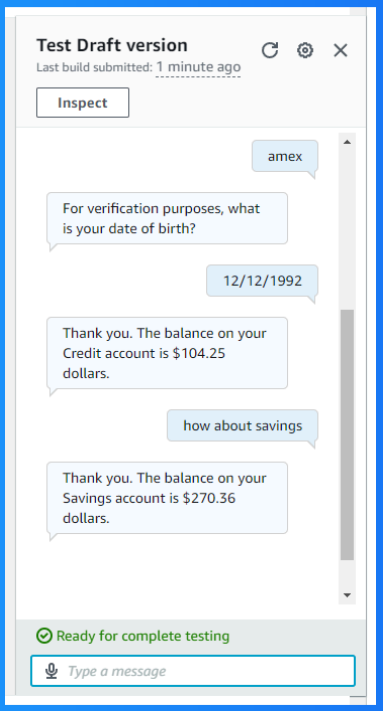
NextWork.org

Save User Info with your Chatbot

Sai Prasanth Reddy



Sai Prasanth Reddy
NextWork Student

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building chatbots using voice and text. It's useful for creating conversational interfaces, automating tasks, and integrating with AWS services, making it ideal for customer support and interactive applications.

How I used Amazon Lex in this project

I used Amazon Lex in today's project to create a chatbot for checking bank balances, set up intents like FollowupCheckBalance, configured context tags to manage follow-ups, and integrated Lambda for dynamic responses.

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

One thing I didn't expect in this project was how essential context tags are for managing follow-up interactions, ensuring the chatbot remembers previous actions and smoothly handles subsequent user requests.

This project took me...

This project took me about 45 mins to complete, including configuring intents and context tags, integrating Lambda, and testing the follow-up interactions with the context tags.

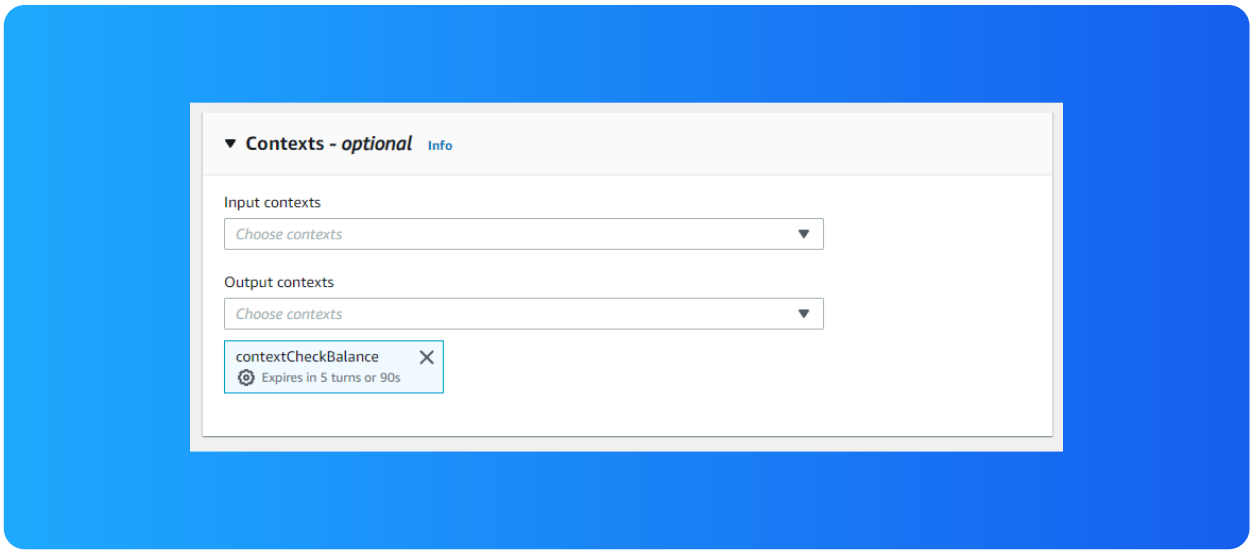
Sai Prasanth Reddy
NextWork Student

Context Tags

Context tags are labels in Amazon Lex that manage the flow of conversations by tracking the state. They help enable or disable specific intents or responses based on the current context, allowing for more dynamic and relevant user interactions.

There are two types of context tags: Input and Output. Input context tags activate an intent when a condition is met, while Output context tags are set after an intent is fulfilled, controlling which intents can be triggered next.

I created a context tag called contextCheckBalance. This context tag was created in the CheckBalance intent. This tag stores information about the user's request to check their account balance and helps manage the conversation flow for further action

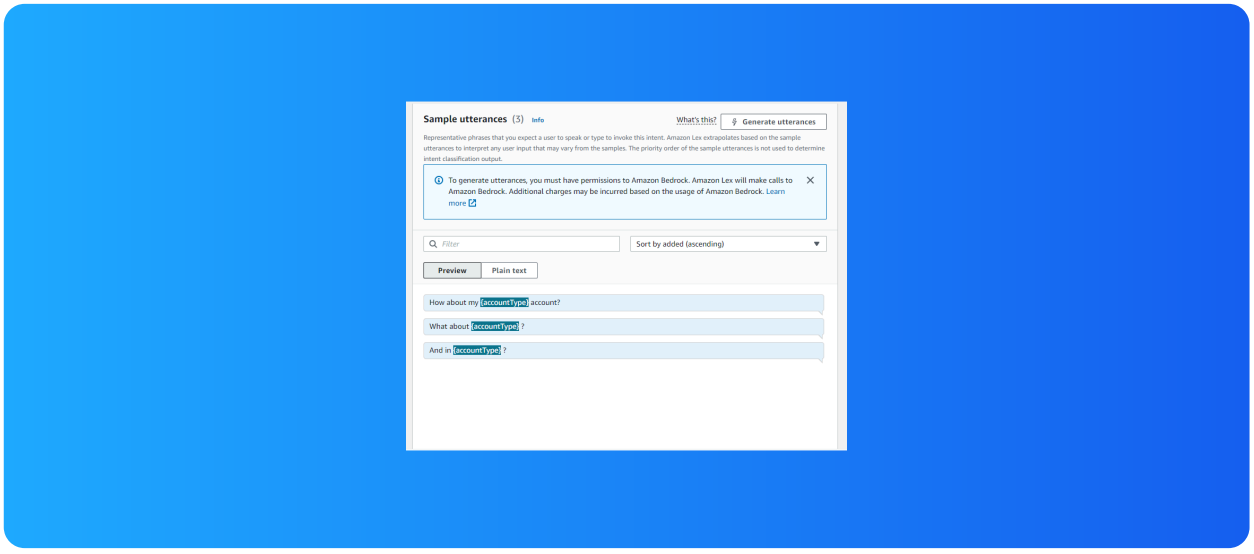


Sai Prasanth Reddy
NextWork Student

FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to allow users to make a follow-up balance check request without needing to re-authenticate, streamlining the interaction after the initial request.

