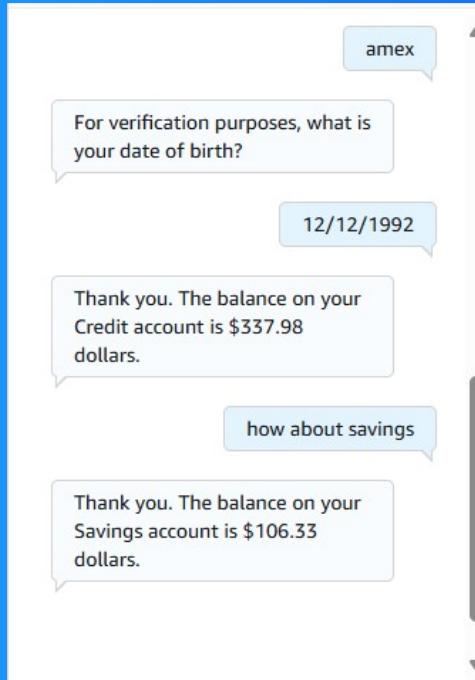




Save User Info with your Chatbot



Adithya Bellamkonda





Adithya Bellamkonda
NextWork Student

NextWork.org

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces using voice and text. It leverages the same deep learning technologies that power Amazon Alexa, providing automatic speech recognition (ASR) and natural language understanding (NLU).

How I used Amazon Lex in this project

I used Amazon Lex to create a banker bot in today's project, setting it up in 10 minutes. I configured intents, created custom slots for user inputs, integrated AWS Lambda for backend processing, and implemented context tags to maintain conversation

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

One thing I didn't expect in this project was the ease and speed of integrating AWS Lambda with Amazon Lex, allowing me to enhance the bot's functionality without complex configurations.

This project took me...

Overall just about 30 minutes to set up the new intent configurations, configure code hooks on fulfillment panels, create new context tags, test custom slots, and through trial and error testing.



Adithya Bellamkonda
NextWork Student

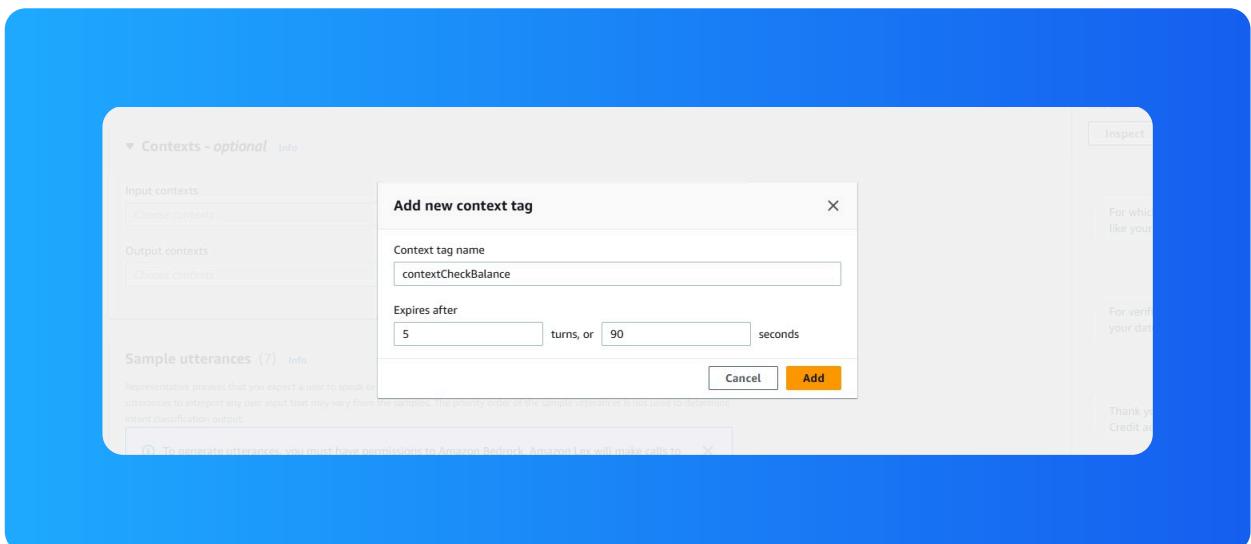
NextWork.org

Context Tags

Context tags are labels used in Amazon Lex to manage the flow of conversations by maintaining the context of the user's interaction. They help the chatbot remember information across multiple turns in a conversation.

In Amazon Lex, output context tags remember details after an intent finishes for later use, like saving the account type. Input context tags check for existing details before activating an intent, like the user's date of birth.

I created a context tag called contextCheckBalance in the BalanceCheck intent. This tag stores information about the user's account type for use in subsequent interactions.





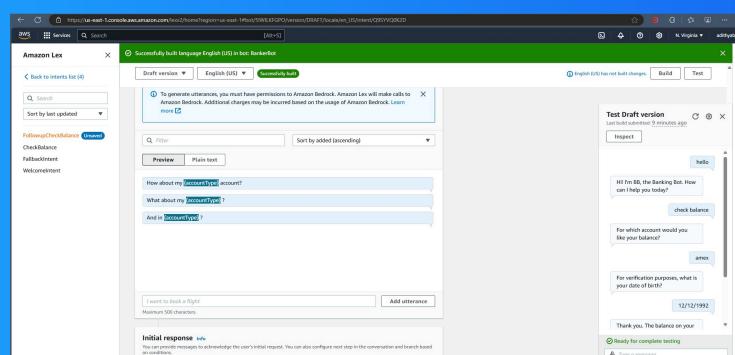
Adithya Bellamkonda
NextWork Student

NextWork.org

FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. This intent handles follow-up queries related to checking account balances, using previously stored context information to streamline the interaction.

