Build a Chatbot with Amazon Lex

Introducing Today's Project!

What is Amazon Lex?
Amazon Lex is a managed AWS service for building conversational interfaces using natural language processing and speech recognition. It enables text and voice-based bots for customer service and automation.

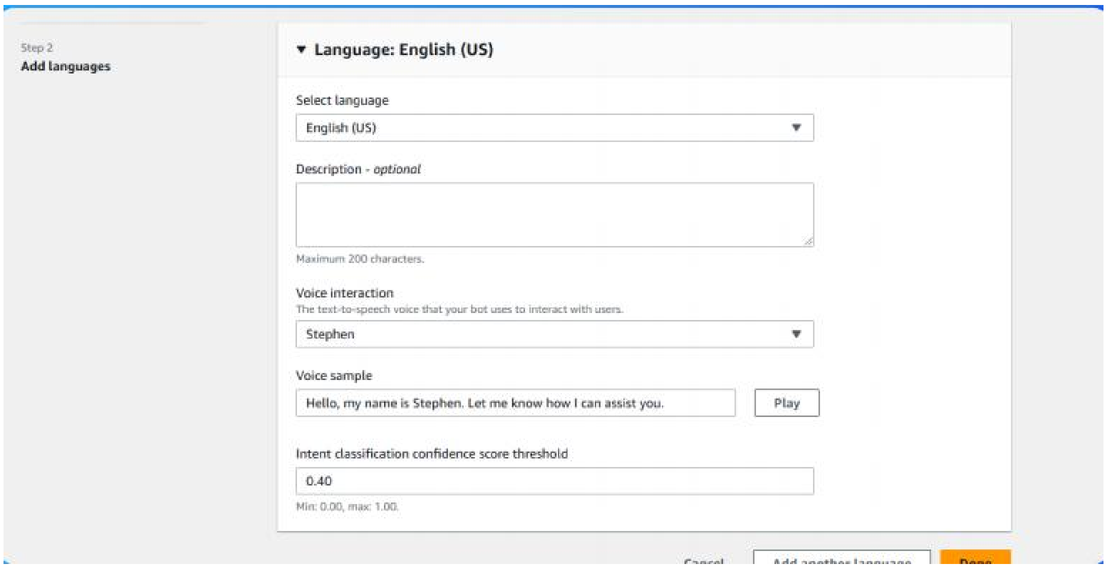
How I used Amazon Lex in this project
In today's project I used Amazon Lex for a banking bot, which assists users for banking related queries in user friendly way.

One thing I didn't expect in this project was...
One unexpected challenge when building a chatbot with Amazon Lex is training it to handle diverse user inputs. Variations in slang, and errors require continuous refinement of intents and responses to ensure accurate and effective interactions.

This project took me...
It took me a hour to complete this project.

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me just few mins. While creating my chatbot, I also created a role with basic permissions because permissions should always follow the principle of least privilege—giving the bot only the permissions it absolutely needs to function, nothing more. In terms of the intent classification confidence score, I kept the default value of 0.40. This means If the system gives a lower score, it might indicate the input is unclear or matches multiple possible intents with similar probability



▼ Language: English (US)

Select language: English (US)

Description - *optional*

Maximum 200 characters.

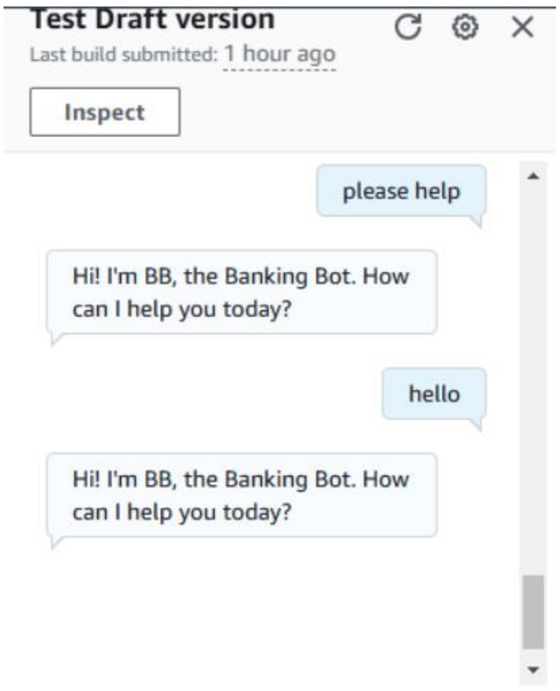
Voice interaction
The text-to-speech voice that your bot uses to interact with users.
Stephen

Voice sample: Hello, my name is Stephen. Let me know how I can assist you.

