#### Amazon Lex Chatbot

**banking chatbot**

**Amazon Lex Chatbot**

30 second Summary

*You can use Amazon Lex to build intelligent chatbots that understand natural language and help users get things done—like checking their bank balance or transferring money!*

*In today’s project, I will create a chatbot called BankerBot, designed for an imaginary banking service. This bot will greet users and be ready to handle conversations through voice or text*

**Things That I do in this project**

[***Build Your First Chatbot with Amazon Lex***](https://link.nextwork.org/projects/aws-ai-lex1?utm_source=project-app)

[***Add Custom Slots to Your Lex Chatbot***](https://link.nextwork.org/projects/aws-ai-lex2?utm_source=project-app)

[***Connect Your Lex Chatbot to Lambda***](https://link.nextwork.org/projects/aws-ai-lex3?utm_source=project-app)

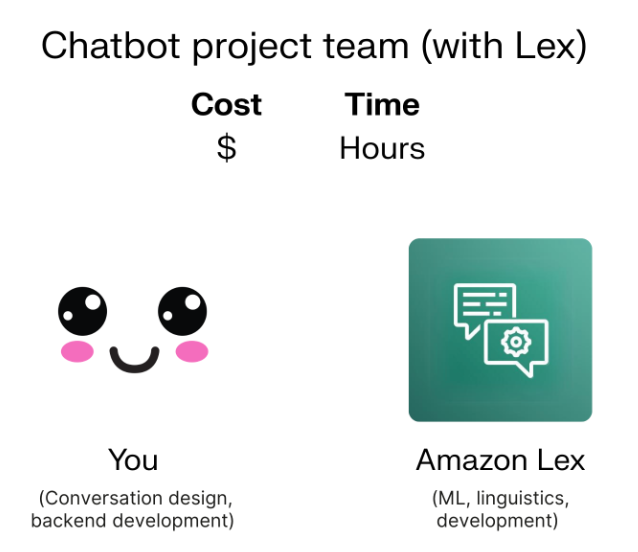
[***Save User Info with Your Lex Chatbot***](https://link.nextwork.org/projects/aws-ai-lex4?utm_source=project-app)

[***Build Complex Conversations with Multiple Slots***](https://link.nextwork.org/projects/aws-ai-lex5?utm_source=project-app)

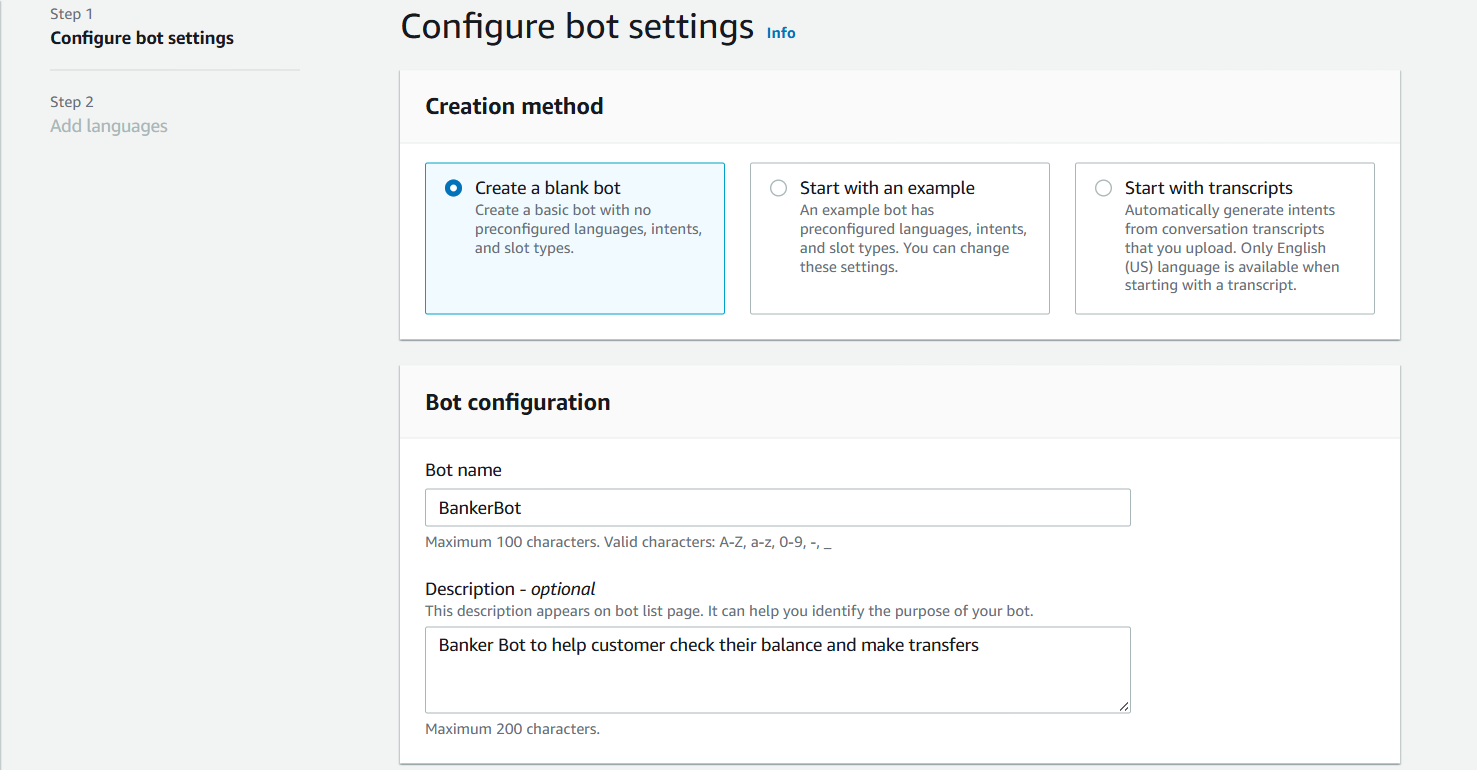
*Amazon Lex is AWS's service for building chatbots - it eliminates most of the complexity and turns chatbot building from taking months to just hours.*

