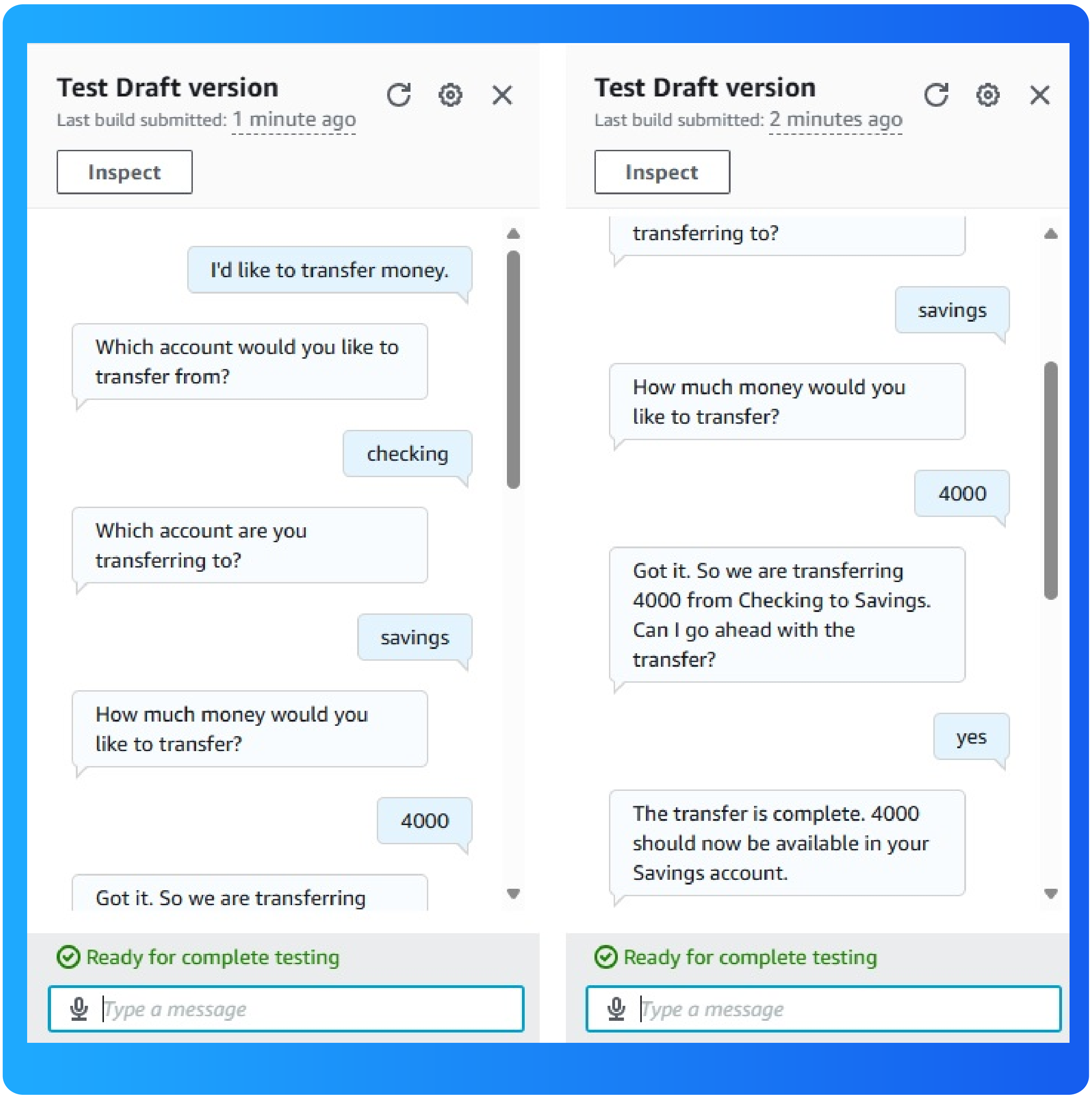
Build a Chatbot with Amazon Lex

Gabriel Taveira Mazer



Test Draft version
Last build submitted: 1 minute ago
Inspect

I'd like to transfer money.

Which account would you like to transfer from?

checking

Which account are you transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring

Ready for complete testing
Type a message

Test Draft version
Last build submitted: 2 minutes ago
Inspect

transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring 4000 from Checking to Savings. Can I go ahead with the transfer?

yes

The transfer is complete. 4000 should now be available in your Savings account.

Gabriel Mazer

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building chatbots using voice and text. It allows the creation of conversational interfaces that understand natural language, helping automate tasks like customer service, making user interactions more efficient.

How I used Amazon Lex in this project

I used Amazon Lex to build a chatbot that understands user inputs, responds to greetings, and helps with tasks like checking account balances. I set up intents, added FallbackIntent for unrecognized inputs, and included variations for better response

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

One thing I didn't expect in this project was how flexible Amazon Lex is with handling different user inputs. The ability to customize fallback responses and variations really showed me how easily I can tailor the chatbot to be more personalized.

This project took me...

