



nextwork.org

Build a Chatbot with Amazon Lex



Leon Williams

The screenshot shows the AWS Lex console interface. At the top, a green banner indicates "Successfully built language English (US) in bot: BankerBot". Below this, the navigation path is Lex > ... > Language: English (US) > Intents > Intent: WelcomeIntent. The status bar shows "Draft version", "English (US)", and "Successfully built". On the left, the "Conversation flow" for the "WelcomeIntent" is displayed, showing three steps: "Initial request - sample utterance" (e.g., "Hi"), "Acknowledge intent - initial response" (e.g., "Hello!"), and "Prompt for more information - slot" (e.g., "e.g. ABC123"). On the right, the "Test Draft version" pane shows a transcript of a test session with the bot responding to user inputs like "Hello!", "Can I go af...", "hi", "hiya", and "good morning". A message at the bottom of the transcript says "Intent FallbackIntent is fulfilled". The bottom of the screen features tabs for "Editor", "Visual builder", and "New", along with a "Save intent" button.



Leon Williams
NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces using voice and text. It enables developers to create chatbots that understand natural language and interact with users intelligently, powered by the same technology as Alexa.

How I used Amazon Lex in this project

In this project, I learned key services like Amazon Lex for building chatbots, and concepts such as intents, utterances, fallback intents, confidence scores, and message variations to create natural, responsive conversations.

One thing I didn't expect was...

One thing I didn't expect in this project was how well Amazon Lex's machine learning can recognize different ways users express the same intent, allowing the chatbot to respond accurately even to unexpected phrases.



Leon Williams
NextWork Student

nextwork.org

This project took me...

This project took me approximately one hour to complete. The most challenging part was fixing my mistakes while fully understanding the material. It was most rewarding to see the chatbot respond correctly and interact naturally.



Leon Williams
NextWork Student

nextwork.org

Setting up a Lex chatbot

It only took me a few minutes to create my chatbot. From naming it BankerBot to choosing a voice and setting permissions, the process was quick and gave me a smooth intro to building AI-driven tools in the cloud.

I gave my chatbot basic Amazon Lex permissions so it can call other AWS services like Lambda. This allows the bot to handle tasks such as checking balances or making transfers in later steps of the project.

I kept the default intent classification confidence score at 0.40, meaning the chatbot must be at least 40% confident it understands the user's input to respond. If confidence is below 0.40, it returns an error message.



Leon Williams
NextWork Student

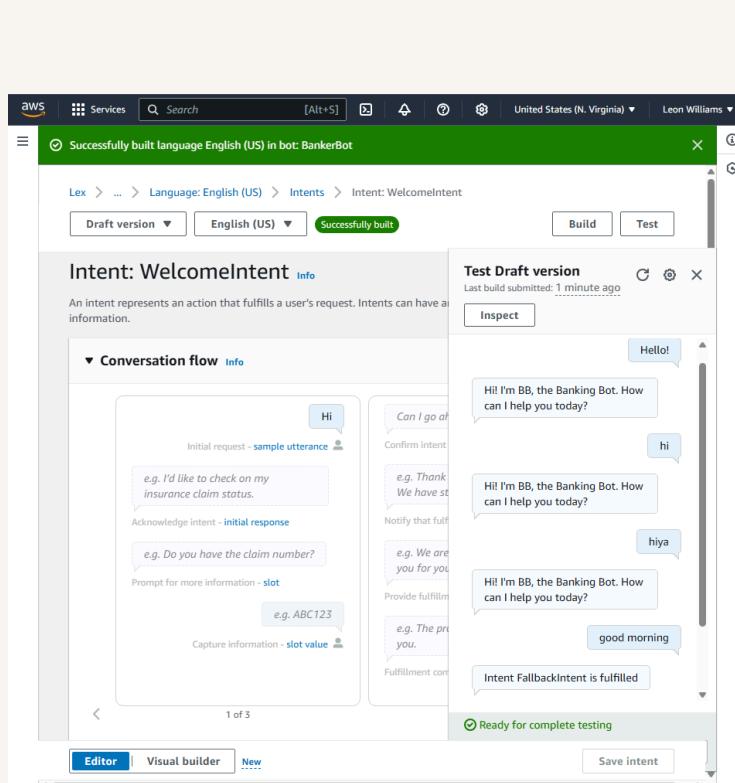
nextwork.org

The screenshot shows the AWS Lex 'Add language to bot' configuration interface. At the top, it says 'Language: English (US)'. Under 'Select language', 'English (US)' is selected. There is a 'Description - optional' field with placeholder text 'Maximum 2000 characters.' Below it, under 'Voice interaction', 'Joanna' is selected from a dropdown. A 'Voice sample' section shows a preview of the voice saying 'Hello, my name is Joanna. Let me know how I can assist you.' A 'Play' button is next to the preview. Under 'Intent classification confidence score threshold', the value '0.40' is entered. At the bottom right, there are 'Cancel', 'Add another language', and a highlighted 'Done' button.