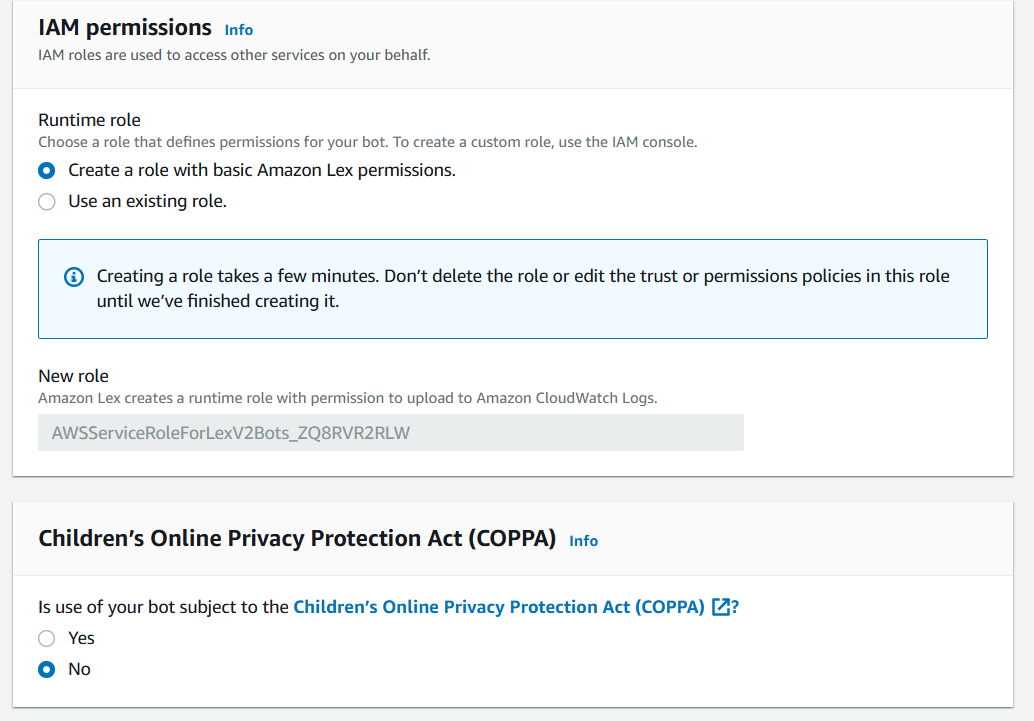
***Tasks handled by Amazon Lex:***

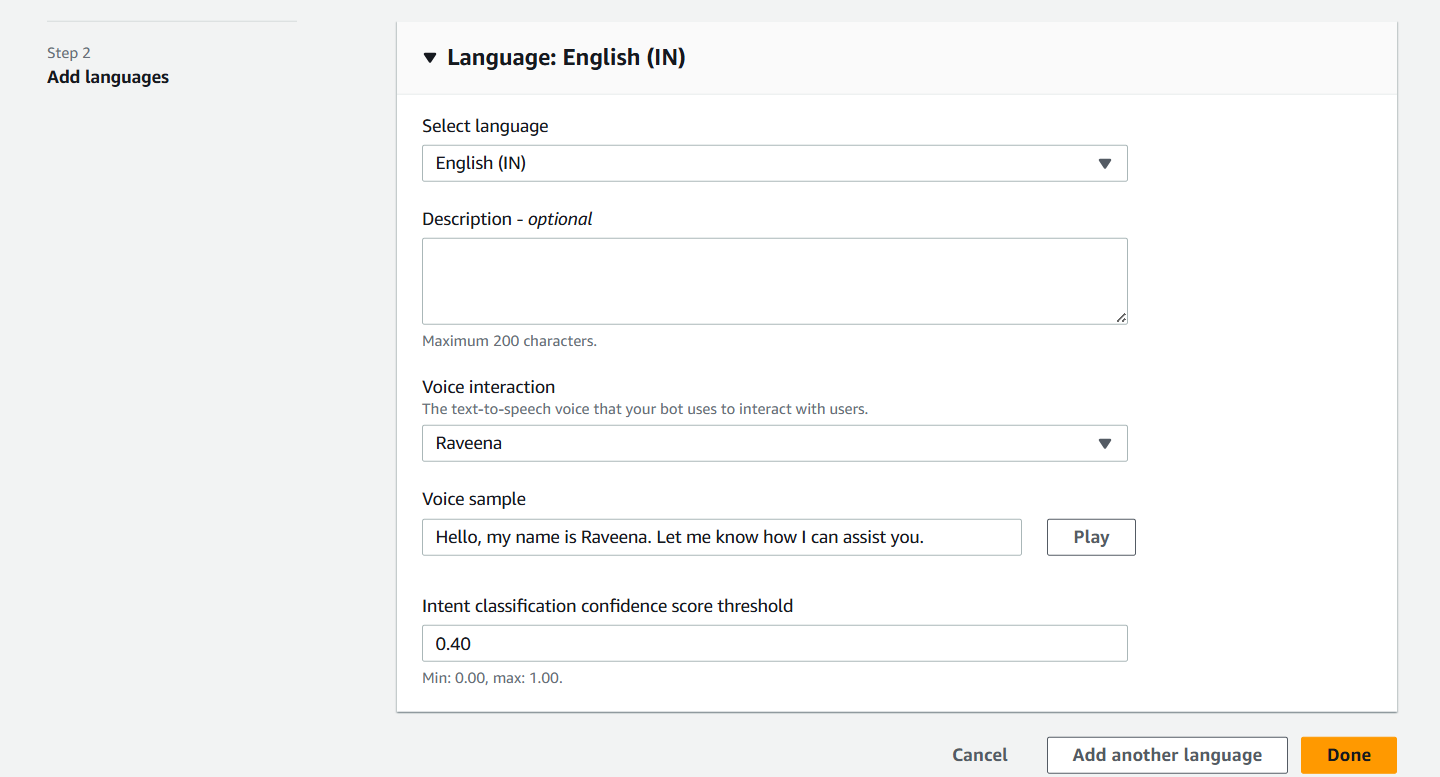
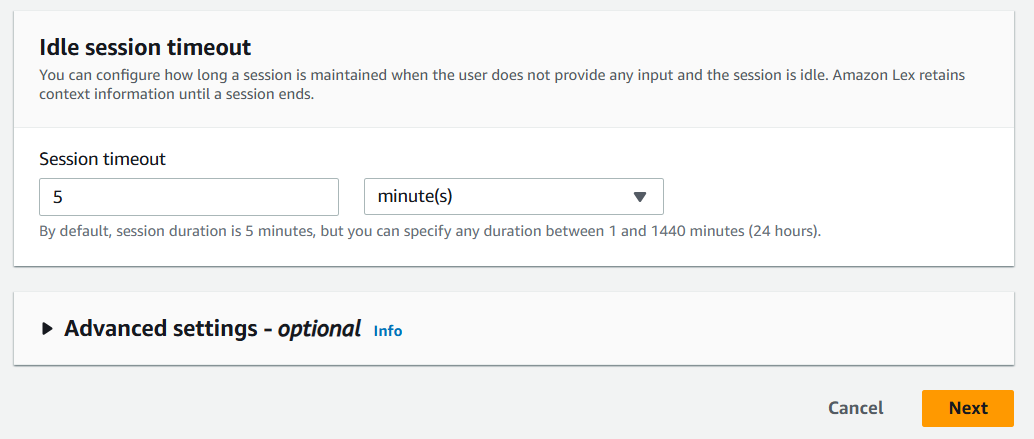
* *Providing pre-trained models that understand user text/speech*
* *Generating thousands of training examples internally*
* *Extract important details from what users say (like dates, names, amounts)*
* *Build logic that pinpoint exactly what users are trying to do*

**

**Creation of lexbot**

****

****

****

**How long did it take to create a chatbot?**

I created my chatbot from scratch with Amazon Lex. Setting it up took me 7 minutes.

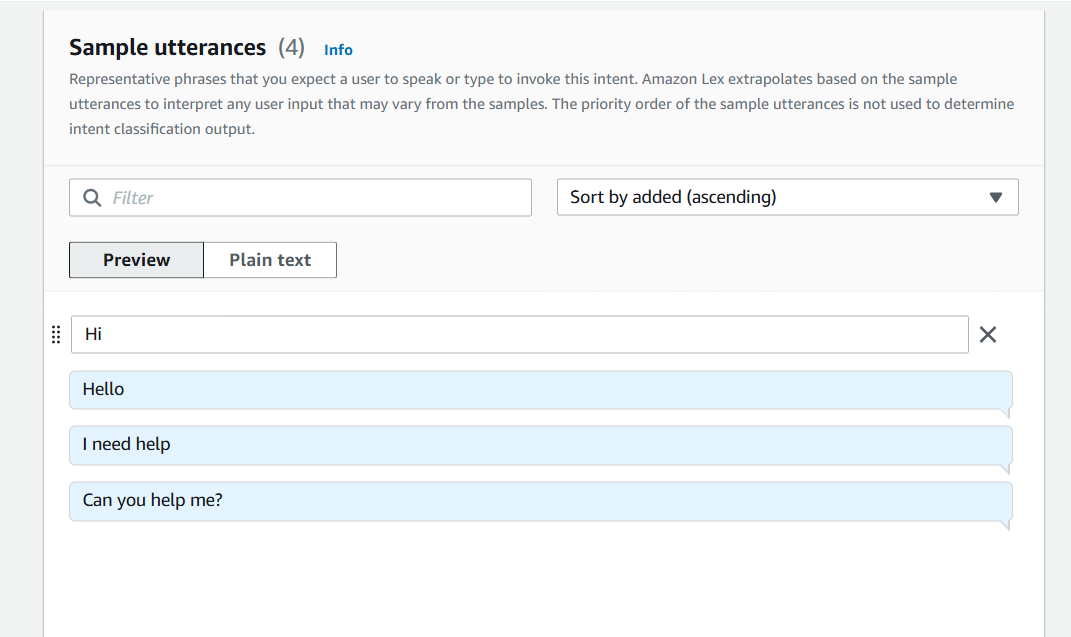
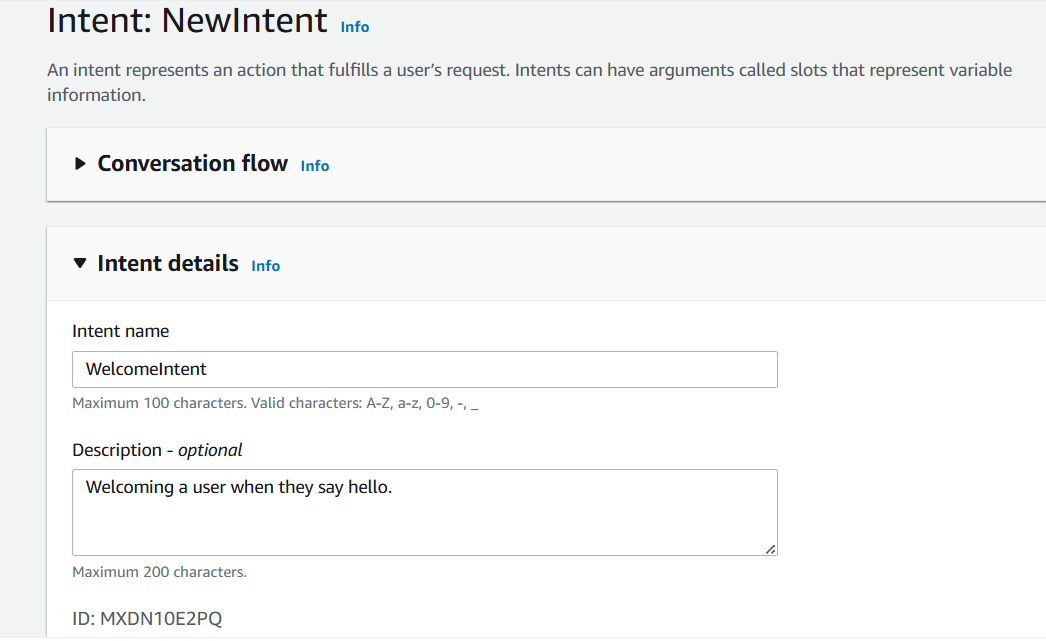
**Why did I give my chatbot basic Amazon Lex permissions?**

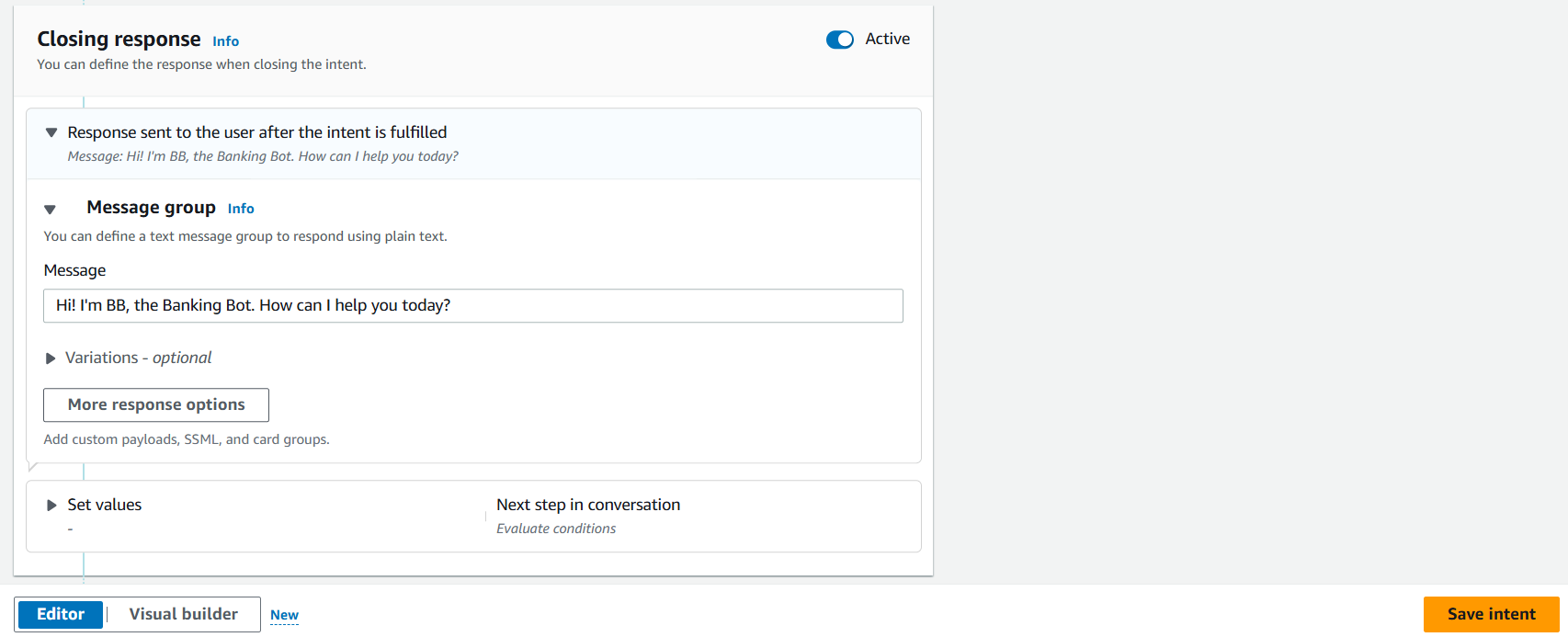
While creating my chatbot, I also created a role with basic permissions because Lex needs permission to interact with other AWS services like Lambda.

**What does an intent classification confidence score mean?**

In terms of the intent classification confidence score, I kept the default value of 0.40. This means my chatbot needs to be at least 40% confident to respond to a user's input.

**Creation Of Intent**

****

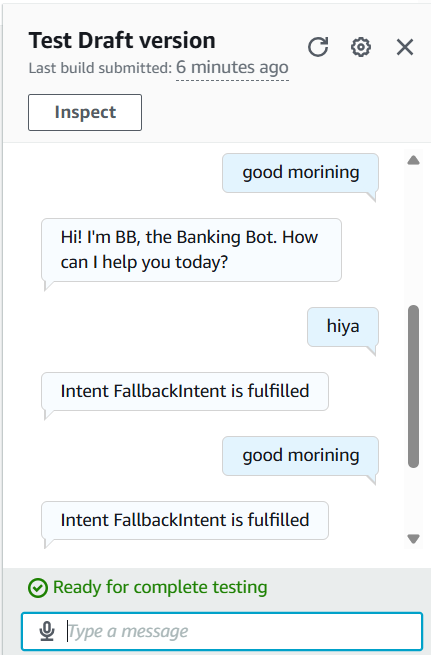
****

**What are intents?**

An intent is what the user is trying to achieve in their conversation with the chatbot. For example, checking a bank account balance; booking a flight; ordering food.

In Amazon Lex, you build your chatbot by defining and categorising different intents. If you set up different intents, one single chatbot can manage a bunch of requests that are usually related to each other.

**Created sample chatbot**

****

**Which of your greetings could your chatbot understand?**

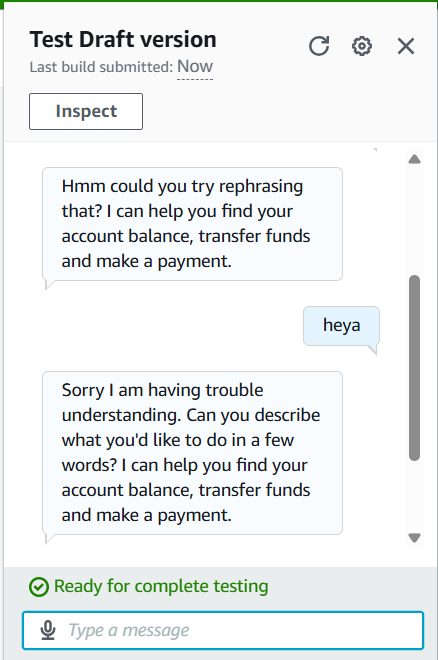
I launched and tested my chatbot, which could respond successfully if I entered "Hiya," and "Hello," and "Hello."

When did your Lex chatbot return an error?

My chatbot returned the error message 'Intent Fallbackintent is fulfilled when I entered "How are you," and "Good Morning" This error message occurred because my bot couldn't match it to the initially defined intents.

**What is FallbackIntent?**  
Remember the intent classification confidence score threshold, and how it's been set to 0.4? If your chatbot has a confidence score **below** 40% for all the intents you've defined (in our case, it's just the WelcomeIntent for now), the FallbackIntent is triggered.

Think of it as a custom error message that your chatbot will use to tell the user it doesn't understand their input.



**How did you configure Fallback intent?**

To configure Fallback intent, I customized the closing response messaged and add variations so it sounds more natural.

**What are variations?**

I also added variations! What this means for an end user is that they will see slightly different responses each time the bot doesn't understand.

**How did you configure Fallback intent?**

To configure Fallback intent, I customized the closing response messaged.

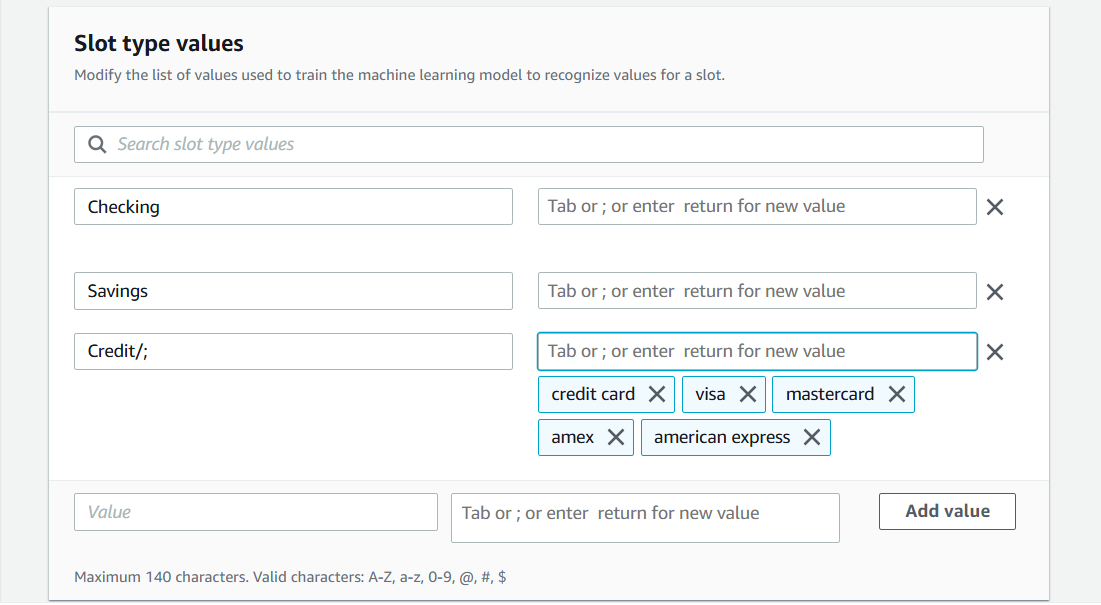
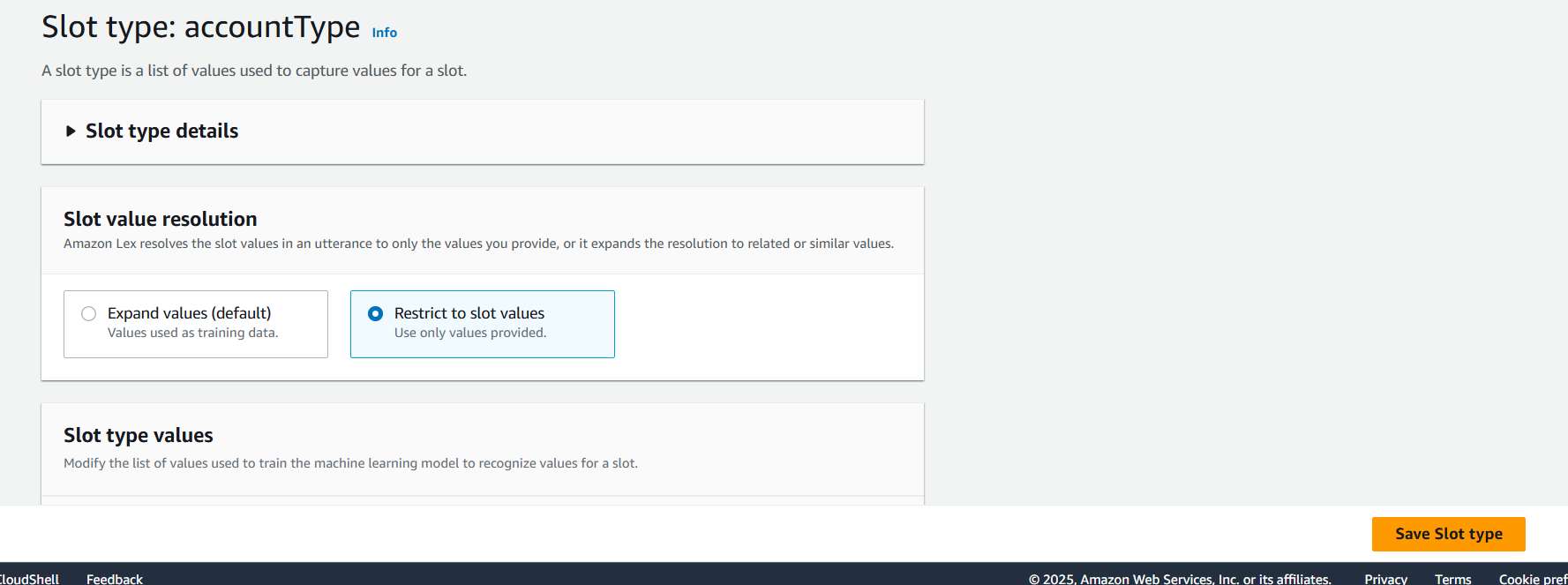
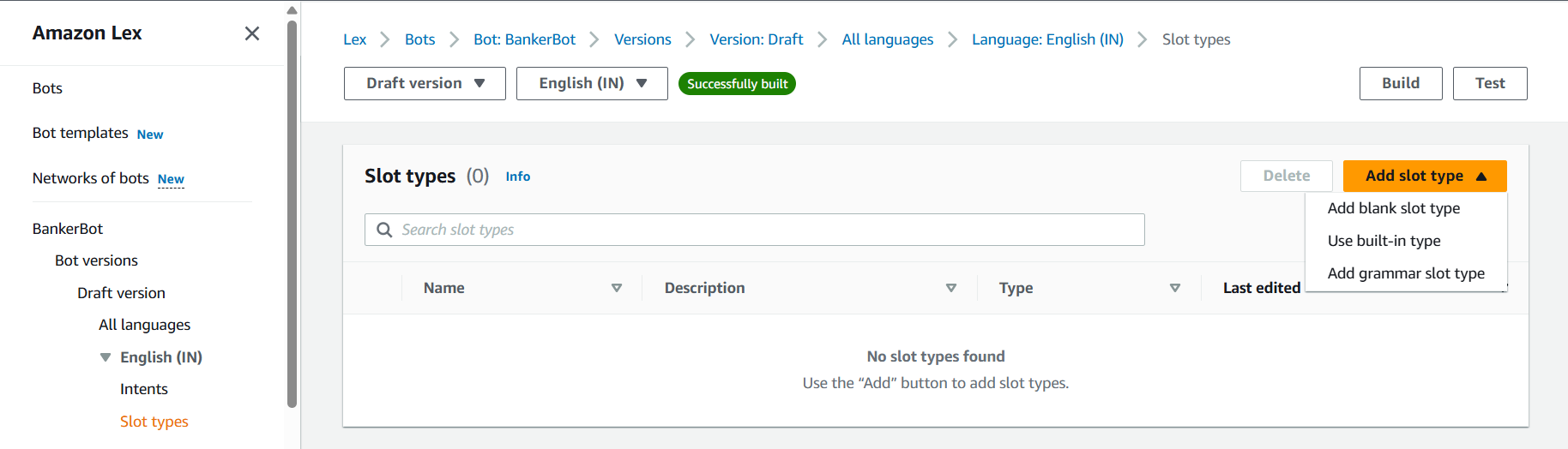
**Add Custom Slots to a Lex Chatbot**

how to create a practical chatbot, BankerBot, that can help your imaginary bank's customers check their account balance and transfer money between accounts

we'll upgrade BankerBot by getting it to check a user's bank balance!

To make that happen, BankerBot will need to ask users for their bank account type and birthday for verification too.

We'll use a cool Lex feature called slots to tell BankerBot what bank account types it should recognize.



**What does the Restrict to slot values setting mean?**

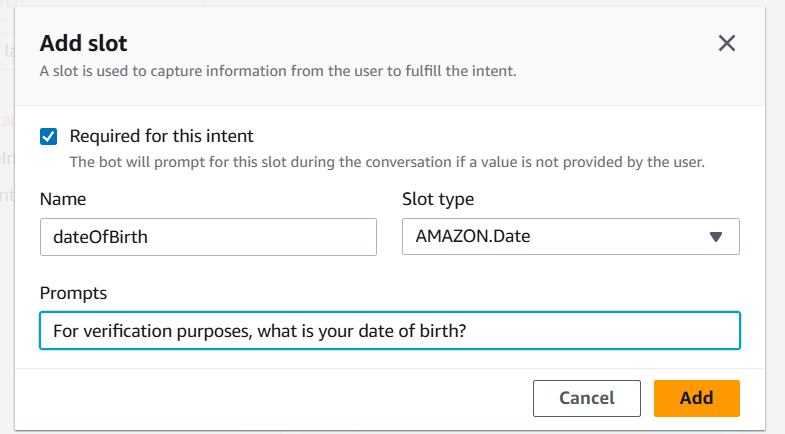
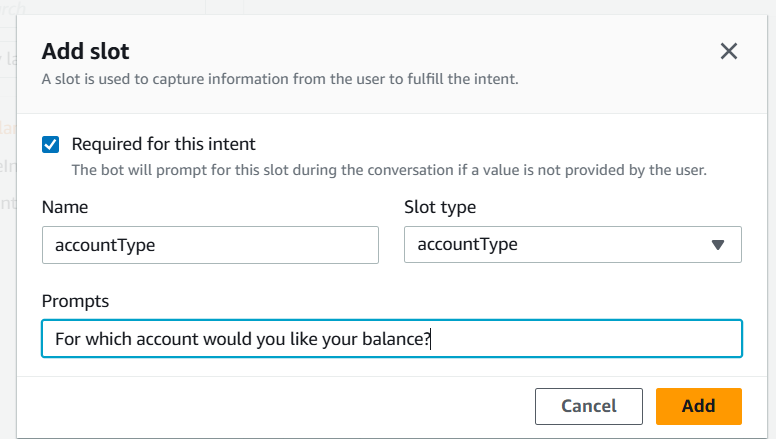
This slot type has restricted slot values, which means Lex will only accept valued i defined and reject anything else

**Create the CheckBalance intent**

Now that BankerBot knows about different bank account types, it's ready to take on users' requests to check their account balance.

In this final step, let's put together two things we know, **intents** and **slots**, to make it happen!

**empty intent and date**

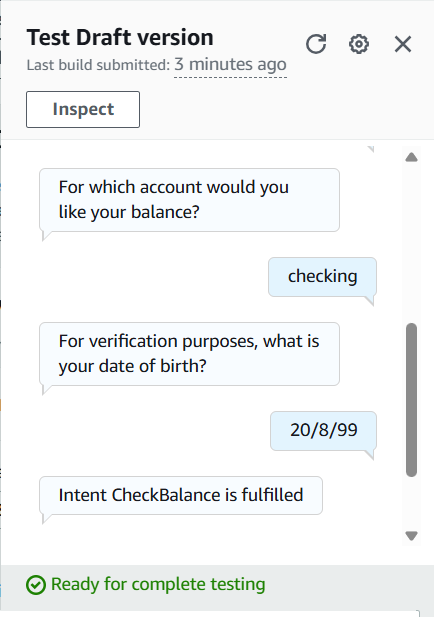
****

**What does the CheckBalance intent do?**

I associated my custom slot with CheckBalance, which is an that collects user data to help them check their bank balance

**How did you include slots values in Check Balance’s utterances?**

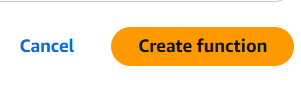
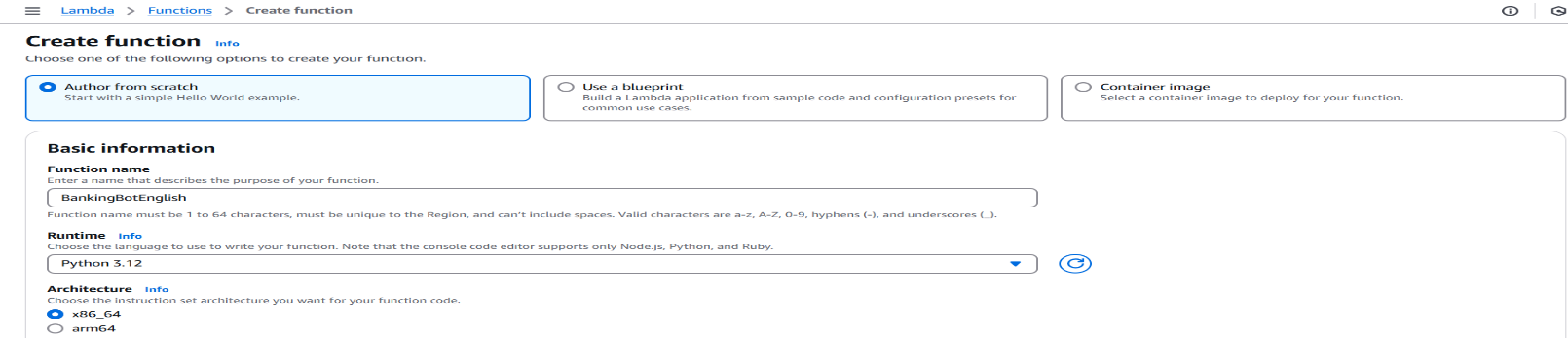
I included slot values in some of the utterances (i.e. user inputs) by using (). For example, What's balance in my (account Type} account?



**What does having custom slots mean for your users?**

By adding custom slots in utterances, my chatbot's users can provide inputs like savings, checking or credit responses.

**Create Your AWS Lambda Function**

****

**Deploy the code in lambda function.py**

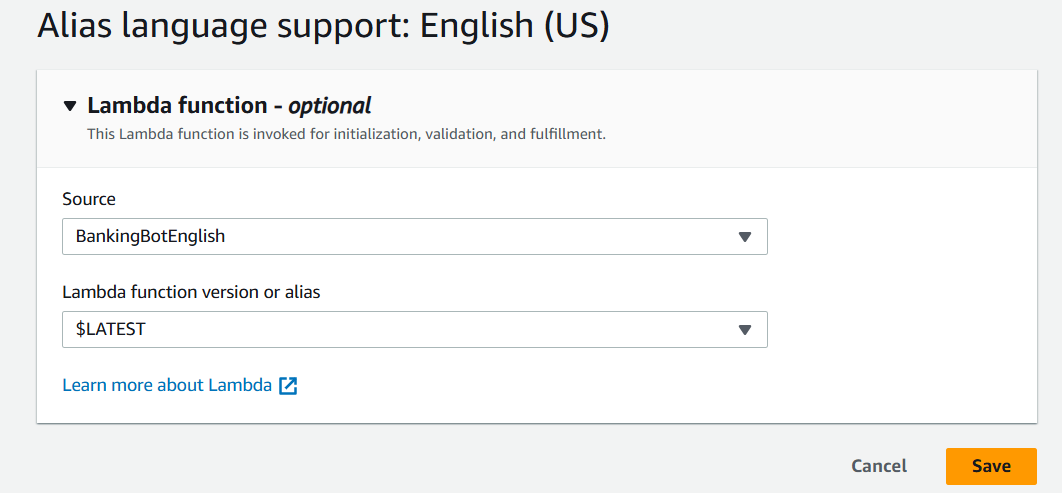
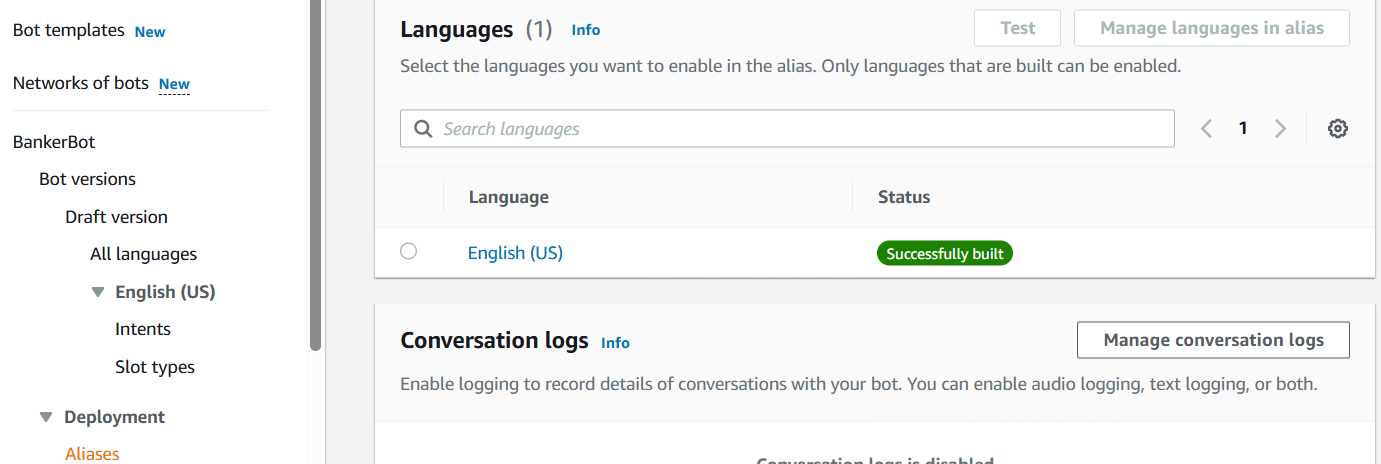
****

**What is lambda**

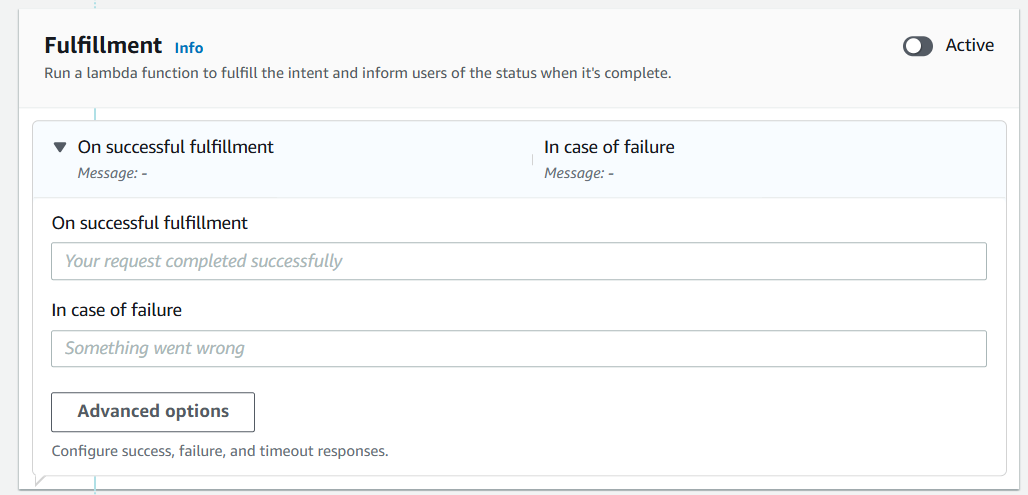
AWS Lambda is an AWS service that help you run code without having to manage servers

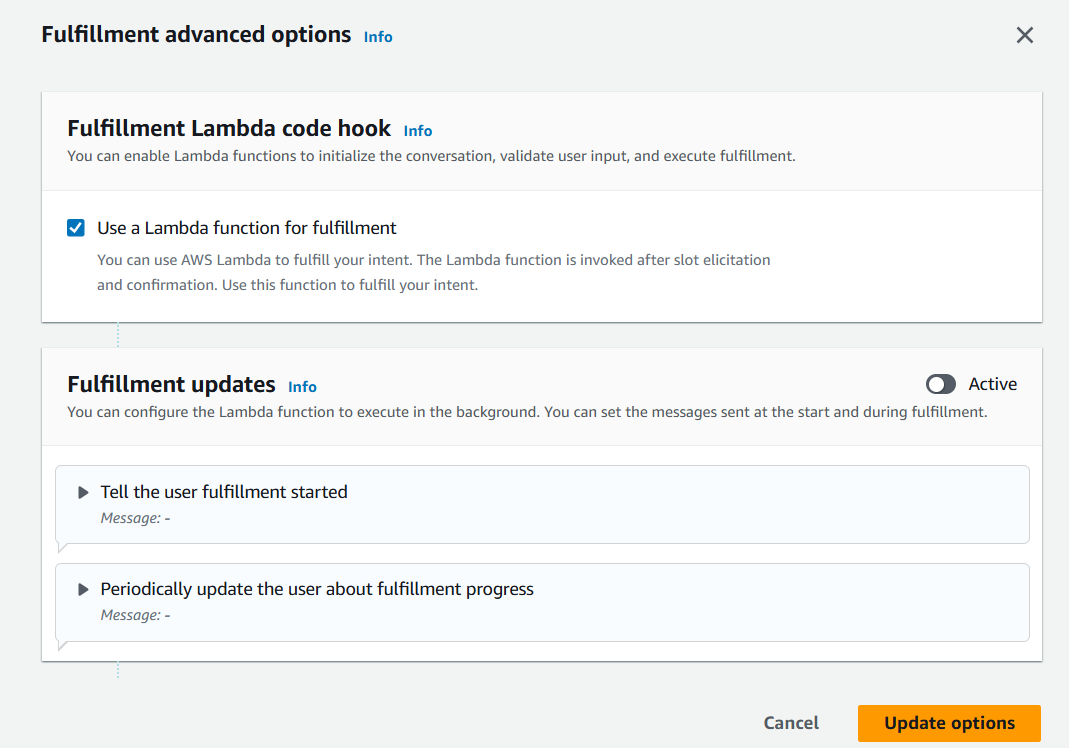
In this project, a Lambda function was created to generate the users' bank balance. In this example, a random figure was generated, however in the real world the Lambda function can be used to extract the user's bank balance from a dtatabase. The Amazon Lex chatbot, on its own, would not be able toi generate a bank balance. That's why this connection to AWS Lambda is crucial!

**Connect AWS Lambda with Amazon Lex**

****

**Connect your CheckBalance intent with your Lambda function**

****

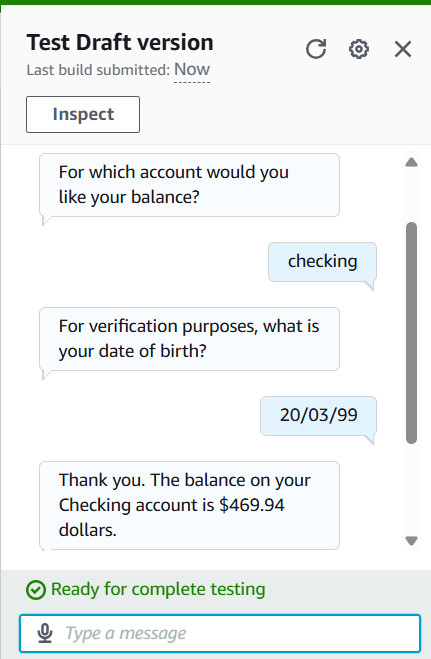
****

**What are code hooks?**Code hooks help you connect your chatbot to custom Lambda functions for doing specific tasks during a conversation.

They're used to handle more complex actions that the basic chatbot setup can't do on its own, like checking data from a database or making decisions based on past conversations.

Essentially, code hooks make your chatbot smarter and more useful by allowing it to perform these extra steps seamlessly during chats

**The final outcome of the banking chatbot**

****

**There were two steps to connecting the Lambda function with my chatbot:**

Step1

To connect Lambda with my chatbot alias, I visited the Alias page of my chatbot and connected my TestBotAlias (my chatbot's default alias, made for development/testing) with the latest version of the AWS Lambda function defined.

Step 2

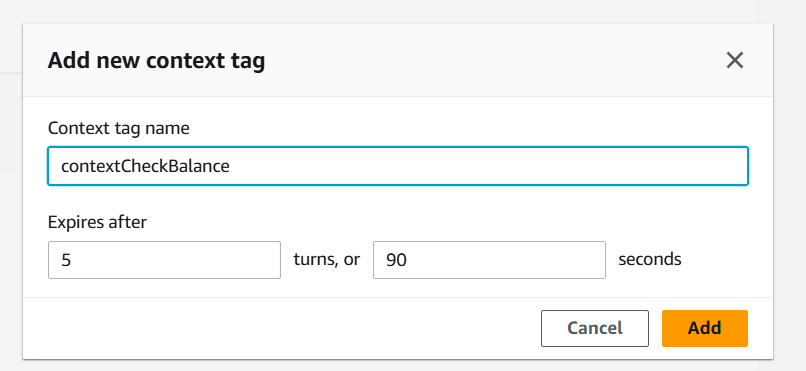
Another intent setting to configure is code hooks.

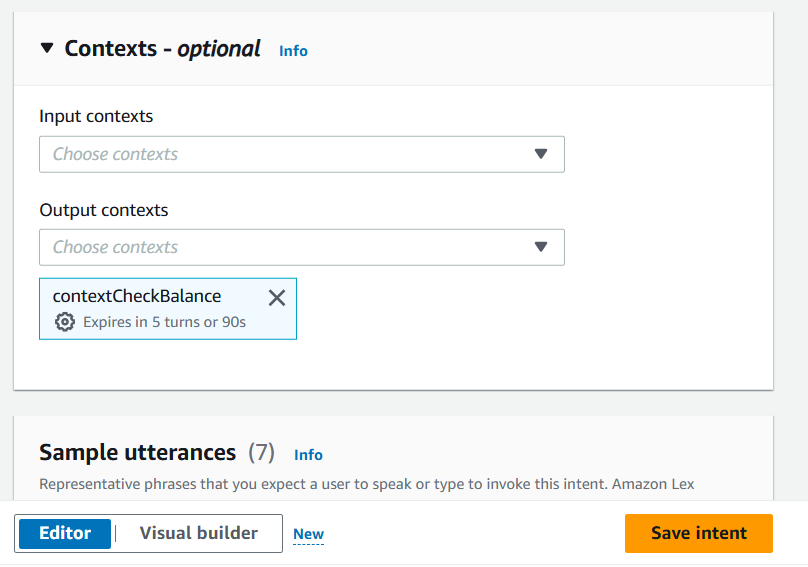
A code hook is a piece of code that can be connected to my chatbot to perform functions/actions that my chatbot cannot do alone/by default.

In this project, I had to use code hooks because the chatbot is not able to calculate/return a bank balance figure on its own.

After connecting Lambda with my Lex bot, my chatbot could immediately start returning specific bank balance figures. The AWS Lambda function would generate a random number each time.

**Save User Info with a Lex Chatbot**

****

****

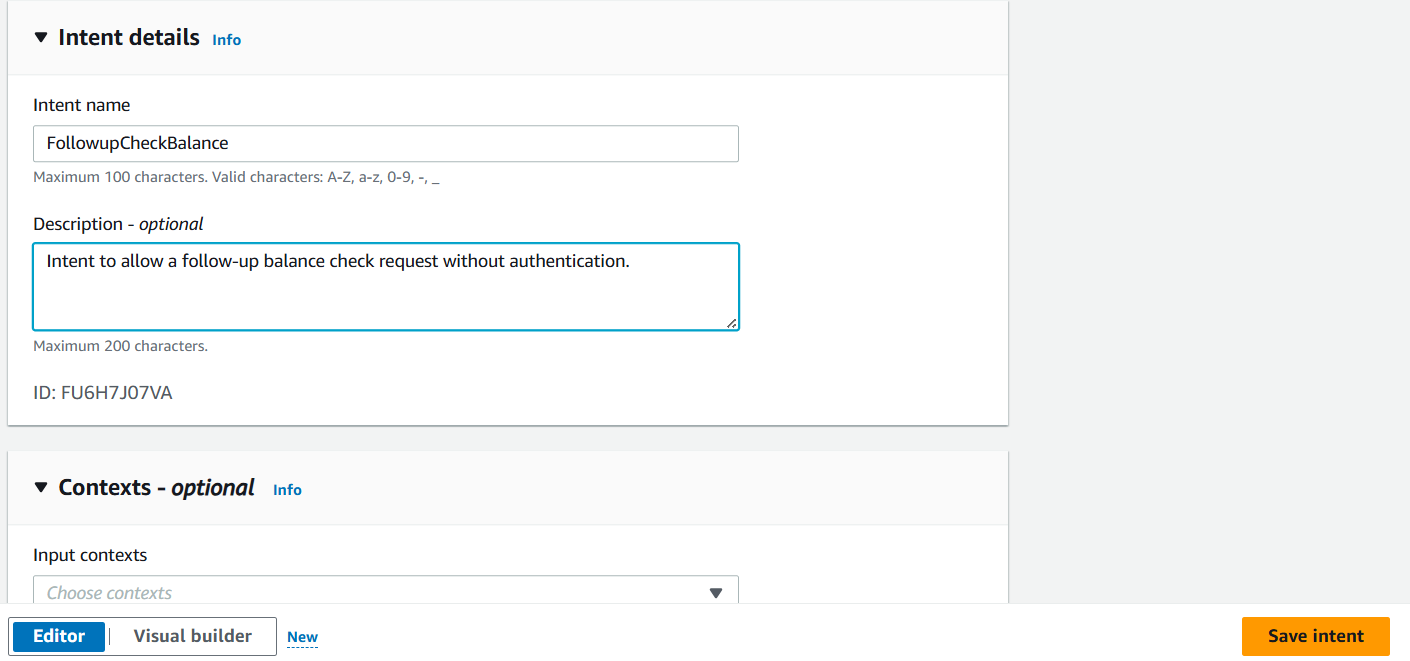
**What are context tags?**

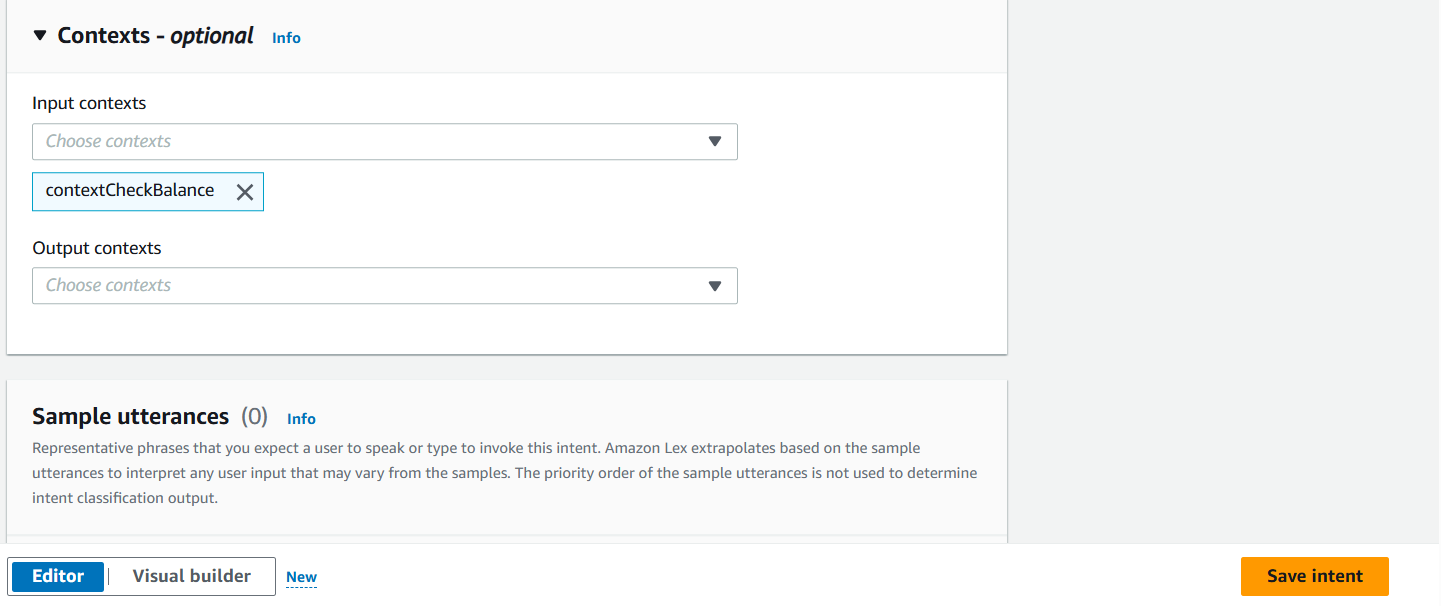
Context tags are tools for Amazon Lex to remember specific pieces of information gathered from a conversation, and reuse that information throughout the session with its user.

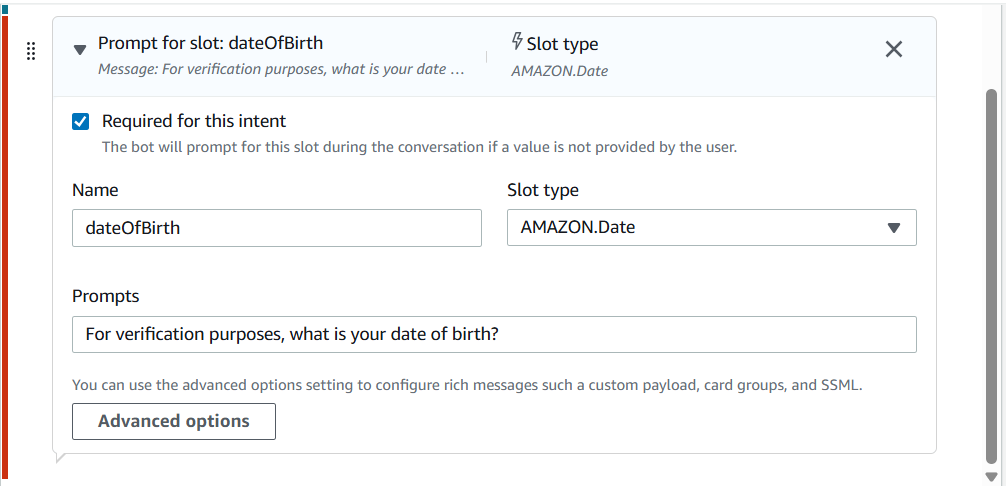
There are two types of context tags, they are output context tags and input context tags.