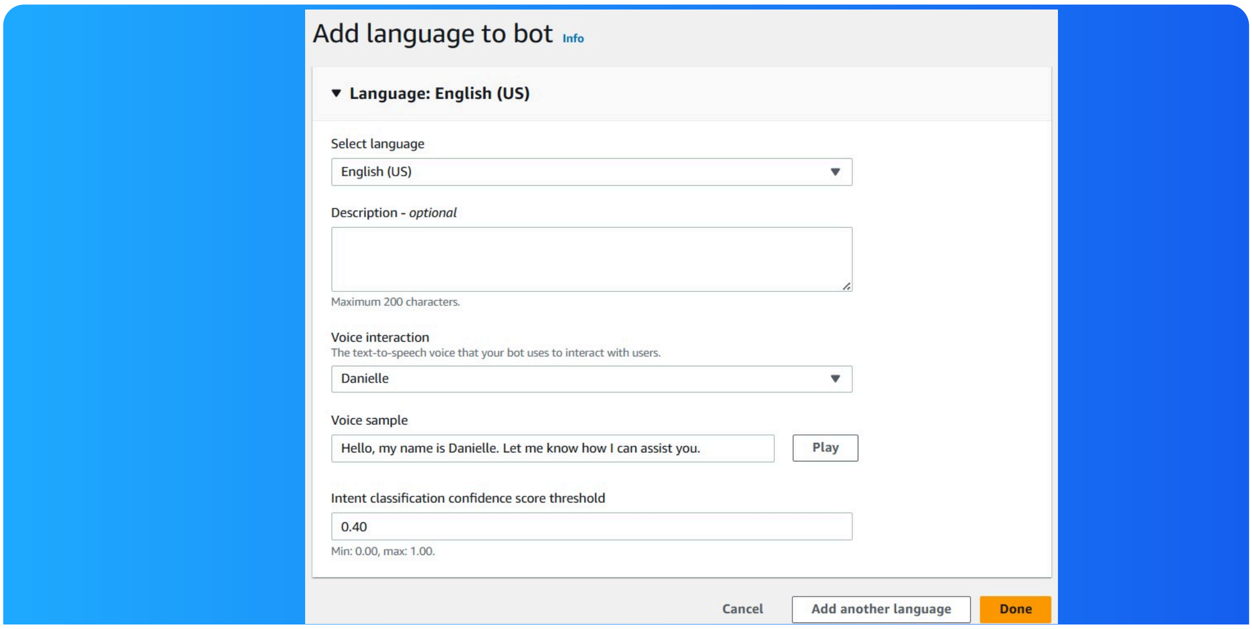
This project took me around one hour to complete, from setting up the intents to configuring FallbackIntent and testing the chatbot's interactions. The process was efficient and allowed me to quickly implement the features.

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me about 30 min, including configuring the intents, slots, and utterances. The interface made it easy to integrate.

While creating my chatbot, I also created a role with basic permissions because it ensures the chatbot has the necessary access to AWS services, like Lambda or CloudWatch, without over-privileging, maintaining security and proper functionality.

In terms of the intent classification confidence score, I kept the default value of 0.40. This way, Lex will only trigger an intent if it's at least 40% confident that the user's input matches. This ensure better accuracy, avoiding misclassification.



Add language to bot

▼ Language: English (US)

Select language: English (US)

Description - *optional*

Maximum 200 characters.

Voice interaction
The text-to-speech voice that your bot uses to interact with users.
Danielle

Voice sample: Hello, my name is Danielle. Let me know how I can assist you.

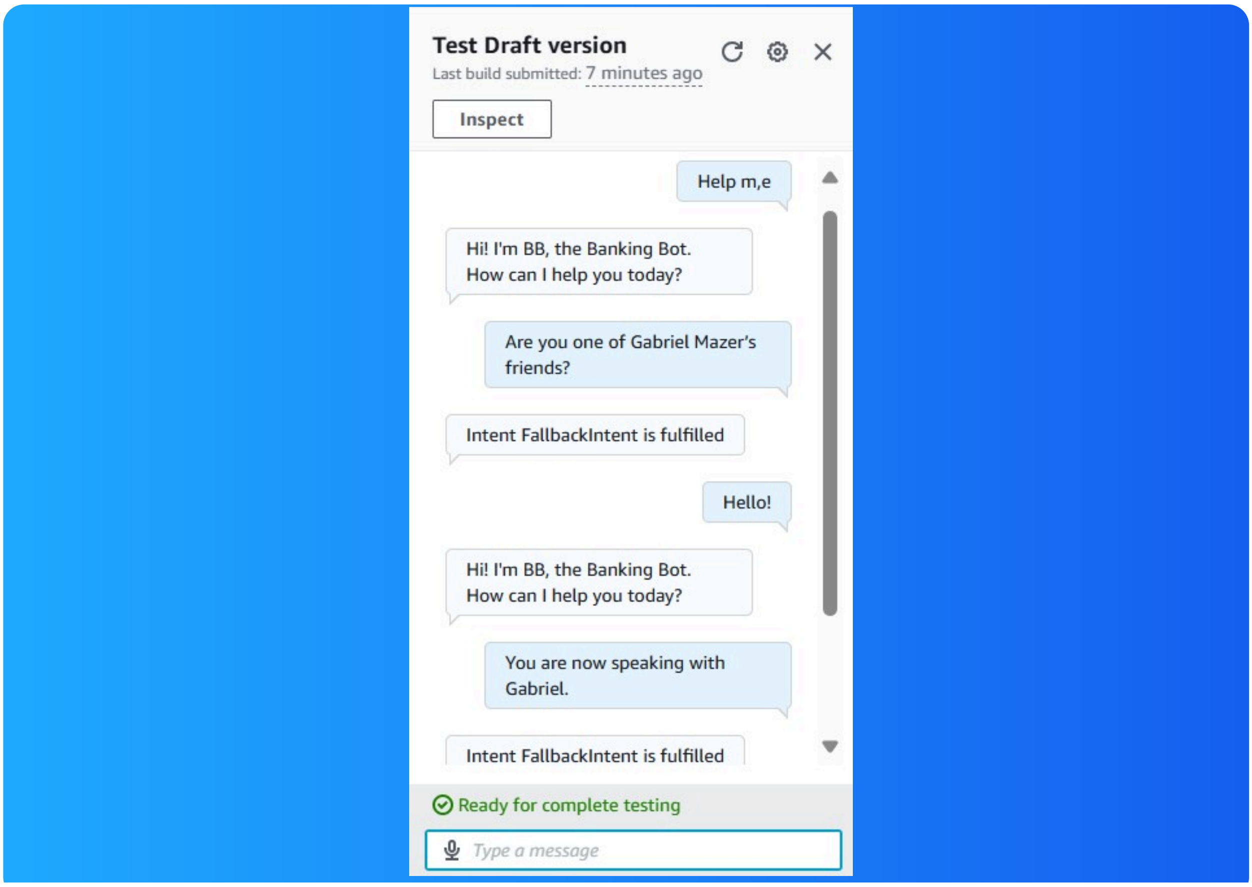
Intent classification confidence score threshold: 0.40
Min: 0.00, max: 1.00.

