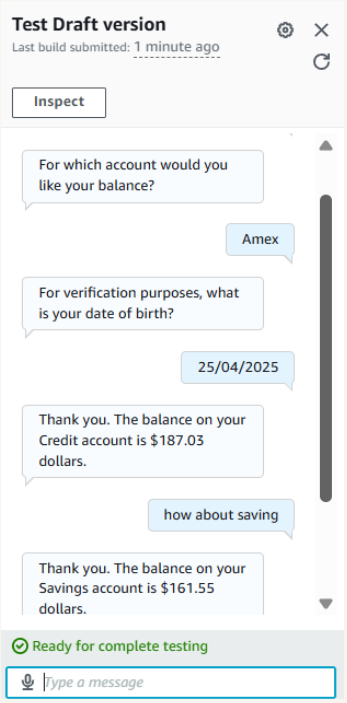
nextwork.org

Save User Info with a Lex Chatbot

lue



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is an AWS service to build chatbots using voice or text. It understands natural language, manages conversations, and integrates with AWS services, making chatbot creation easier and scalable for customer support and automation.

How I used Amazon Lex in this project

I used Amazon Lex to create a CheckBalance intent that saves the user's date of birth in a context tag. This lets the FollowupCheckBalance intent access that info, so the bot remembers details and avoids asking the user repeatedly in the same session

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

One thing I didn't expect was how easily Amazon Lex handles context tags to remember user info across intents. It made the chatbot feel much smarter and saved users from repeating themselves, which really improved the overall conversation flow.

This project took me...

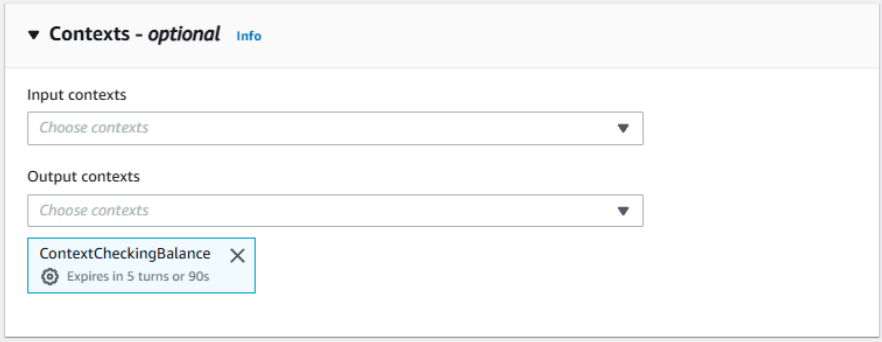
30 mins

Context Tags

Context tags are variables in Amazon Lex that help track the conversation's state. They control which intents can be triggered next by setting or checking these tags, enabling more natural, guided, and context-aware interactions in the chatbot.

There are two types of context tags: output context tags, which the bot sets to mark the current state, and input context tags, which must be active for an intent to trigger. They help manage conversation flow by controlling when intents can be used.

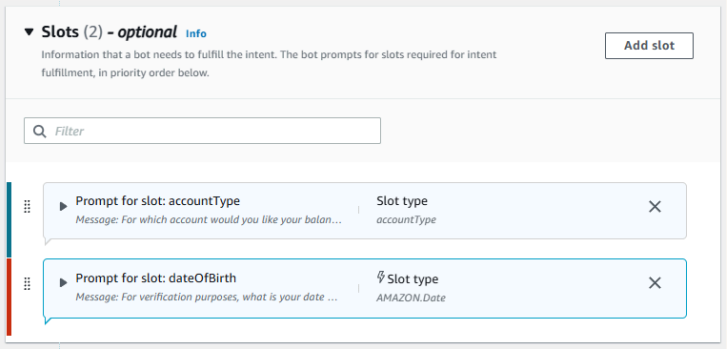
I created a context tag called "contextCheckBalance." This context tag was created in the intent "CheckBalance." It stores information about when a user is checking their balance, helping the bot manage the conversation flow for that task.



FollowUpCheckBalance

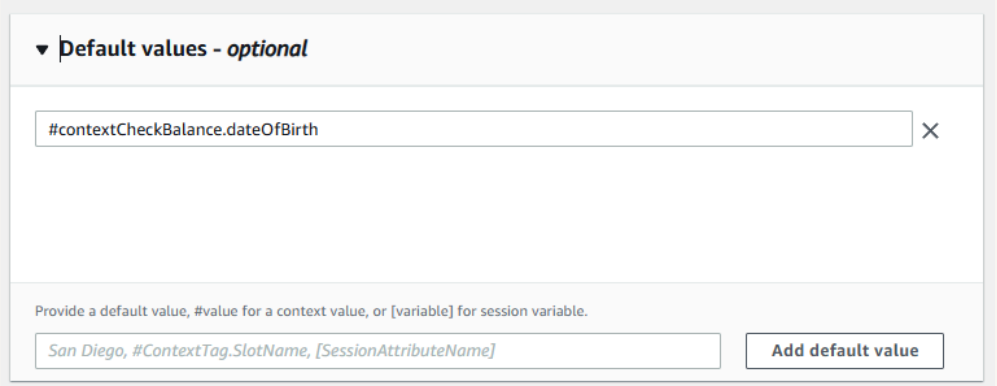
I created a new intent called FollowupCheckBalance. The purpose of this intent is to handle any follow-up questions a user might have after checking their balance, like asking for recent transactions or more account details.

This intent is connected to the previous intent I made, CheckBalance, because it uses the context tag set by CheckBalance. This allows FollowupCheckBalance to handle follow-up questions smoothly within the same conversation flow.



Input Context Tag

I created an input context, contextCheckBalance, that connects to the output context from CheckBalance. This lets FollowupCheckBalance access saved info, like dateOfBirth, so users don't need to repeat details in follow-up questions.



The final result!

To see the context tags and followup intent in action, I first asked my chatbot, “What’s my account balance?” Then, I followed up with, “What about my savings account?” to trigger the FollowupCheckBalance intent without repeating my date of birth.

If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, the chatbot wouldn’t have the needed info, like my date of birth, and would prompt me for verification again or not activate the intent at all.