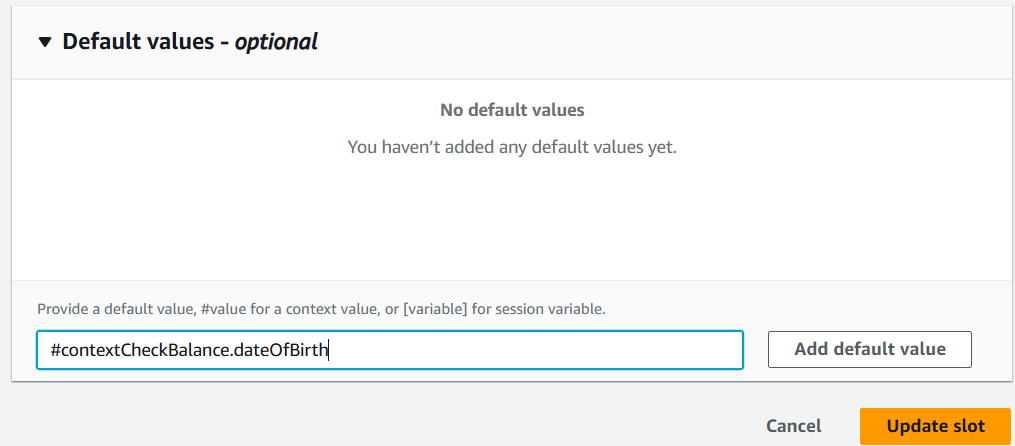
I created an output context tag called context CheckBalance, and I created this in the intent CheckBalance,

**Create the Follow up CheckBalance intent**

****

****

**Finishing Touches for Follow up CheckBalance**

****

I created a new intent called FollowupCheckBalance. The purpose of this intent is to let the user check another account's balance without having to provide their date of birth again.

This intent is related to the previous intent I made, CheckBalance, because FollowUpCheckBalance will only get triggered after the user has checked their balance once already (i.e. triggered CheckBalance).

I created an input context, contextCheckBalance, that is using the exact same tag as the output context tag I've set up in the CheckBalance intent. What this means is, input information we are looking for in this intent (Follow Up CheckBalance) can now be retrieved from the CheckBalance intent through this tag

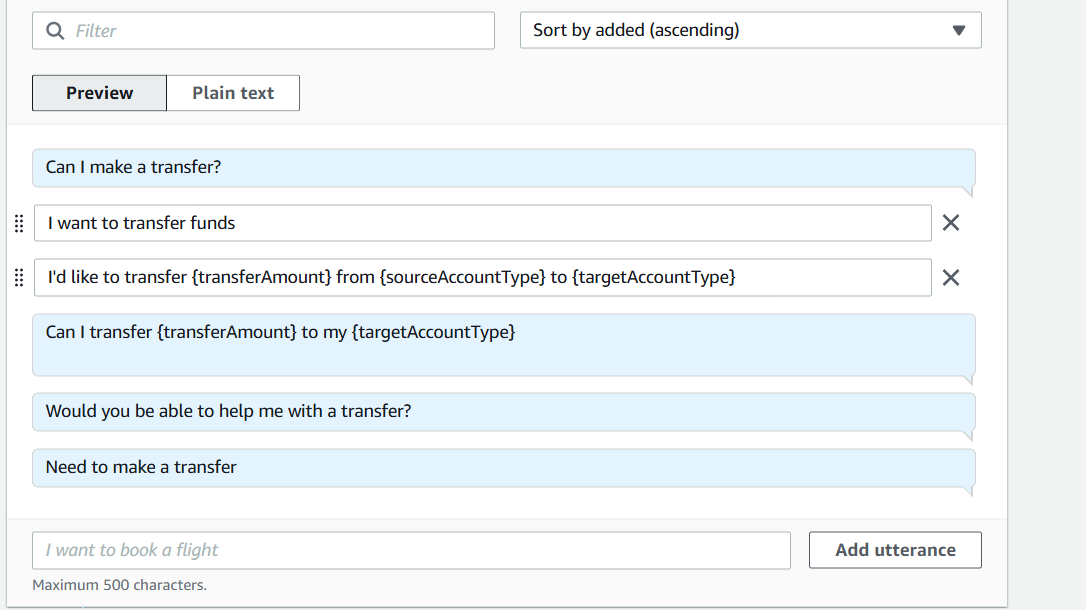
**Context tag**

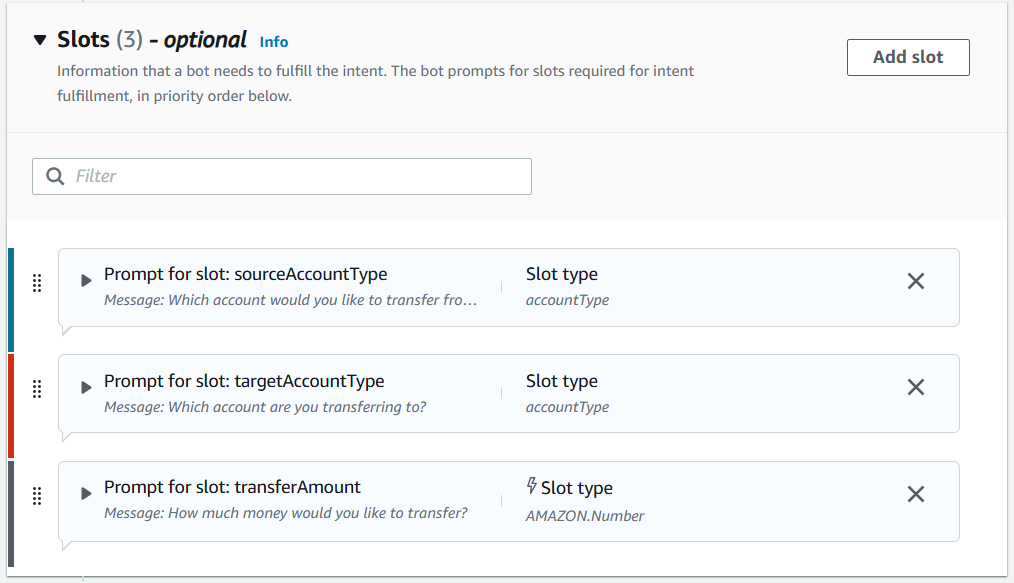
Conversation time! I built and tested my bot after creating the context tags and new intent.

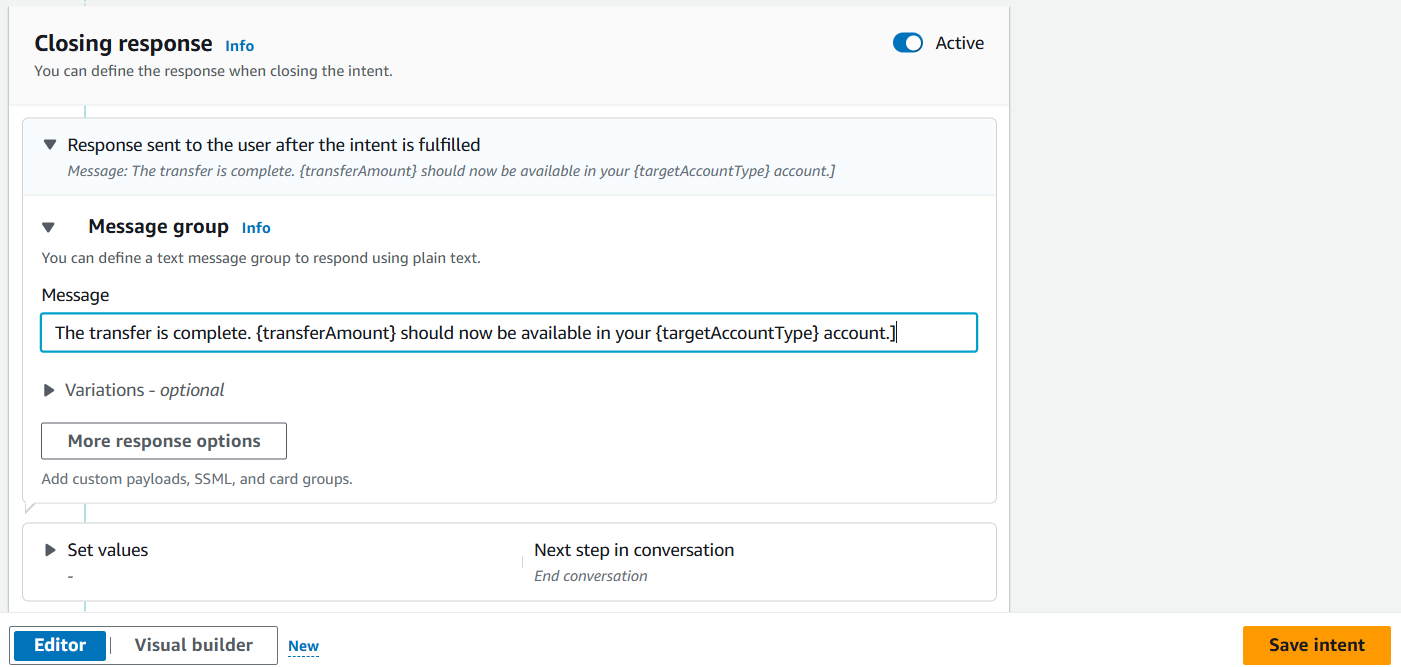
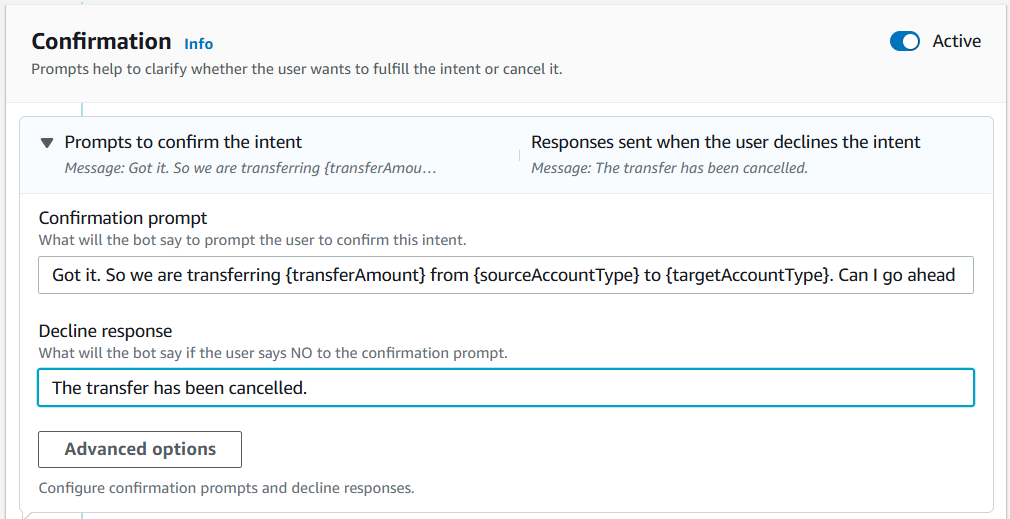
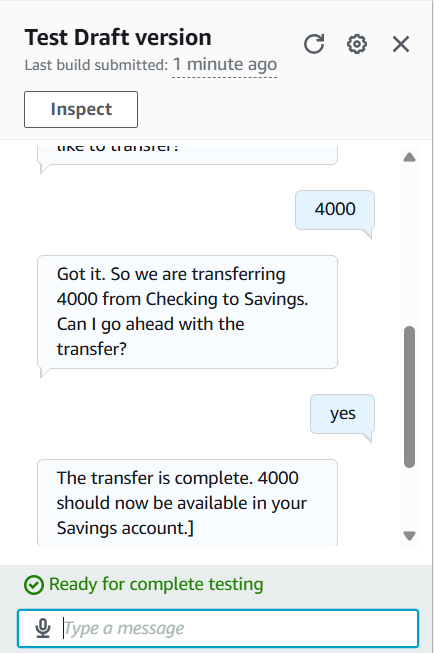
To see the context tags and the follow up in intent in action, I first triggered the CheckBalance intent, then I followed up with the utterance "what about savings" to trigger Follow Up Check Balance.