Gabriel Mazer

Intents

Intents are predefined actions or tasks that a chatbot recognizes and executes based on user input. In Amazon Lex, intents define the purpose of the user's request, like booking a hotel or checking the weather, guiding the chatbot’s responses.

I created my first intent, WelcomeIntent, to greet users when they start interacting with the chatbot. The response is: 'Hi! I'm BB, the Banking Bot. How can I help you today?' This sets a friendly tone and invites the user to engage with the bot.


Test Draft version
Last build submitted: 7 minutes ago
Inspect

Help m,e

Hi! I'm BB, the Banking Bot.
How can I help you today?

Are you one of Gabriel Mazer's friends?

Intent FallbackIntent is fulfilled

Hello!

Hi! I'm BB, the Banking Bot.
How can I help you today?

You are now speaking with Gabriel.

Intent FallbackIntent is fulfilled

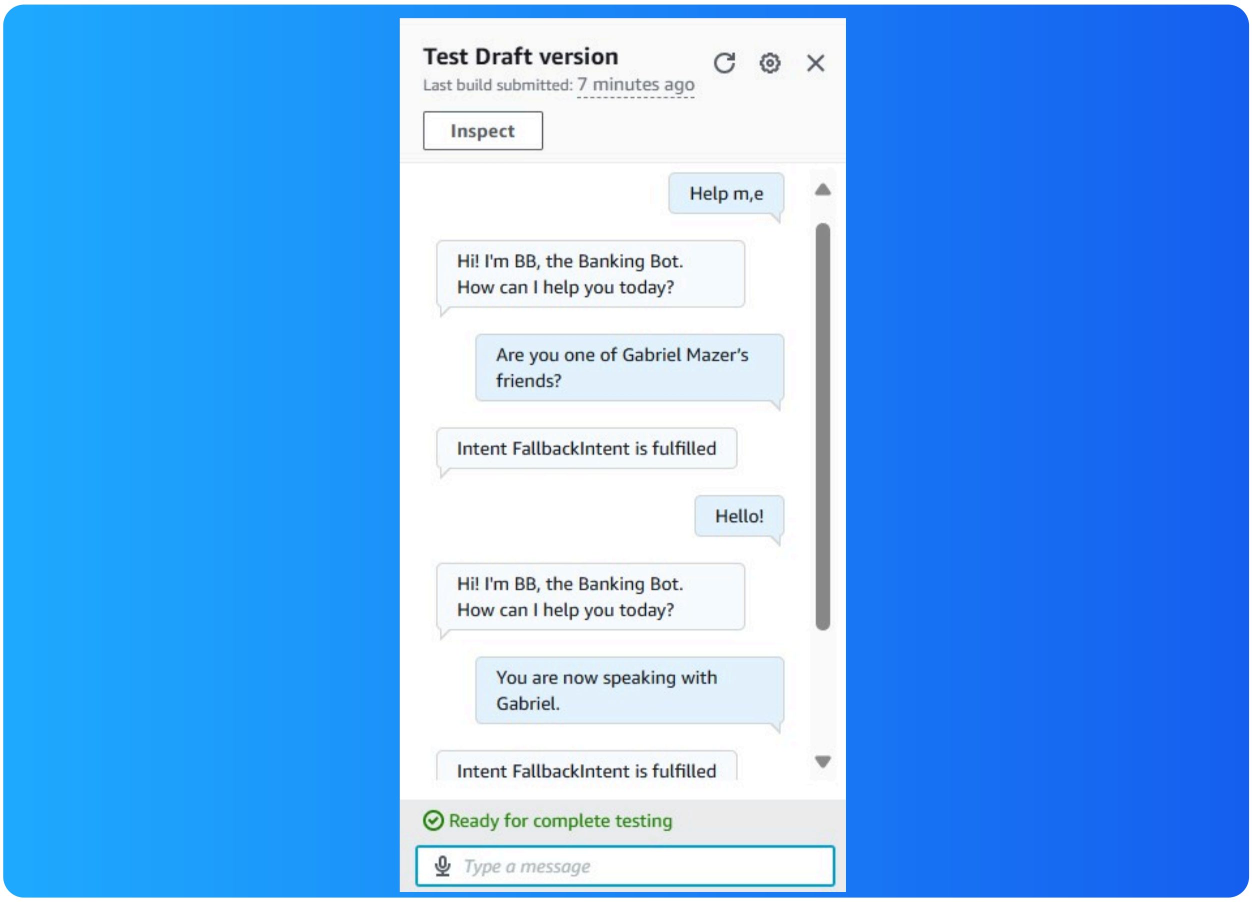
Ready for complete testing
Type a message

Gabriel Mazer

FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter greetings like 'Hello!' or 'Help m,e.' These phrases trigger the WelcomeIntent, which prompts the chatbot to introduce itself and offer assistance to the user.

