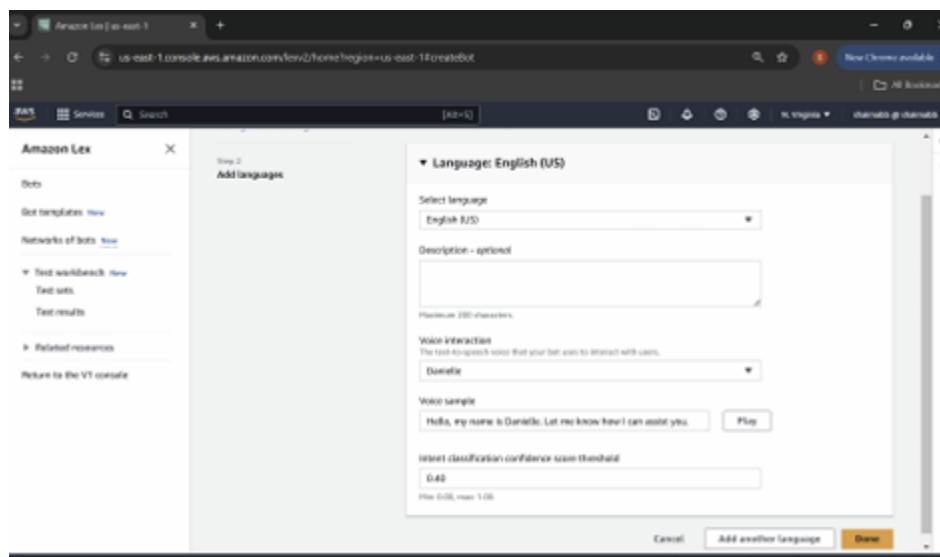


Banker-Chatbox-AWS

I created a conversational interface using Amazon Lex to handle user intents such as greetings, checking account balances, and transferring money. I used Amazon Lex as the core platform for natural language understanding and dialog management, enabling seamless user interactions. To process user requests and provide dynamic functionality, I integrated AWS Lambda for backend logic. I also incorporated context tags to maintain conversational memory, allowing the chatbot to manage multi-turn conversations effectively. Additionally, I implemented customized fallback responses to enhance the user experience by providing tailored guidance when the chatbox could not understand a query.

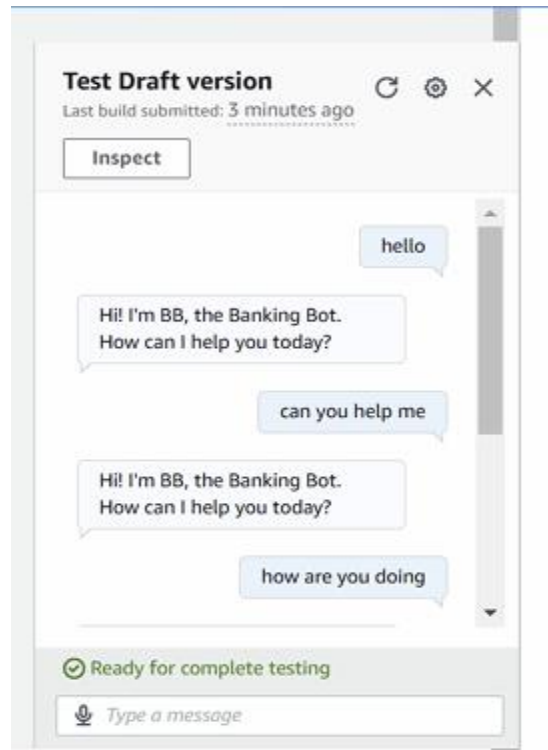
Setting up a Lex Chatbox

While creating my chatbot, I also created a role with basic permissions because this will create a policy to call other AWS services on your behalf. In terms of the intent classification confidence score, I kept the default value of 0.40. This means my chat box is 40% confident to understand what the user's goal is.



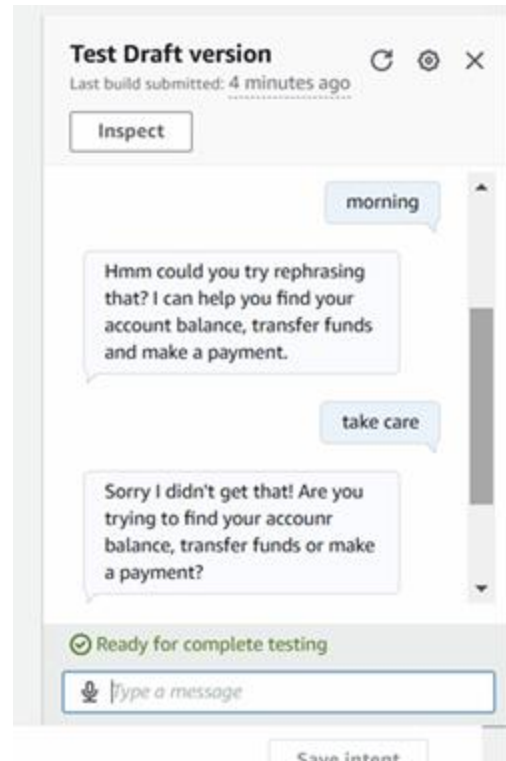
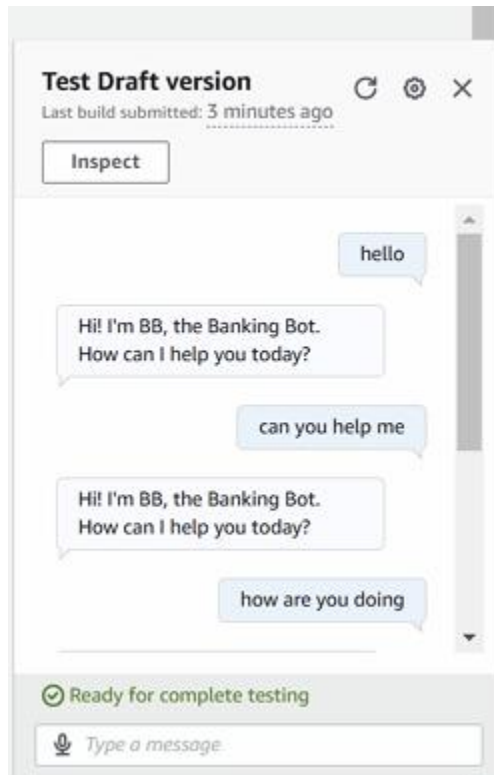
Intents

Intents are what the user is trying to achieve in their conversation with the chatbox. In Amazon Lex, a chatbox is defined by the intents that it supports. I created my first intent, WelcomeIntent, to greet the user when they say Hello.



FallbackIntent

It could understand all recorded texts I set up in the Sample utterances. My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered. This error message occurred because I entered a text which is not recorded in our Sample Utterances, as the AI cannot comprehend it.



To configure FallbackIntent, I had to create my own closing response in the intent's set up page "Sorry I am having trouble understanding. Can you describe what you'd like to do in few words?". I also added variations! What this means for an end user is they get to see different forms of the my chatbox's closing response.

Context Tags

Context tags are tools for Amazon Lex to remember specific pieces of information gathered from a conversation, and reuse that information throughout the session with its user. There are two types of context tags: they are output context tags and input context tags. I created an output context tag called contextCheckBalance, and I created this in the intent CheckBalance.

Add new context tag ×

Context tag name

contextCheckBalance

Expires after

5

turns, or

90

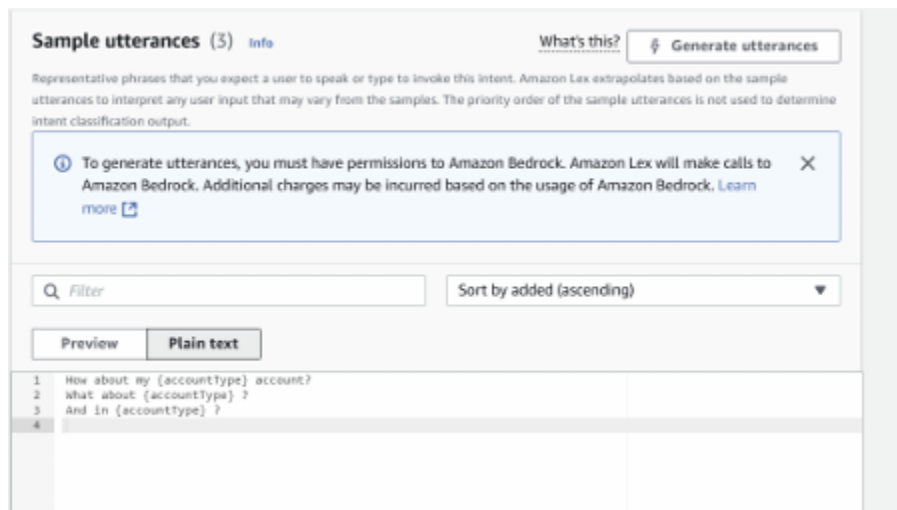
seconds

Cancel

Add

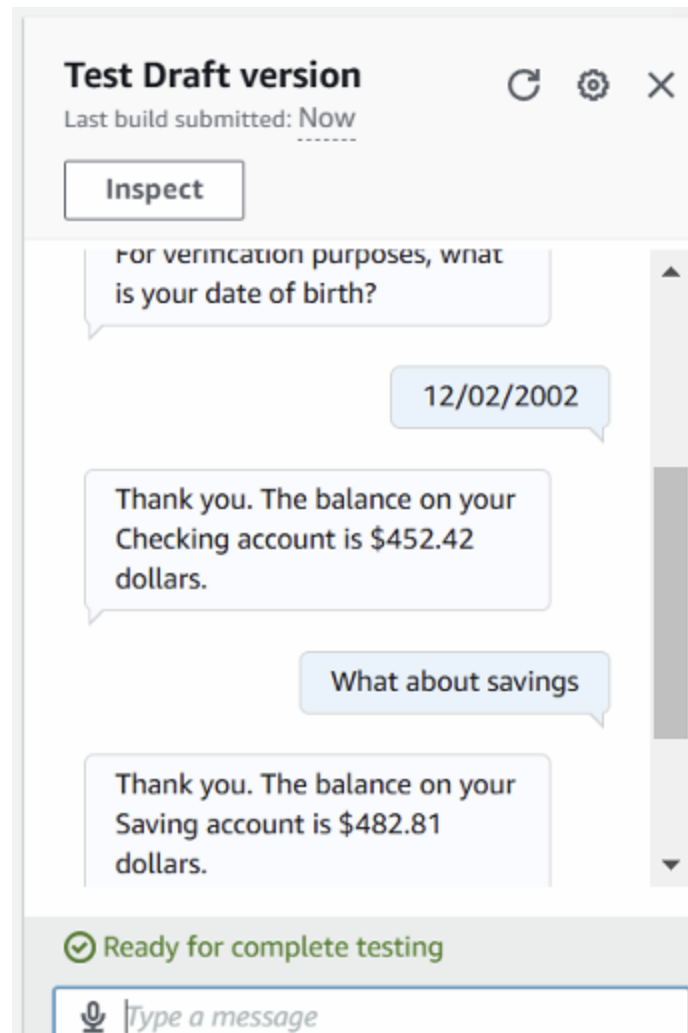
FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to handle follow up questions without asking for the user's birthday again. This intent is connected to the previous intent I made, CheckBalance, because FollowUpCheckBalance will only get triggered after the user has checked their balance once already (i.e. triggered CheckBalance).



I created an input context, contextCheckBalance, using the exact same tag as the output context tag. I've setup in the CheckBalance intent i.e. input information we are looking for in this intent (FollowUpCheckBalance) can be retrieved through this tag.

To see the context tags and follow up intent in action, I asked an utterance question to show me the balance of my 'savings' account than my previously 'checking' account without triggering the birthday message. If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, my chatbox would not have the context needed to fulfil the conversation. As result, we'll get FallBackIntent that is let user know it doesn't understand request.



I set up the TransferFunds intent to handle transferring money between accounts (Savings account to Checking). Within this intent, I defined the input information needed for the transfer and ensured it could be retrieved using a FollowUpTransferFunds context tag. This tag allows the chatbot to maintain continuity in multi-turn conversations and efficiently manage user inputs related to transferring funds.

Here the default savings account had \$2,000, the checking account balance was \$1,000, and \$20 was transferred from savings to checking.

Test Draft version

Last build submitted:

13 minutes ago



Inspect

Savings to checking

You want to transfer 20 from
your Savings to your Checking.
Is that correct?

yes

Transfer of \$20.00 from
Savings to Checking was
successful. New balances:
Savings - \$1980.00, Checking -
\$1020.00.

✓ Ready for complete testing



Type a message