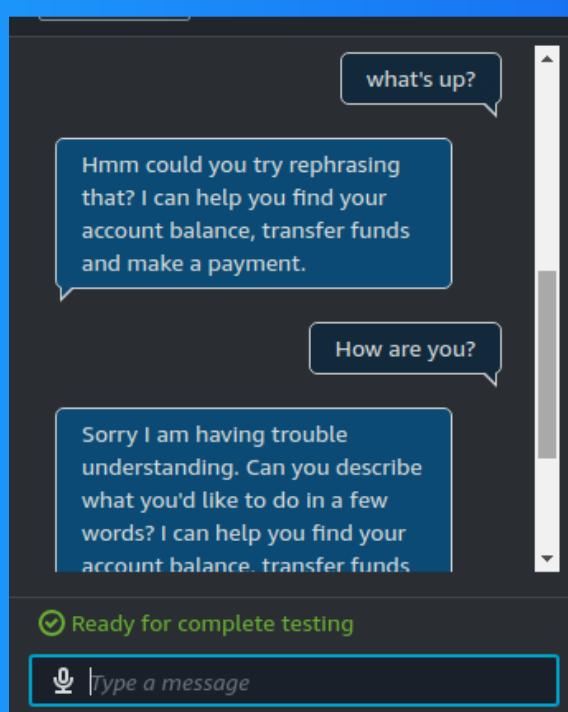
tskhelp@gmail.com  
NextWork Student

[NextWork.org](http://NextWork.org)

# Variations

To configure FallbackIntent, I chose 'FallbackIntent' in the navigation panel, scrolled down to 'Closing responses', expanded the arrow for 'Response sent to the user after the intent is fulfilled' and then in the 'Message' field I added some text.

I also added variations! What this means for an end user is my bot will give my users a dynamic range of responses, making them sound more conversational!





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