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered 'Are you one of Gabriel Mazer's friends?' and 'You are now speaking with Gabriel.' This occurred because these inputs didn't match any of the predefined intents



Test Draft version

Last build submitted: 7 minutes ago

Help m,e

Hi! I'm BB, the Banking Bot.
How can I help you today?

Are you one of Gabriel Mazer's friends?

Intent FallbackIntent is fulfilled

Hello!

Hi! I'm BB, the Banking Bot.
How can I help you today?

You are now speaking with Gabriel.

Intent FallbackIntent is fulfilled

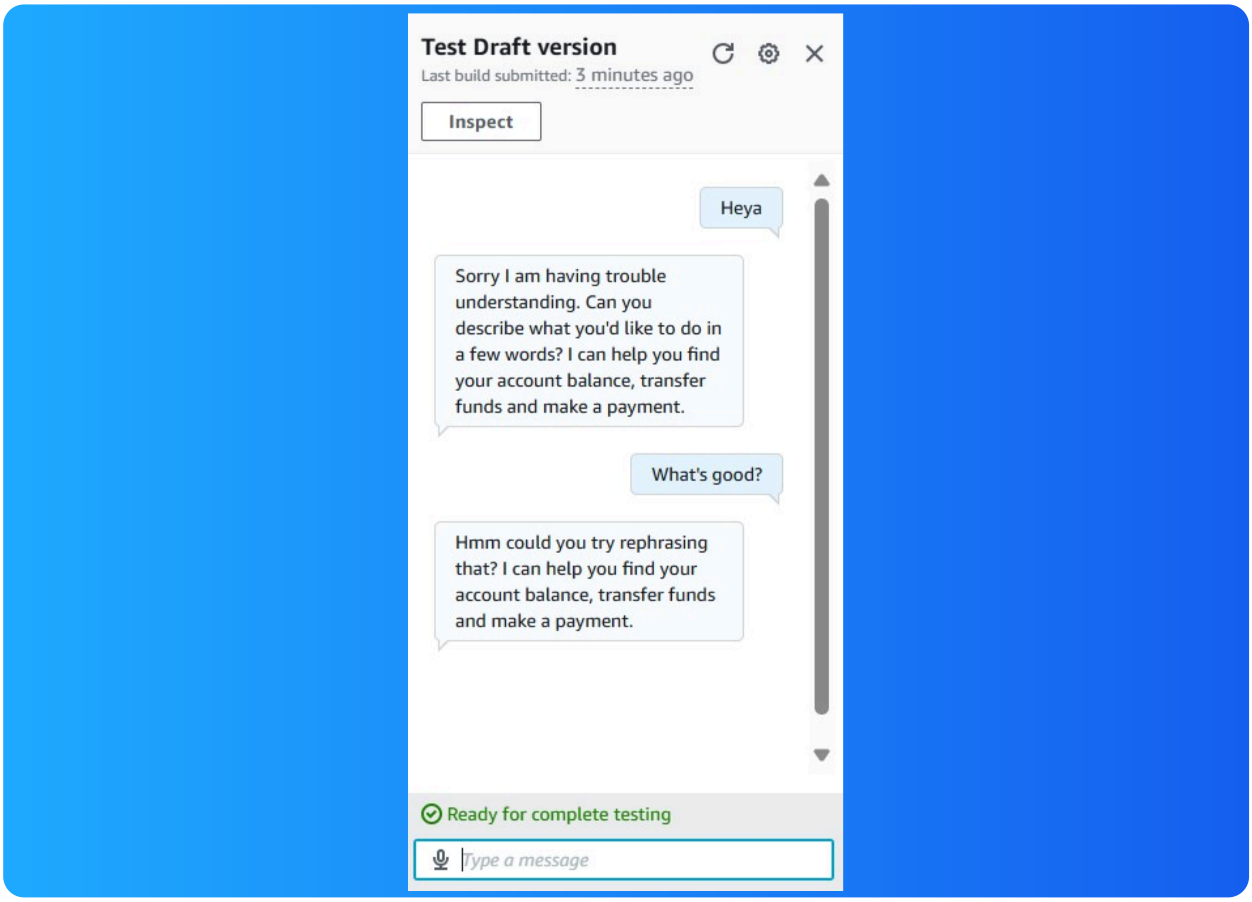
Ready for complete testing

Gabriel Mazer

Variations

To configure FallbackIntent, I had to create my own closing response in the intent's set up page "Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words?...".

I also added variations! What this means for an end user is they get to see different forms of my chatbot's closing response.


Test Draft version
Last build submitted: 3 minutes ago
Inspect

Heya

Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment.