Intent classification confidence score threshold: 0.40
Min: 0.00, max: 1.00.

Intents

Intents are the goals or actions a user wants to achieve when interacting with a chatbot or virtual assistant. They define what the system should do in response to user input, such as providing weather updates, or answering a question.
I created my first intent, WelcomeIntent, to greet or welcome users when they first interact with the bot. It is typically triggered when a user starts a conversation, often by sending a greeting message like "Hello" or "Hi.”


Test Draft version

Last build submitted: 1 hour ago

Inspect

please help

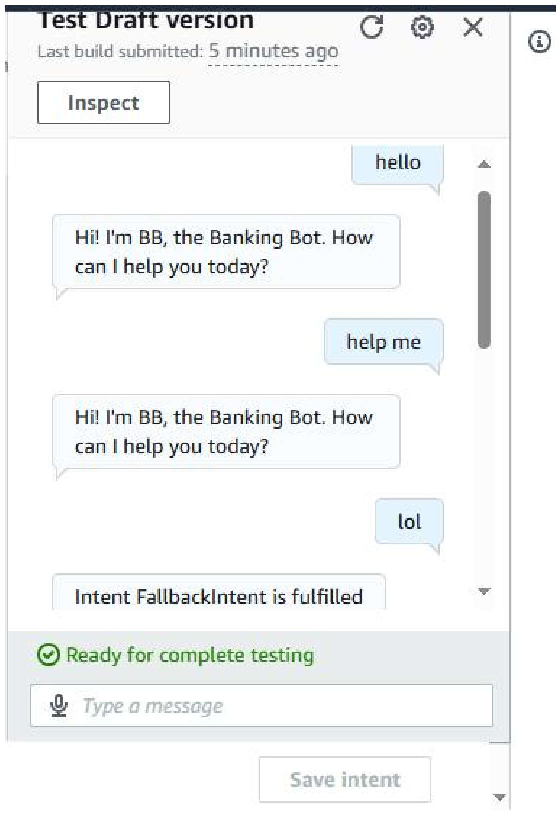
Hi! I'm BB, the Banking Bot. How can I help you today?

hello

Hi! I'm BB, the Banking Bot. How can I help you today?

FallbackIntent

'I launched and tested my chatbot, which could respond successfully if I enter hi, hello, help etc. My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered good morning, how are you, etc. This error message occurred because Amazon Lex doesn't quite recognize the utterance.


Test Draft version

Last build submitted: 5 minutes ago

Inspect

hello

Hi! I'm BB, the Banking Bot. How can I help you today?

help me

Hi! I'm BB, the Banking Bot. How can I help you today?

