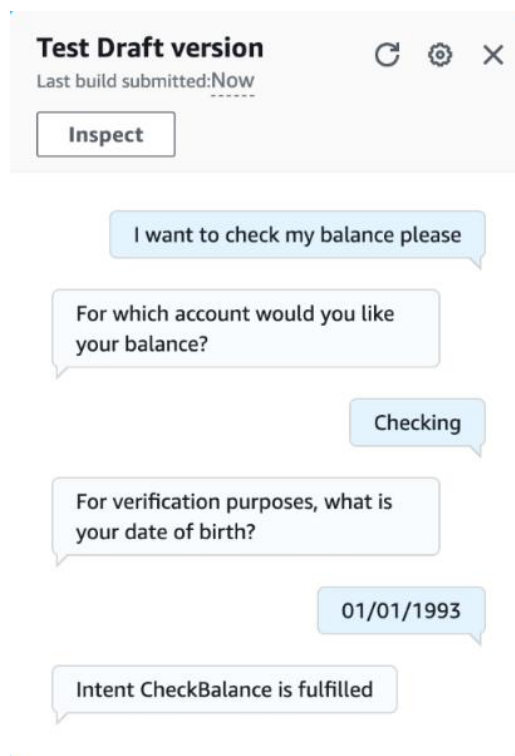


Build a Chatbot with Custom Slots



Introducing Today's Project!

In today's project, I used Amazon Lex to build a banking chatbot. It handles queries like checking balances and providing account information. I defined intents, configured slots for account details, and set up a fallback for unrecognized inputs.

What is Amazon Lex?

Amazon Lex is an AWS service for building conversational interfaces with natural language processing and speech recognition. It allows you to create text or voice-based chatbots, enhancing automation, customer service, and scalability.

One thing I didn't expect in this project was

One thing I didn't expect in this project was the challenge of training AWS Lex to handle a wide variety of user inputs. Variations in phrasing, slang, and incomplete information required constant tweaking of intents and responses to ensure accuracy.

This project took me

It took me almost an hour to complete this project.

Slots

Slots are pieces of information that a chatbot needs to complete a user's request. Think of them as blanks that need to be filled in a form. For example, to book tables in restaurant, a chatbot needs details like: restaurant name, date, time, people.

By adding custom slots in utterances, my chatbot can capture specific details like dates, names, or numbers. This enables the bot to process complex requests accurately, enhancing the user experience with personalized, efficient interactions. In this project, I created a custom slot type to capture specific information from users, like restaurant name, date, time, and people count for tasks such as booking a table, ensuring accurate and structured data collection.

Successfully built language English (IN) in bot: BankerBot

Draft version English (IN) Successfully built English (IN) has not built changes. Build Test

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

Slot type	Value	Action
Checking	Tab or ; or enter return for new value	X
Savings	Tab or ; or enter return for new value	X
Credit	Tab or ; or enter return for new value	X

credit card X visa X mastercard X
amex X american express X

Value Tab or ; or enter return for new value Add value

Save Slot type

Connecting slots with intents

This slot type has restricted slot values, which means only the values that you specify will count as a valid accountType! Otherwise, Amazon Lex might use machine learning to accept other values that it frequently encounters from users. I associated my custom slot with CheckBalance, which is typically designed to handle user requests related to checking their account balance. When a user interacts with the bot and asks for their balance the CheckBalance intent is triggered.

Successfully built language English (IN) in bot: BankerBot

Draft version English (IN) Successfully built English (IN) has not built changes. Build Test

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

Slot type	Value	Action
Checking	Tab or ; or enter return for new value	X
Savings	Tab or ; or enter return for new value	X
Credit	Tab or ; or enter return for new value	X

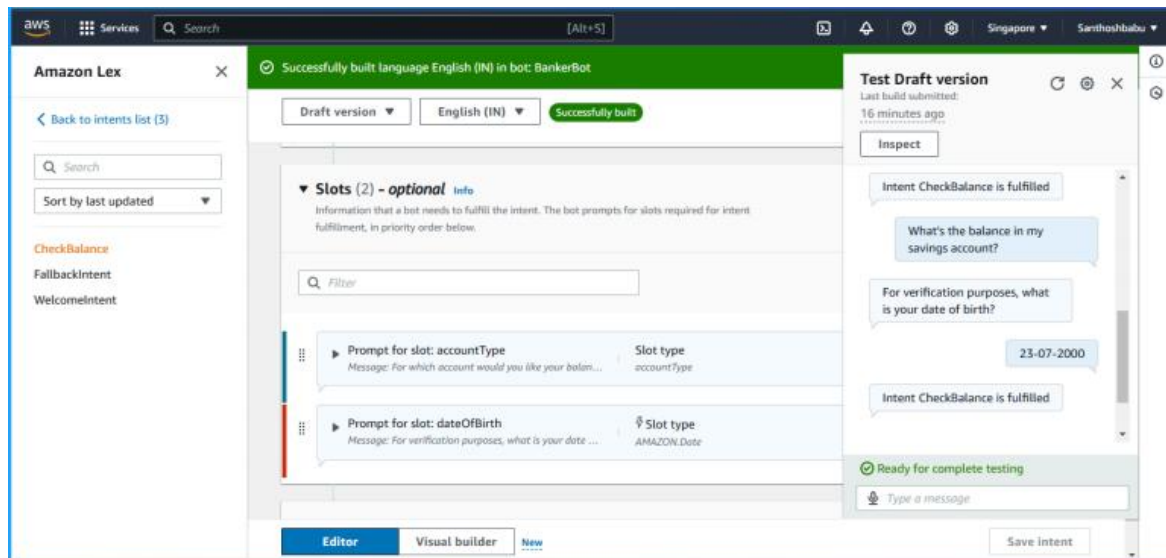
credit card X visa X mastercard X
amex X american express X

Value Tab or ; or enter return for new value Add value

Save Slot type

Slot values in utterances

I included slot values in some of the utterances (i.e. user inputs) by navigating to Slots -> Add slot -> "add the details" -> Add. For example I created "accountType" and "dateOfBirth" slots.



To wrap things up, today you've learnt how to:

1. **Set up a custom slot type:** You created a custom slot type in Amazon Lex to help your chatbot understand different bank account types.
2. **Link slots to your intent:** You connected both custom and standard slots to your intent.
3. **Check for slot values in conversation:** You've learnt how to use Input to check for the slot values your chatbot collects on the user during a conversation.