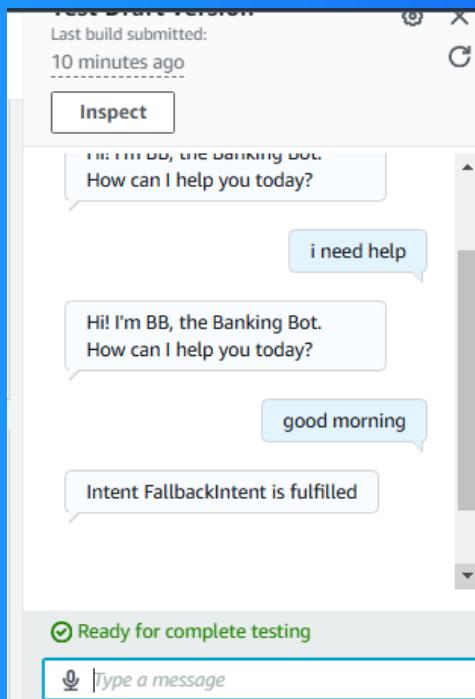




Build a Chatbot with Amazon Lex

Gloria





Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a powerful AWS service that lets you build smart chatbots and conversational interfaces with ease. It uses AI to understand user intent, supports voice and text interactions, and simplifies creating intelligent communication tools .

How I used Amazon Lex in this project

I used Amazon Lex to create an interactive chatbot capable of understanding user inputs and responding naturally. I configured intents, added sample utterances, and customized responses, including a fallback intent to handle unrecognized inputs.

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

One thing I didn't expect in this project was how simple and fast the process without any extra knowledge in coding and the rest. I always wondered how Bots is set up and its operations , and this project satisfied my curiousity.

This project took me...

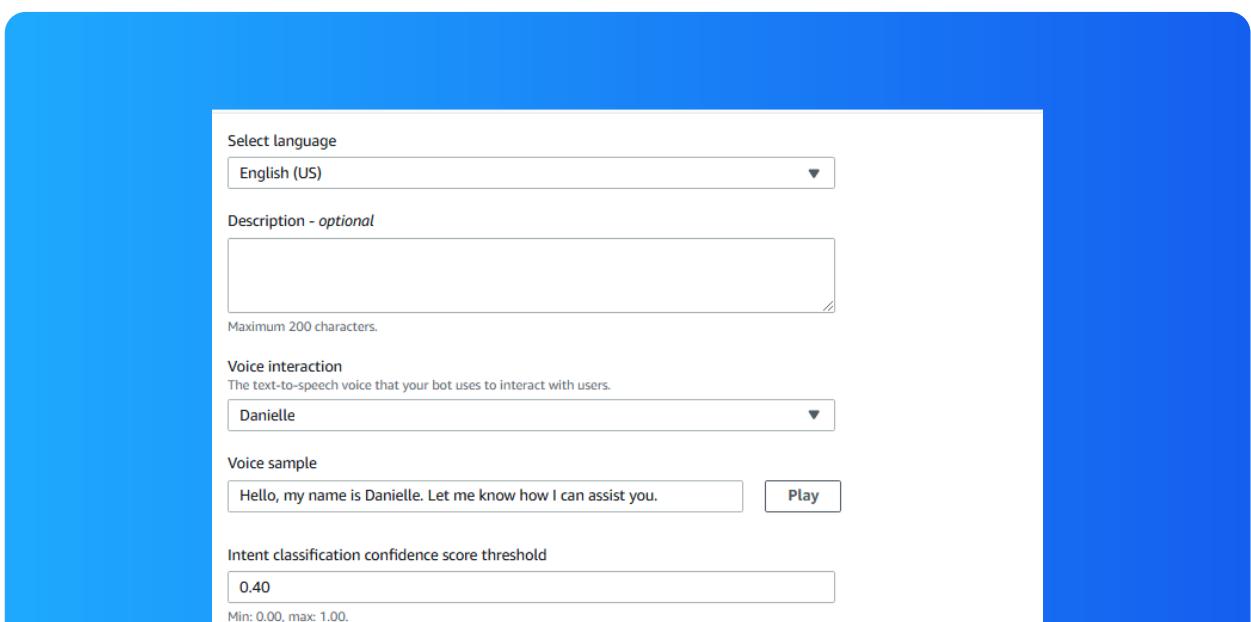
The project took 20 minutes because it was easier to executes the task and the Amazon Lex platform was also easy to interact with and understand.

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me 2 minutes including the features such as utterances. It was straightforward thanks to Lex's intuitive interface, which made the process efficient and seamless.

