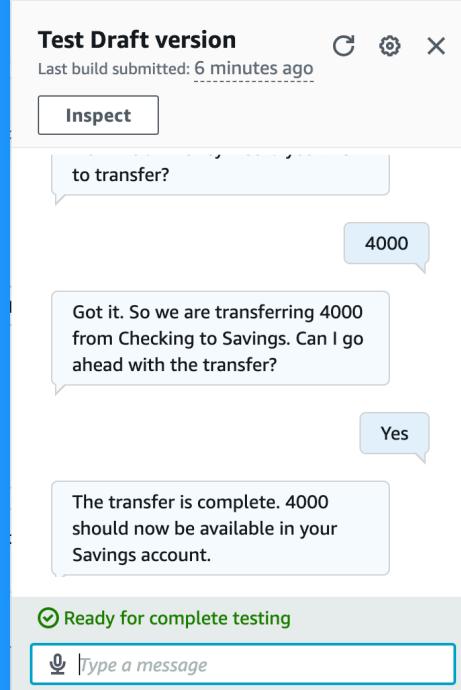




Build a Chatbot with Multiple Slots



imly04@proton.me



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a tool that allows anyone to create a chatbot that can be used by customers/users 24/7, it can act as a virtual assistant and automate tasks for users, with high customisation and ease of use.

How I used Amazon Lex in this project

I used Amazon Lex to add a chatbot through CloudFormation which is an AWS service that allows you to create and deploy AWS resources such as a chatbot by providing code.

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

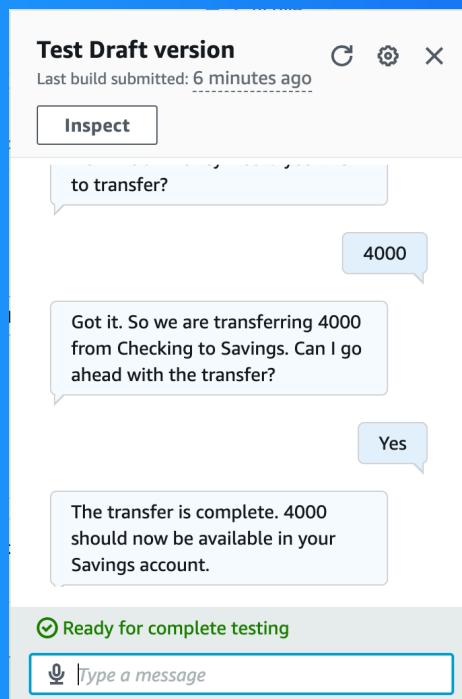
How quick the setup process of a chatbot by using CloudFormation

This project took me...

45 minutes

TransferFunds

An intent I created for my chatbot was TransferFunds, which allows the user to transfer funds between account by using the BankerBot.



Using multiple slots

For this intent, I had to use the same slot type twice. This is because they require to use the same information such as accountType stores the information on all types of valid bank accounts, therefore, there is no need to create a new one again.

I also learnt how to create confirmation prompts, which are prompts which are repeated back to the user and the user can confirm if that is what they wanted to do.

Confirmation Info

Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

Prompts to confirm the intent

Message: Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I go ahead?

Confirmation prompt

What will the bot say to prompt the user to confirm this intent.

Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I go ahead

Decline response

What will the bot say if the user says NO to the confirmation prompt.

The transfer has been cancelled.

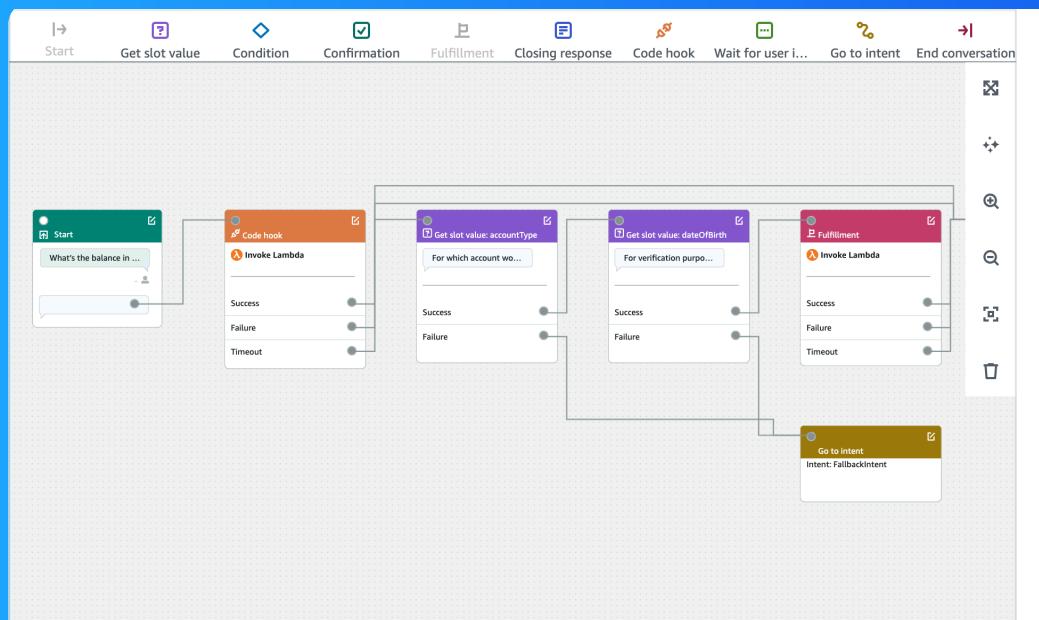
Advanced options

Configure confirmation prompts and decline responses.

Exploring Lex features

Lex also has a special conversation flow feature that allows you to view every step of the chatbot's conversation and it has possible recommendation which you can add simply by clicking on add, you can also edit any conversation sections as well

You could also set up your intent using a visual builder! A visual builder is way to build a create chatbot intents by visual coding similar to scratch rather than coding and physically setuping every intent.



AWS CloudFormation

AWS CloudFormation is service that allows you create and deploy AWS services, it is known as a infrastructure as code service it is complete online.

I used CloudFormation to deploy a new complete bankerbot.

The screenshot shows a table of intents with the following data:

Name	Description	Last edited
TransferFunds	Help user transfer funds between bank accounts	6 minutes ago
FollowupCheckBalance	Intent to allow a follow-up balance check request without authentication	6 minutes ago
CheckBalance	Intent to check the balance in the specified account type	6 minutes ago
Welcome	Welcome intent	6 minutes ago
FallbackIntent	Default fallback intent when no other intent matches	6 minutes ago

The final result!

Re-building my bot with CloudFormation took me less than 2 minutes.

There was an error after I deployed my bot! The error was with the Lambda function not being invoked, I fixed this by creating a resource-based policy statement which solved the error.

Edit policy statement

AWS account
Grant permissions to another AWS account, user, or role.

AWS service
Grant permissions to another AWS service.

Function URL
Grant permissions to invoke your function through the function URL.

Service
The AWS service to grant permissions to.

Statement ID
Enter a unique statement ID to differentiate this statement within the policy.

Principal
The service principal for this AWS service. [Learn more](#)

Source ARN
The ARN for a resource. Find the ARN in the related service console.

Action
Choose an action to allow.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

