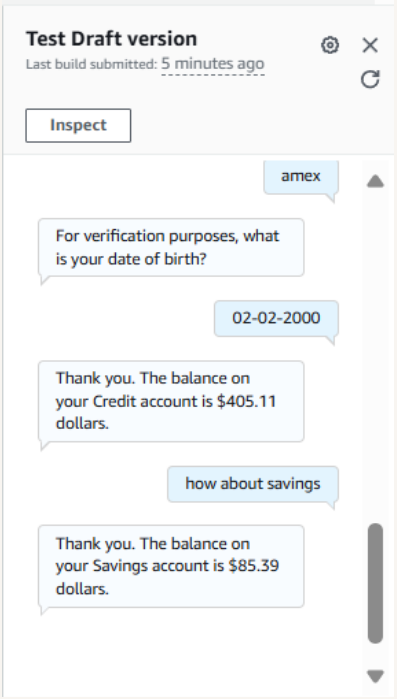
NextWork.org

Save User Info with your Chatbot

Dineshraj Dhanapathy



DI
Dineshraj Dhanapathy
NextWork Student
NextWork.org

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building chatbots using AI and NLP. It enables voice and text interactions, automates responses, integrates with AWS, and helps create smart, scalable conversational interfaces.

How I used Amazon Lex in this project

I used Amazon Lex to create a chatbot that handles balance inquiries. It stored user context with output tags in CheckBalance and reused it in FollowupCheckBalance, reducing repeated questions.

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

I didn't expect FollowupCheckBalance to fail when context wasn't set properly. Testing showed that without CheckBalance running first, the chatbot asked for missing details instead of retrieving them.

This project took me...

This project took me a few hours, mainly for setting up Amazon Lex intents, context tags, and testing. Debugging FollowupCheckBalance to ensure it correctly used stored context took extra time.

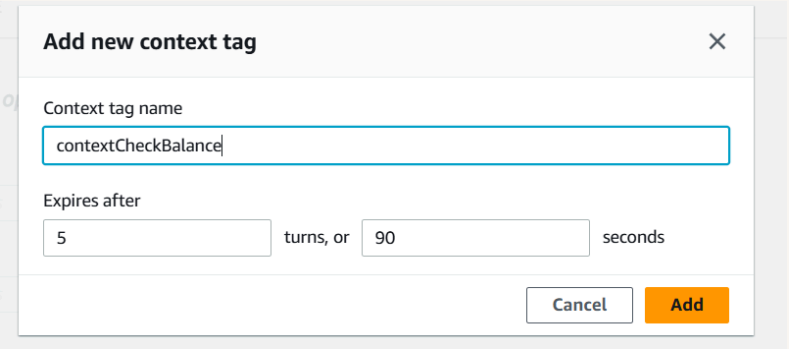
DI
Dineshraj Dhanapathy
NextWork Student
NextWork.org

Context Tags

Context tags are metadata labels used to categorize, filter, or track information within a system. They help in organizing data, improving searchability, and providing relevant context for better analysis and automation.

There are two types of context tags: output and input. Output tags store details for later use, while input tags check if needed details exist before triggering an intent, avoiding redundant questions.

I created a context tag called BalanceContext. This context tag was created in the intent BalanceCheck. This tag stores information about the account type, allowing reuse in follow-up interactions.