If I had gone straight to trying to trigger Follow Up CheckBalance without setting up any context, my chatbot would not have the context needed to fulfil the conversation. As a result, it will return the Fallback intent i.e. let the user know it doesn't understand the requestbeing made.

**Set Up Multiple Slots in a Lex Chatbot**

****



**Slots**

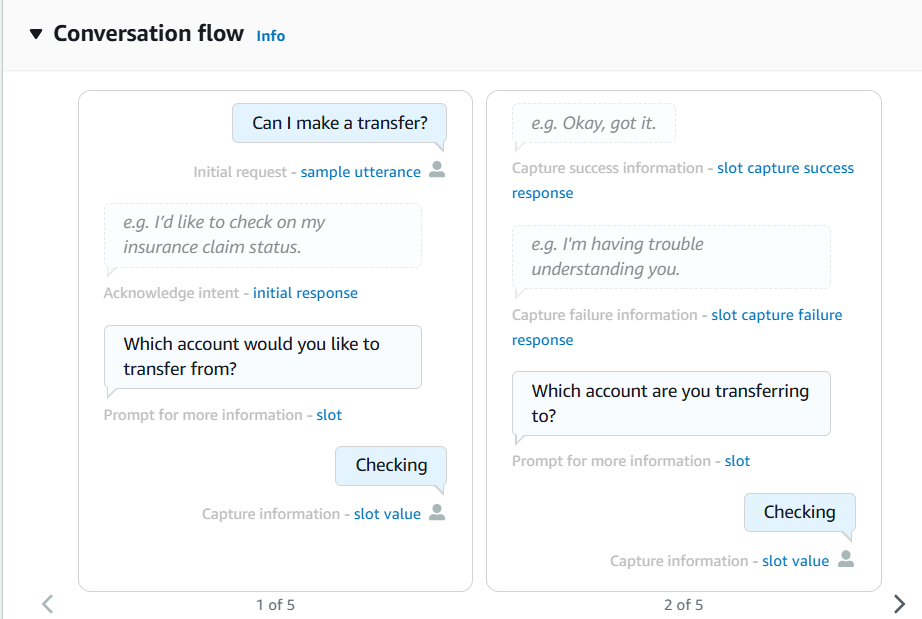
Slots are pieces of information that my chatbot needs in order to fulfill an intent

The final intent for my chatbot was TransferFunds, which will help the user transfer money between bank accounts

For this intent, I had to use the same slot type twice. This is because the TransferFunds intent involved two different accounts - the source account (i.e. the account that we are transferring money from) and the target account (i.e., the account that the money will land in)

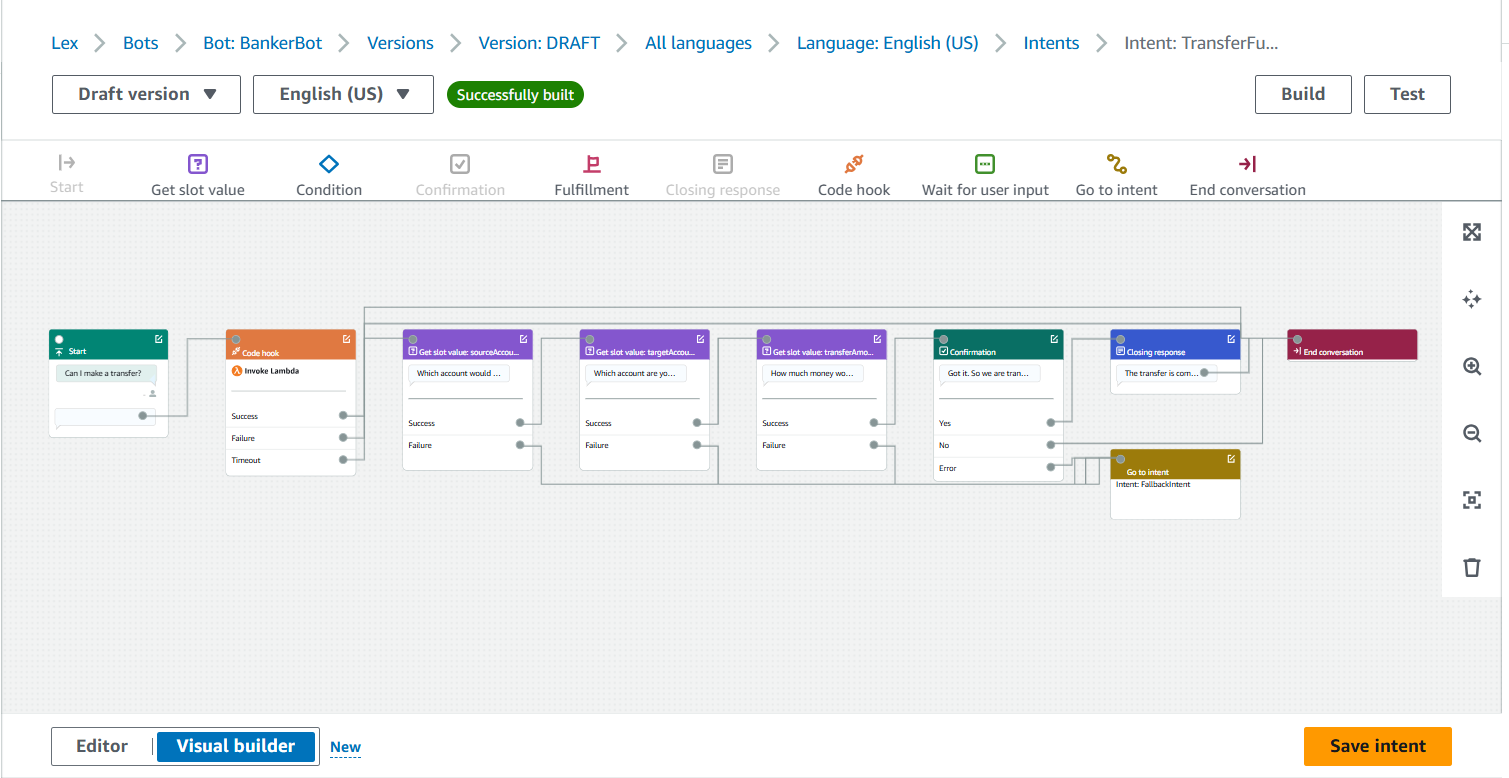
For this intent, I had to use the same slot type twice. This is because the TransferFunds intent involved two different accounts - the source account (i.e. the account that we are transferring money from) and the target account (i.e., the account that the money will land in)

**Cool features in Amazon Lex**

****

1. This flow will update as you continue editing this intent.
2. It shows every step in a conversation in a logical, chronological order.
3. You'll also see some blank 'ghost like' responses. These are recommendations for what you could add to your Intent set up and they're clickable!
4. If you click on the chat bubble, you'll get taken to an edit screen.

**Visual builder**



**What is amazon lex**

Amazon Lex is an AWS service that allows developers to build conversational interfaces into applications using voice and text. It provides the same conversational engine that powers Amazon Alexa, enabling the creation of sophisticated, natural language chatbots and voice assistants.

**Thoughts on Amazon Lex**

Amazon Lex is a conversational interface service by Amazon Web Services (AWS) that enables you to build chatbots and other voice and text-based applications. It leverages natural language understanding (NLU) and automatic speech recognition (ASR) to create interactive experiences. Essentially, Lex is the technology behind Alexa and similar virtual assistants, allowing you to build your own chatbot or voice application

**Thanks**

**Rony Joseph Thomas**