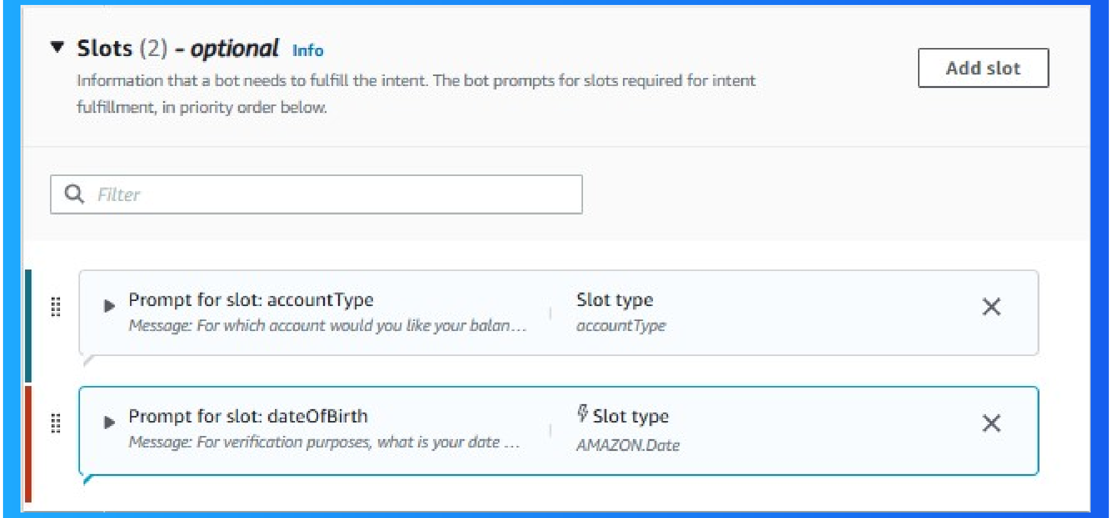
What's good?

Hmm could you try rephrasing that? I can help you find your account balance, transfer funds and make a payment.

Ready for complete testing
Type a message

Custom Slots

Gabriel Taveira Mazer



Slots (2) - *optional*

Information that a bot needs to fulfill the intent. The bot prompts for slots required for intent fulfillment, in priority order below.

Prompt for slot: accountType
Message: For which account would you like your balan…
Slot type
accountType

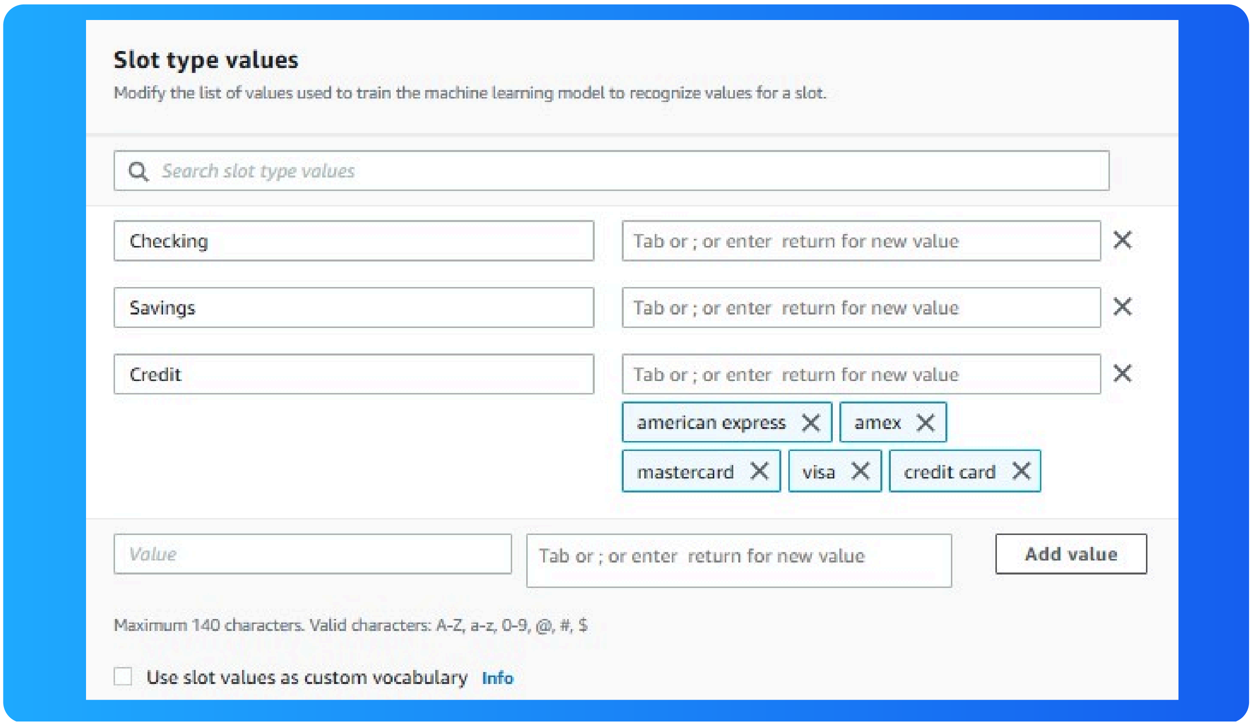
Prompt for slot: dateOfBirth
Message: For verification purposes, what is your date …
Slot type
AMAZON.Date

Slots

Slots are variables that capture specific information from users during their interaction with the chatbot. They help the chatbot gather details like account types or other necessary inputs to fulfill a user's request.

In this project, I created a custom slot type to allow users to specify account types, such as checking, savings, or credit accounts, making the chatbot more adaptable to different financial requests.

This slot type has restricted slot values, which means users can only select from predefined options, like 'Visa' or 'Mastercard.' This helps ensure valid responses and improves the chatbot's accuracy.



Slot type values

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

Checking	
Savings	
Credit	american express amex mastercard visa credit card

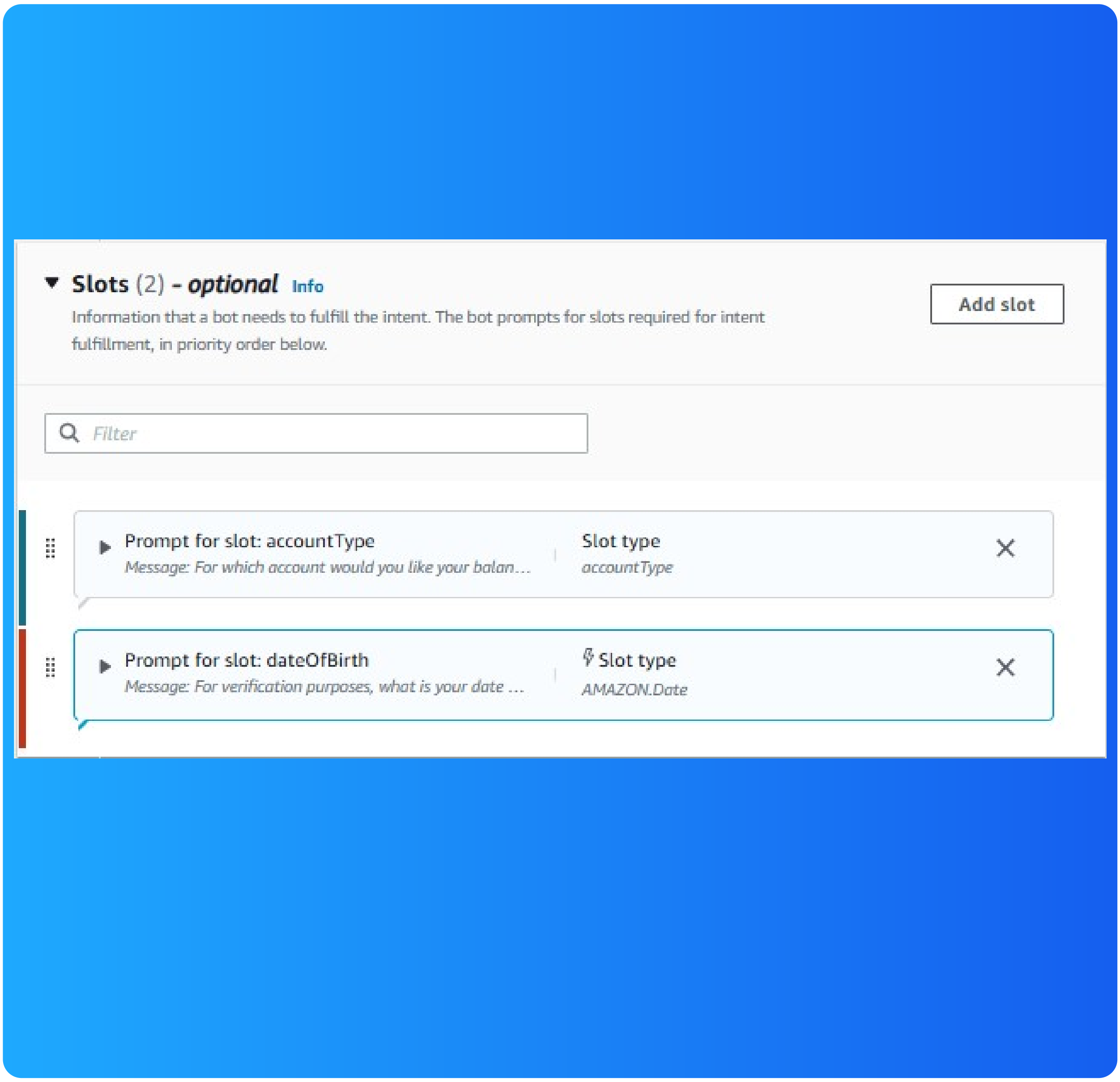
Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

☐ Use slot values as custom vocabulary

Gabriel Mazer

Connecting slots with intents

I associated my custom slot with CheckBalance, which prompts the user to provide their account type and date of birth for verification. It helps the chatbot check the balance of a specified account after collecting this information.



Slots (2) - *optional* Info

Information that a bot needs to fulfill the intent. The bot prompts for slots required for intent fulfillment, in priority order below.

Prompt for slot: accountType

Message: For which account would you like your balan…

Slot type

accountType

Prompt for slot: dateOfBirth

Message: For verification purposes, what is your date …

Slot type

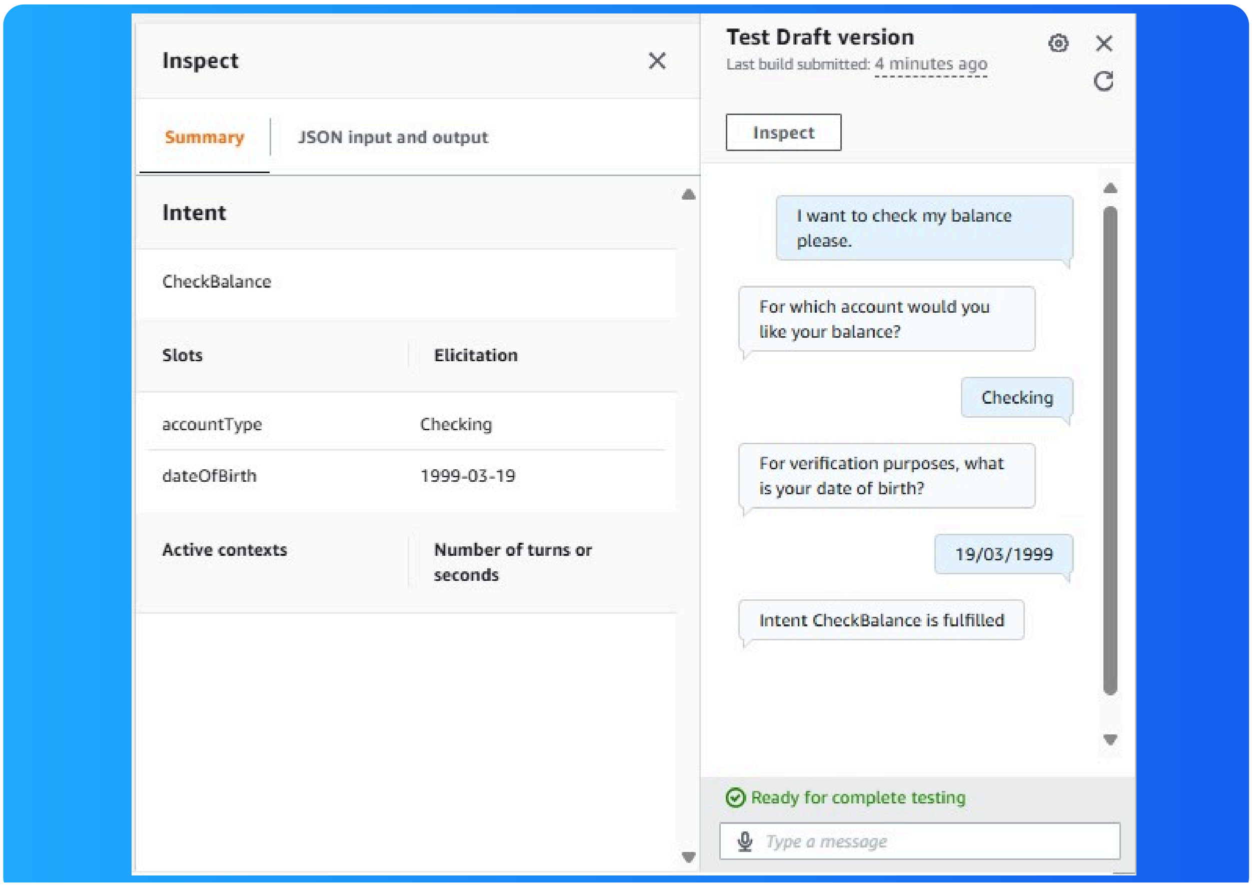
AMAZON.Date

Gabriel Mazer

Slot values in utterances

I included slot values in some of the utterances (i.e., user inputs) by using placeholders for accountType. For example, 'What’s the balance in my {accountType} account?' allows the bot to capture and use the user’s account type dynamically.

By adding custom slots in utterances, the chatbot can collect specific information, like the account type, directly from the user input. This makes the conversation more natural and ensures the bot can accurately fulfill the user’s request.



Intent

CheckBalance

Slots	Elicitation
accountType	Checking
dateOfBirth	1999-03-19

Active contexts	Number of turns or seconds

I want to check my balance please.

For which account would you like your balance?