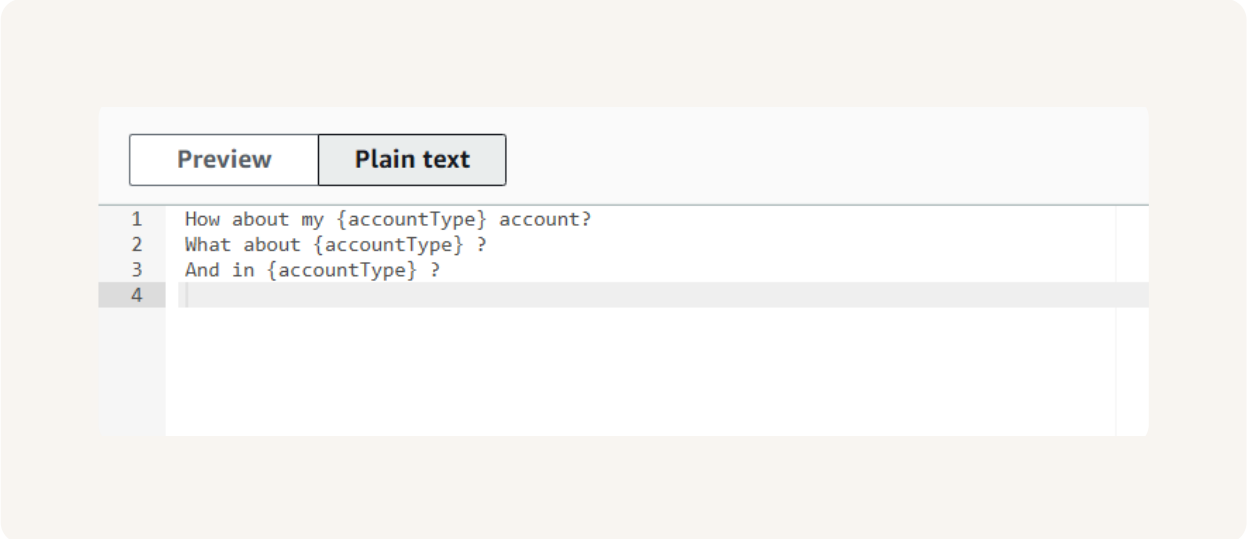


DI
Dineshraj Dhanapathy
NextWork Student

FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to retrieve the user's account balance using stored details, avoiding redundant questions like account type or DOB.

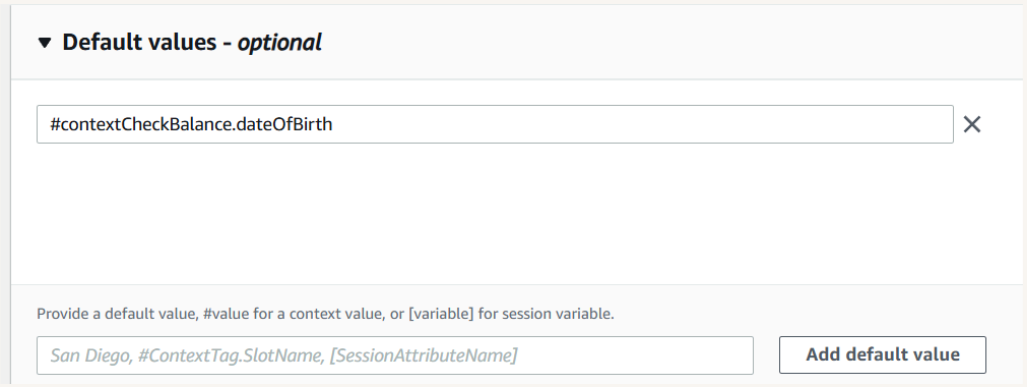
This intent is connected to the previous intent I made, CheckBalance, because FollowupCheckBalance uses stored details from CheckBalance, avoiding repeated questions and ensuring a smooth conversation flow.



DI
Dineshraj Dhanapathy
NextWork Student

Input Context Tag

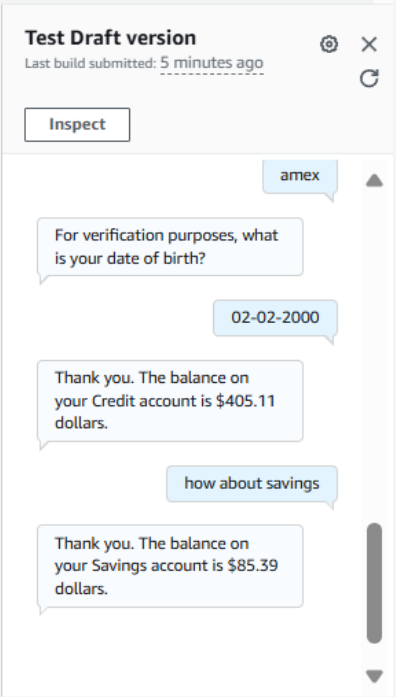
I created an input context, contextCheckBalance, that uses stored details from the output context of CheckBalance, ensuring the system remembers account info and avoids asking repeated questions.



The final result!

To see the context tags and follow-up intent in action, I asked my chatbot, "Can you check my balance again?" Since CheckBalance stored my account type, FollowupCheckBalance retrieved it without asking again.

If I had gone straight to trying to trigger FollowupCheckBalance without setting up any context, the chatbot wouldn't have the needed details and would ask again for account type or other missing info.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects