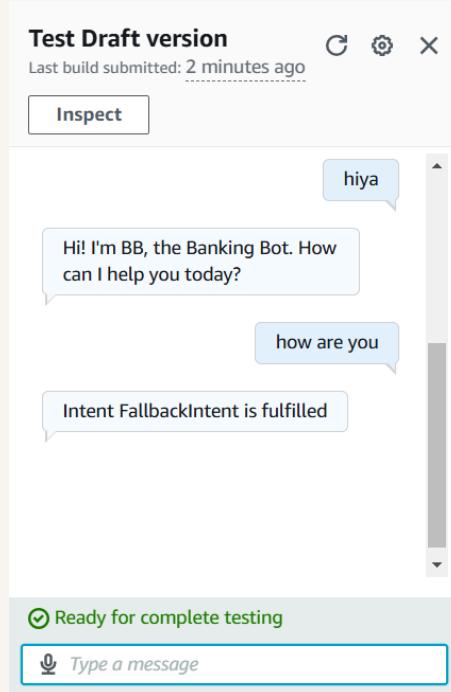


Build a Chatbot with Amazon Lex



Ece Pelin



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a fully managed service by AWS that allows you to build conversational interfaces using voice and text. It's designed to help you create chatbots, virtual assistants, and automated customer service applications.

How I used Amazon Lex in this project

In today's project, I used Amazon Lex to create a chatbot that could interact with users in a natural and intuitive way. I set up different intents, such as a WelcomeIntent for greeting users and a FallbackIntent for handling unrecognized inputs.

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

One thing I didn't expect in this project was how much fine-tuning would be required for the intent recognition.

This project took me...

This project took me around 45 minutes to complete. I spent the first part setting up the basic intents, defining sample phrases, and configuring the responses.

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me around 10 minutes to get familiar with the interface and design the basic intents and responses.

While creating my chatbot, I also created a role with basic permissions because Amazon Lex requires a role that grants it the necessary permissions to interact with other AWS services, such as Lambda functions or DynamoDB, if used.

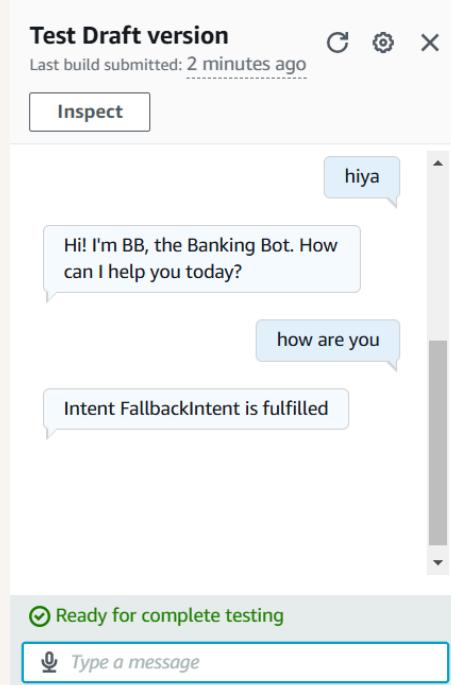
In terms of the intent classification confidence score, I kept the default value of 0.40. This means that if the chatbot's confidence in predicting the user's intent is 40% or higher, it will consider the intent to be valid.

The screenshot shows the 'Step 2 Add languages' section of the Amazon Lex configuration interface. It is titled 'Language: English (US)'. The form includes fields for 'Select language' (set to English (US)), 'Description - optional' (empty), 'Voice interaction' (set to Ruth), 'Voice sample' (a text input containing 'Hello, my name is Ruth. Let me know how I can assist you.' with a 'Play' button next to it), and 'Intent classification confidence score threshold' (set to 0.40). At the bottom, there are 'Cancel', 'Add another language', and 'Done' buttons.

Intents

Intents are the goals or purposes behind a user's input in a chatbot or conversational system. They represent what the user wants to achieve or the action they want to perform.