lol

Intent FallbackIntent is fulfilled

Ready for complete testing

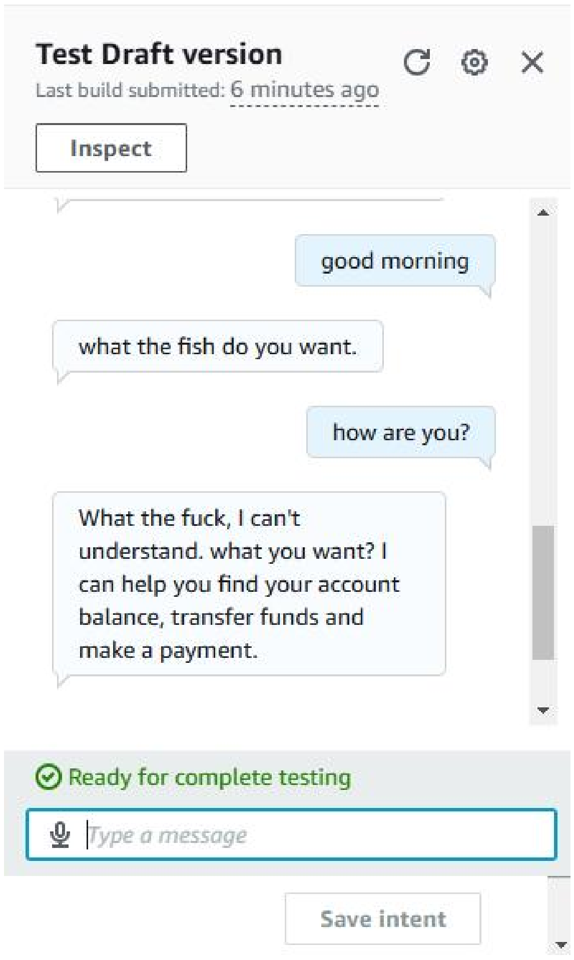
Type a message

Save intent

Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when the bot doesn't recognize user input or can't match it to any intent, prompting the user to clarify or rephrase.

'I wanted to configure FallbackIntent because to handle unrecognized or ambiguous user inputs. It ensures the chatbot can prompt the user to clarify or rephrase, maintaining conversation flow and improving user experience.



Test Draft version

Last build submitted: 6 minutes ago

good morning

what the fish do you want.

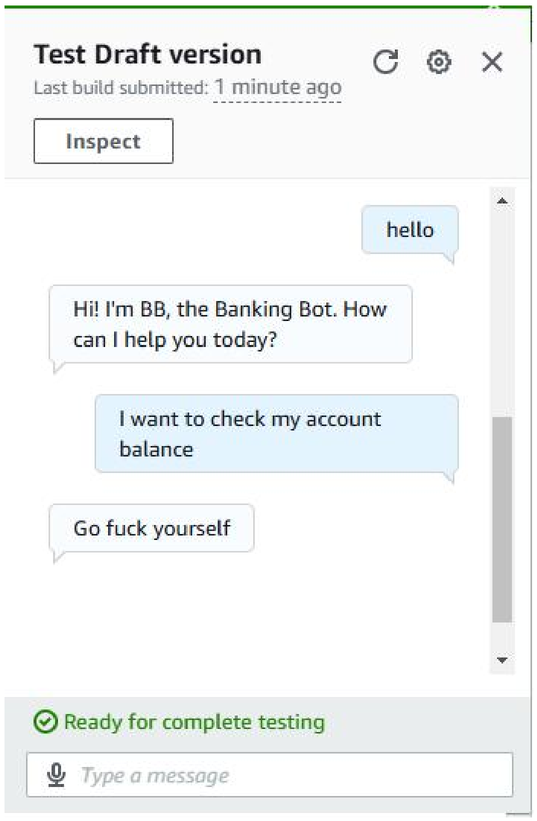
how are you?

What the fuck, I can't understand. what you want? I can help you find your account balance, transfer funds and make a payment.

Ready for complete testing

Variations

To configure FallbackIntent, I navigated to FallbackIntent -> Closing responses -> Response sent to the user after the intent is fulfilled. -> Message " customize the message" -> Save intent -> build.
I also added variations! What this means for an end user is When Amazon Lex needs to return a Fallback response, it will randomly choose a message from the group and return that. Variations also offer diverse, conversational responses for users.



Test Draft version

Last build submitted: 1 minute ago

hello

Hi! I'm BB, the Banking Bot. How can I help you today?

I want to check my account balance

Go fuck yourself

Ready for complete testing

To wrap things up, today we've learnt how to:

1. **Define a basic intent:** You configured a basic intent in Amazon Lex to control how your chatbot responds to user inputs.

2. **Create lists of utterances:** You compiled a list of sample utterances that trigger different intents, which means your chatbot can recognize different user phrases and respond.

3. **Handle failures with FallbackIntent:** You edited FallbackIntent to handle unrecognized or poorly understood user inputs.

4. **Define variations to provide semi-random responses:** You designed a MessageGroup of variations, which makes your chatbot sound more natural and conversational.

5. **Build and test your bot:** You tested your bot's setup in both text and voice formats. This comprehensive testing helps you confirm that your chatbot is ready for real-world interactions!