Intents

Intents represent what the user wants to achieve in a conversation with the chatbot, like checking a bank balance or ordering food. In Amazon Lex, defining multiple intents lets one bot handle various related requests smoothly.

WelcomeIntent handles greeting users when they say hello or ask for help. It recognizes phrases like "Hi," "Hello," or "Can you help me?" and responds with a friendly message to start the conversation.



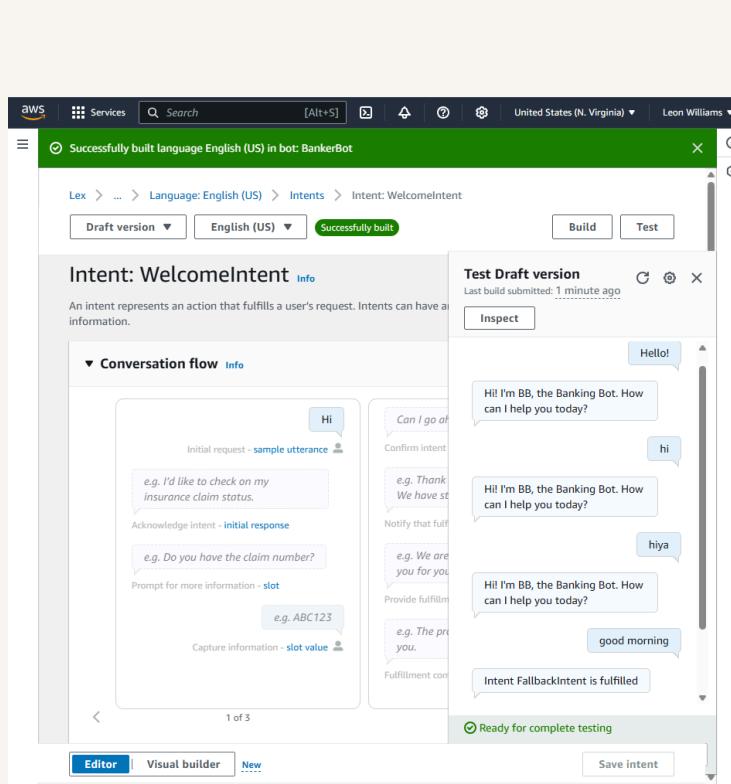
Leon Williams
NextWork Student

nextwork.org

FallbackIntent

I launched and tested my chatbot, which responded to "Hi," "Hello," "I need help," and "Can you help me?" It also recognized similar phrases like "Help me," "Hiya," and "How are you," but didn't recognize "Good morning," causing a fallback response.

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered phrases like "Good morning." This error message occurred because the input didn't match any defined utterances or meet the confidence threshold.





Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when the chatbot's confidence score for all defined intents falls below 40%. It lets the bot respond with a helpful message when it doesn't understand the user's input.

I configured FallbackIntent to provide user-friendly messages when the chatbot doesn't understand an input. This improves the user experience by guiding users on what the bot can help with instead of returning confusing errors.

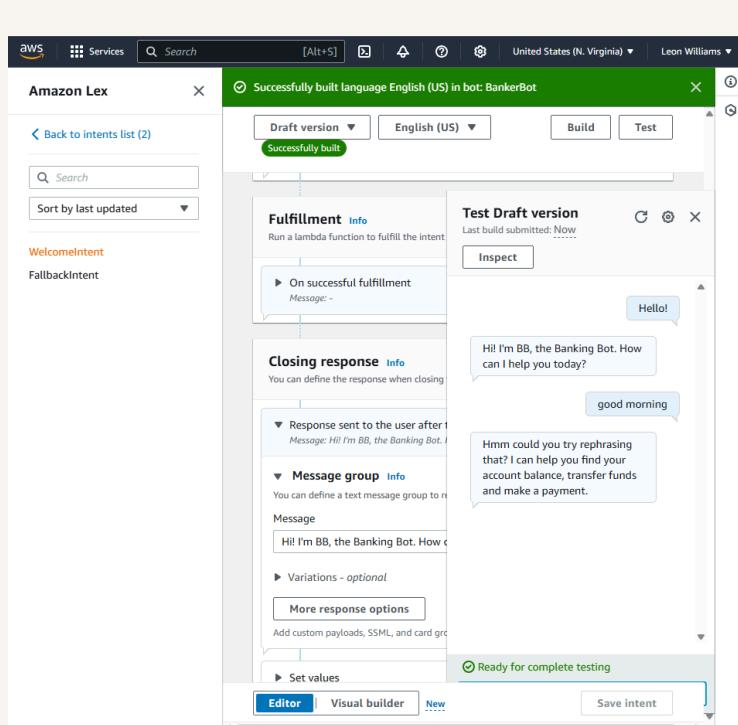
Leon Williams
NextWork Student

nextwork.org

Variations

To configure FallbackIntent, I edited its response messages to include friendly, helpful text that guides users when the bot doesn't understand their input. I added message variations to make the replies feel more natural and conversational.

Variations are different versions of the same response message that the chatbot can randomly choose from when replying. They help make the chatbot sound more natural and conversational by avoiding repetitive replies.





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