While creating my chatbot, I also created a role with basic permissions because it ensures the chatbot can securely access essential AWS services without exposing unnecessary resources or creating security risks

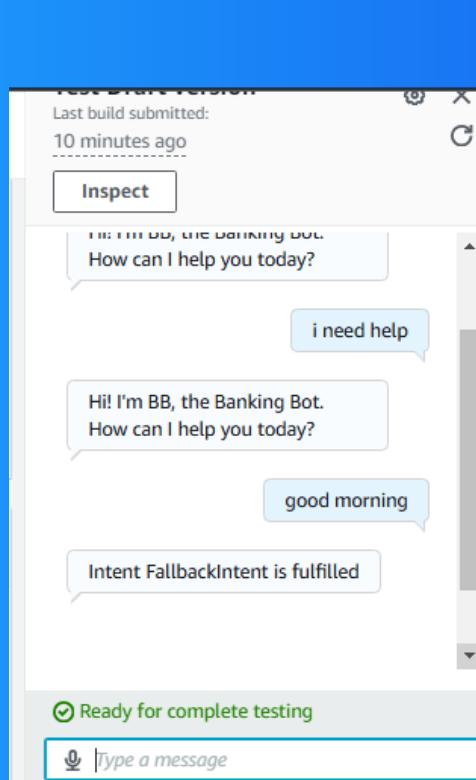
In terms of the intent classification confidence score, I kept the default value of 0.40. This means the bot is like a careful translator who only speaks up when they're more than 40% sure what you mean, avoiding wild guesses and uncertain answers



Intents

Intents are the core components of a chatbot that define the purpose of a user's interaction. Each intent represents a specific action or goal, such as making enquiries, and includes sample utterances and slots to guide the conversations.

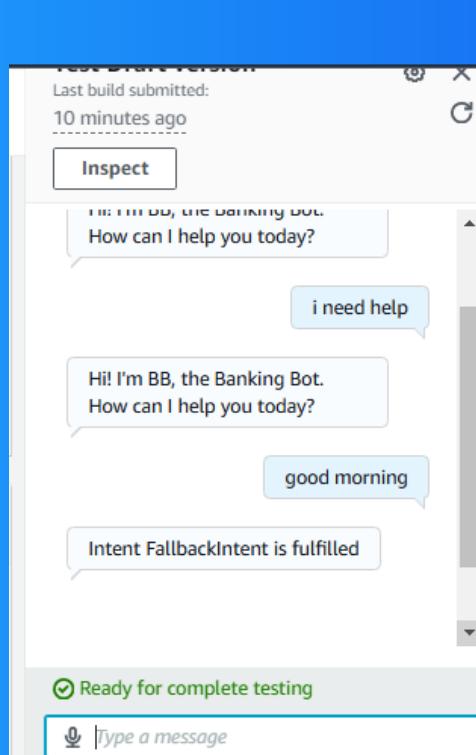
I created my first intent, WelcomeIntent, to welcome users when they greet the chatbot or request assistance. It triggers on inputs like Hi , I need help and responds with "Hi! I'm BB, the Banking Bot. How can I help you today?" making interaction fun



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter greetings like "Hi," "Hello," "I need help," or "Can you help me?" These utterances trigger the WelcomeIntent and ensure a welcoming and helpful response.

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered "good morning." This error message occurred because the utterance "good morning" was not included in the defined sample utterances for the WelcomeIntent.





Configuring FallbackIntent

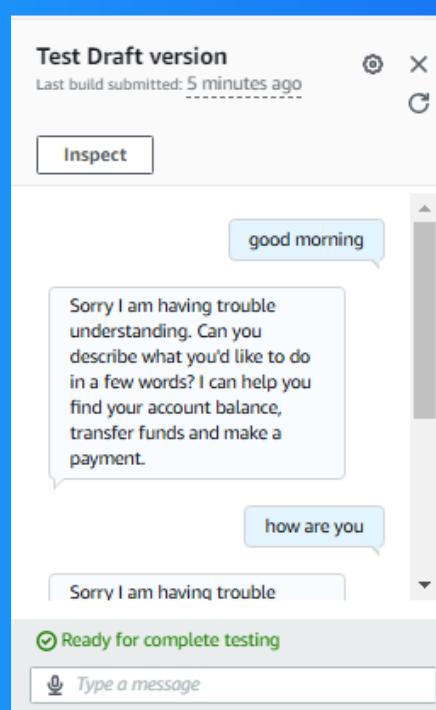
FallbackIntent is a default intent in every chatbot that gets triggered when the user's input does not match any of the defined intents or sample utterances. It acts as a safety net to handle unrecognized inputs and provide a default response.

I wanted to configure FallbackIntent because it's a conversational safety net. When the AI can't understand an input, it provides a friendly, helpful response instead of leaving users confused.

Variations

To configure FallbackIntent, I customized its response so it could be user friendly and have a natural instead of the sophisticated response it generated earlier

I also added variations! What this means for an end user is that the chatbot can recognize and respond to multiple ways of saying the same thing, improving the user experience by understanding diverse inputs and making interactions feel more natural.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