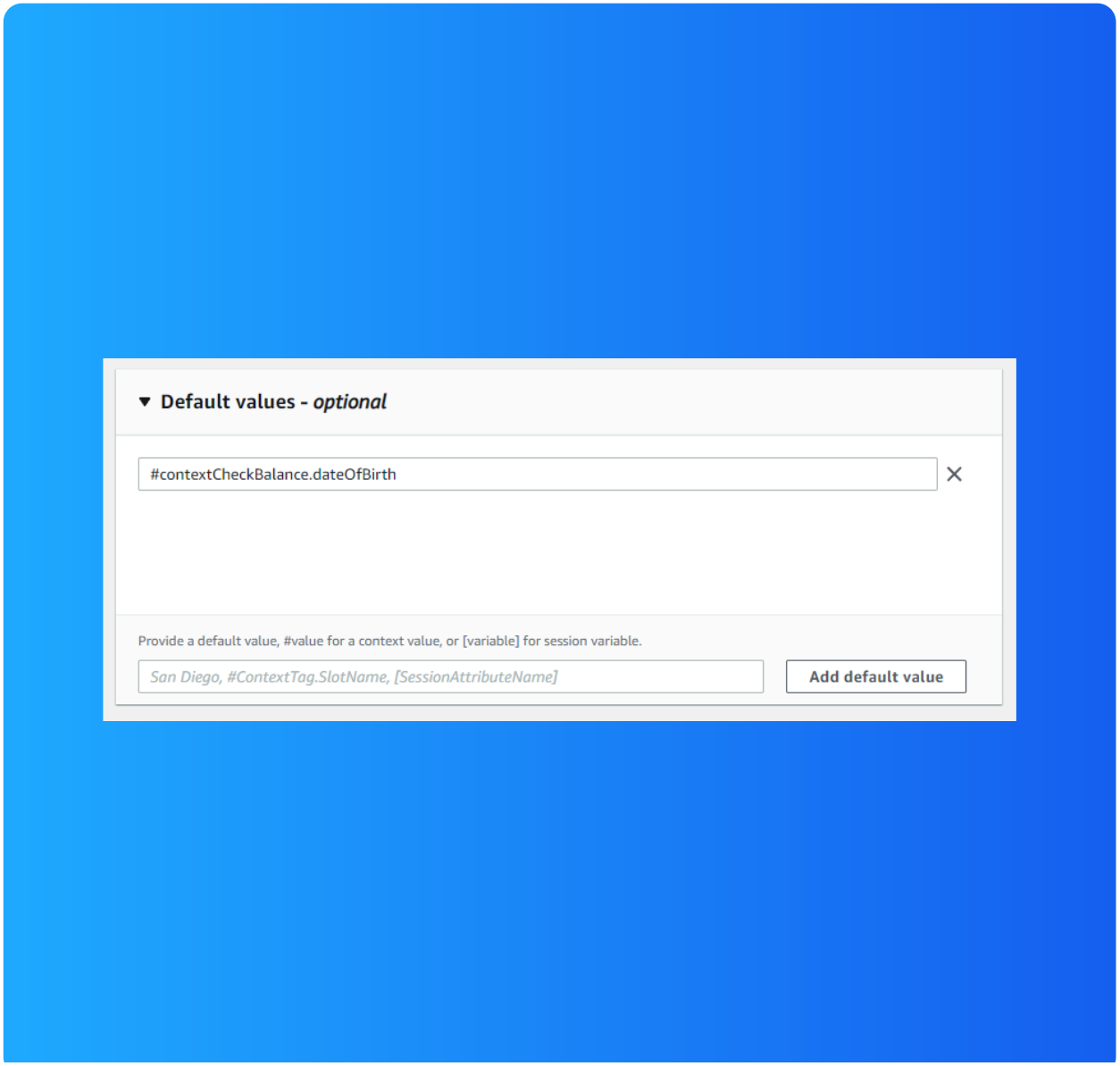
This intent is connected to the previous intent, CheckBalance, because it uses the **contextCheckBalance** tag, allowing users to make a follow-up balance check without re-authenticating after the initial balance inquiry.



Sai Prasanth Reddy
NextWork Student

Input Context Tag

I created an input context, contextCheckBalance, that activates when the CheckBalance intent is fulfilled. It connects to the output context, allowing the FollowupCheckBalance intent to be triggered without requiring re-auth for follow-up requests.

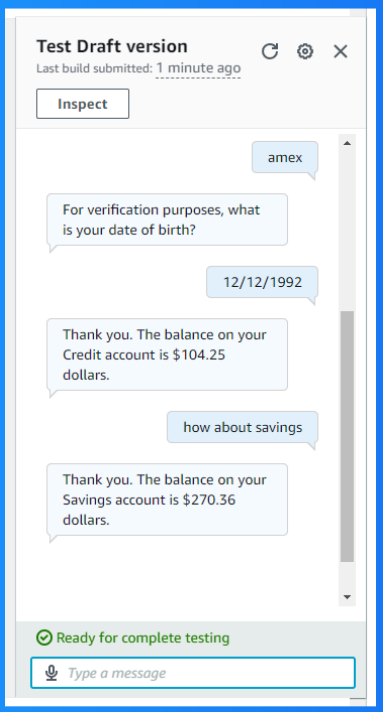


Sai Prasanth Reddy
NextWork Student

The final result!

To see the context tags and the follow-up intent in action, I said "How about savings?" to my chatbot, which triggered the FollowupCheckBalance intent using the contextCheckBalance tag without needing re-authentication.

If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, the intent wouldn't have activated, as it relies on the contextCheckBalance tag from the CheckBalance intent to allow follow-up requests.



Everyone should be in a job they love.

Check out nextwork.org for more projects