This intent is connected to the previous intent I made, CheckBalance, because it leverages the context tag created in that intent to retrieve and use the user's account type, enhancing the continuity of the conversation.





Adithya Bellamkonda
NextWork Student

NextWork.org

Input Context Tag

I created an input context, contextCheckBalance, that ensures the FollowupCheckBalance intent can only be activated when the user's account type information is available, preventing unnecessary questions.

The screenshot shows the configuration for a slot named 'dateOfBirth' in the Amazon Lex console. The slot is set to the 'AMAZON.Date' slot type. It has a required prompt: 'For verification purposes, what is Your date ...'. A checkbox 'Required for this intent' is checked. In the 'Default values - optional' section, the value '#contextCheckBalance.dateOfBirth.' is specified. A modal window titled 'Slot: dateOfBirth' provides more details about the slot, including options for handling slot value errors and setting values.

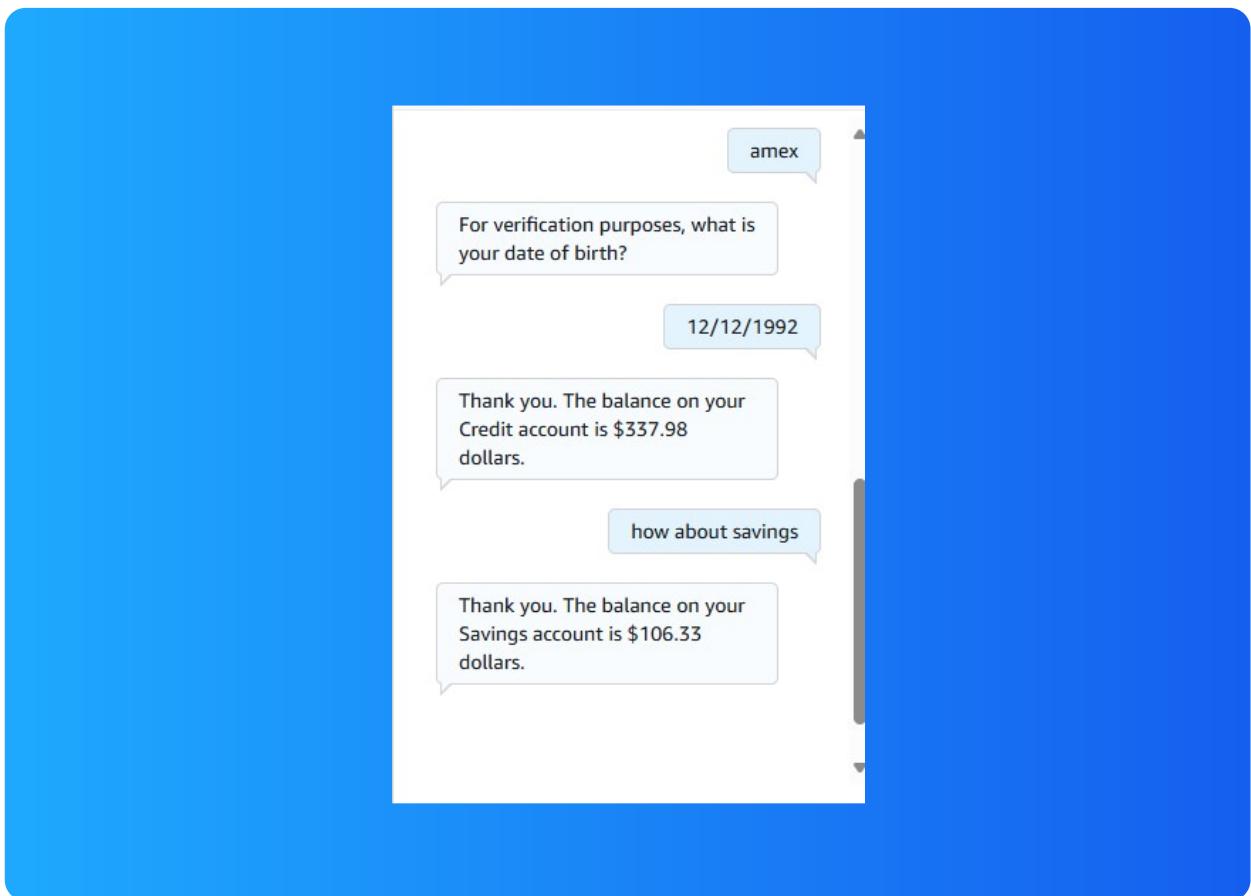
Adithya Bellamkonda
NextWork Student

NextWork.org

The final result!

To see the context tags and the follow-up in intent in action, I tested the conversation flow by initiating a balance check and then proceeding to ask follow-up questions, verifying that the context was maintained and used correctly.

If I had gone straight to trying to trigger FollowupCheckBalance without setting up any context, the intent would not have been activated because it relies on the availability of the user's account type information.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