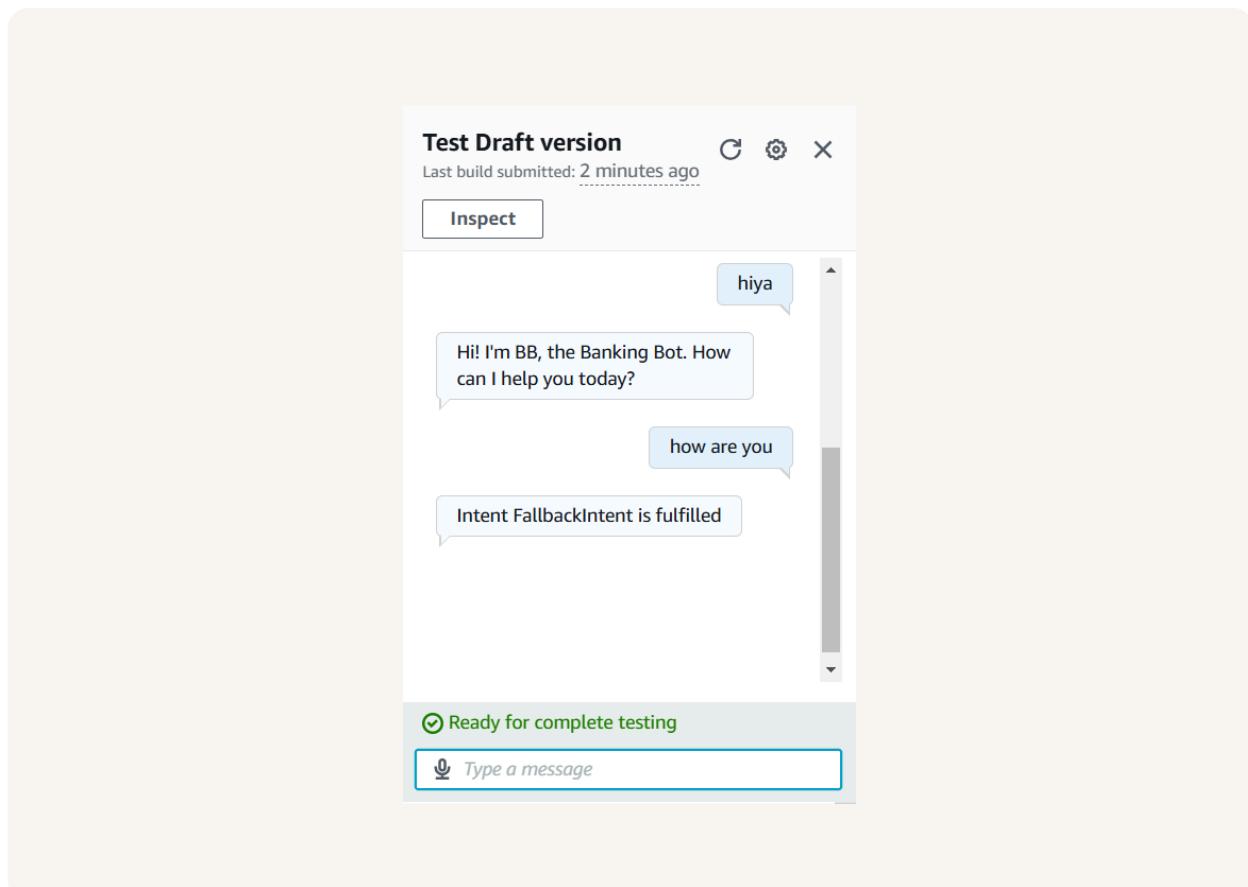
I created my first intent, WelcomeIntent, to greet users when they interact with the chatbot for the first time or after a period of inactivity.



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter greetings like "Hello," "Help me" "hiya" . These phrases were mapped to the WelcomeIntent.

My chatbot returned the error message "Intent FallbackIntent is fulfilled" when I entered a query that it didn't recognize, such as 'How are you?'. This error message occurred because the chatbot couldn't match the input to any of the intents.





Configuring FallbackIntent

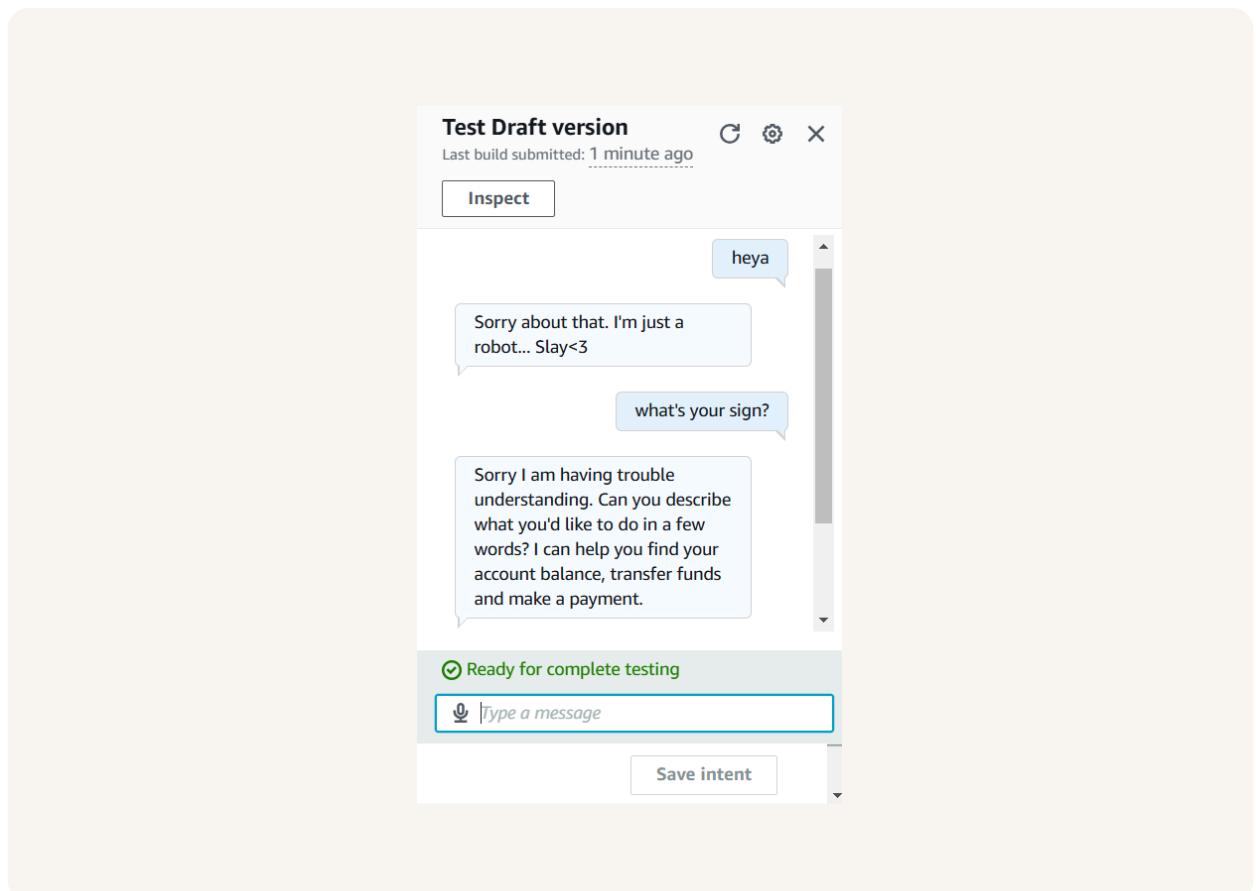
FallbackIntent is a default intent in every chatbot that gets triggered when the chatbot cannot recognize or classify the user's input into any of the predefined intents.

I wanted to configure FallbackIntent because it's essential to ensure that my chatbot can handle unexpected or unrecognized input gracefully.

Variations

I defined the intent in Amazon Lex and added sample phrases that the bot could recognize as fallback cues, like "I don't understand," or "Can you help me?" I also customized the response to guide the user, such as "Sorry, I didn't quite catch that."

I also added variations! What this means for an end user is that the chatbot can recognize different ways of phrasing the same request or greeting.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