Checking

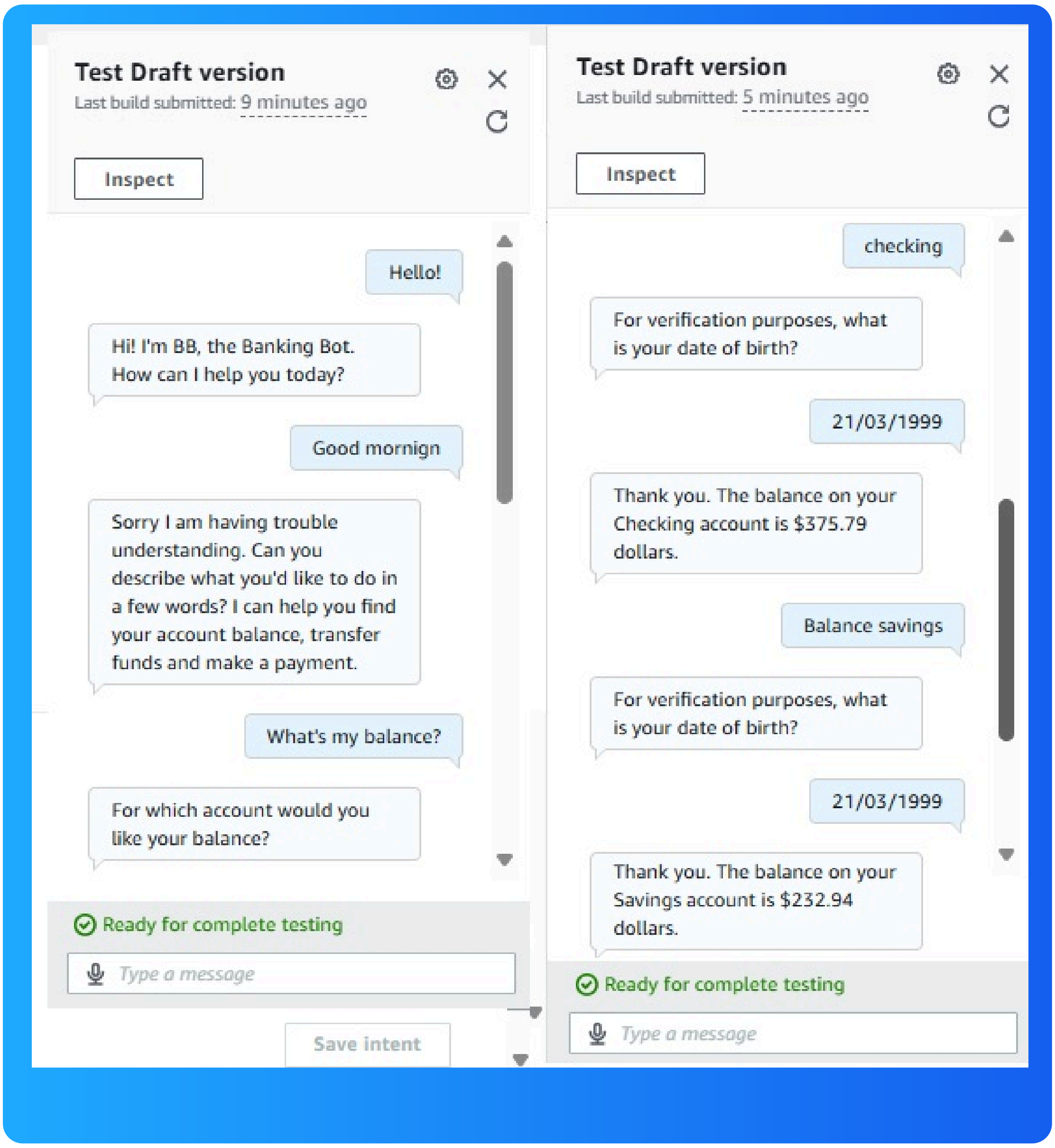
For verification purposes, what is your date of birth?

19/03/1999

Intent CheckBalance is fulfilled

Connect with Lambda

Gabriel Taveira Mazer


Test Draft version
Last build submitted: 9 minutes ago
Inspect

Hello!

Hi! I'm BB, the Banking Bot. How can I help you today?

Good mornign

Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment.

What's my balance?

For which account would you like your balance?

Ready for complete testing
Type a message
Save intent
Test Draft version
Last build submitted: 5 minutes ago
Inspect

checking

For verification purposes, what is your date of birth?

21/03/1999

Thank you. The balance on your Checking account is $375.79 dollars.

Balance savings

For verification purposes, what is your date of birth?

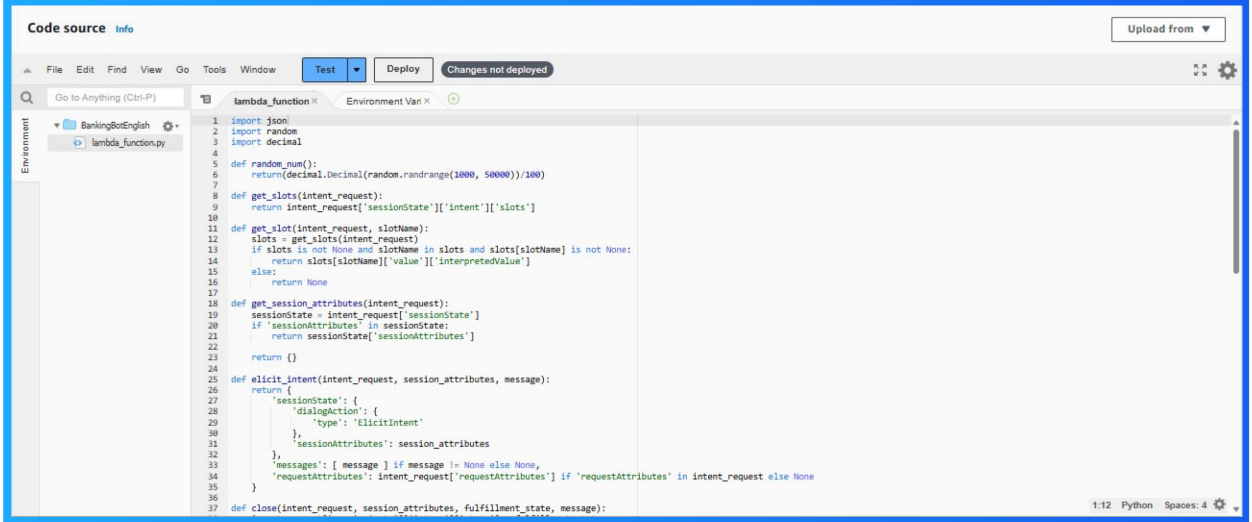
21/03/1999

Thank you. The balance on your Savings account is $232.94 dollars.

AWS Lambda Functions

AWS Lambda is a serverless compute service that automatically runs code in response to events. It scales automatically, eliminating the need to manage servers. Lambda is ideal for running backend services, including API requests and data processing.

In this project, I created a Lambda function to simulate retrieving a user's account balance. The function generates a random balance number and sends it to Amazon Lex, which then delivers the balance information to the user through the chatbot.



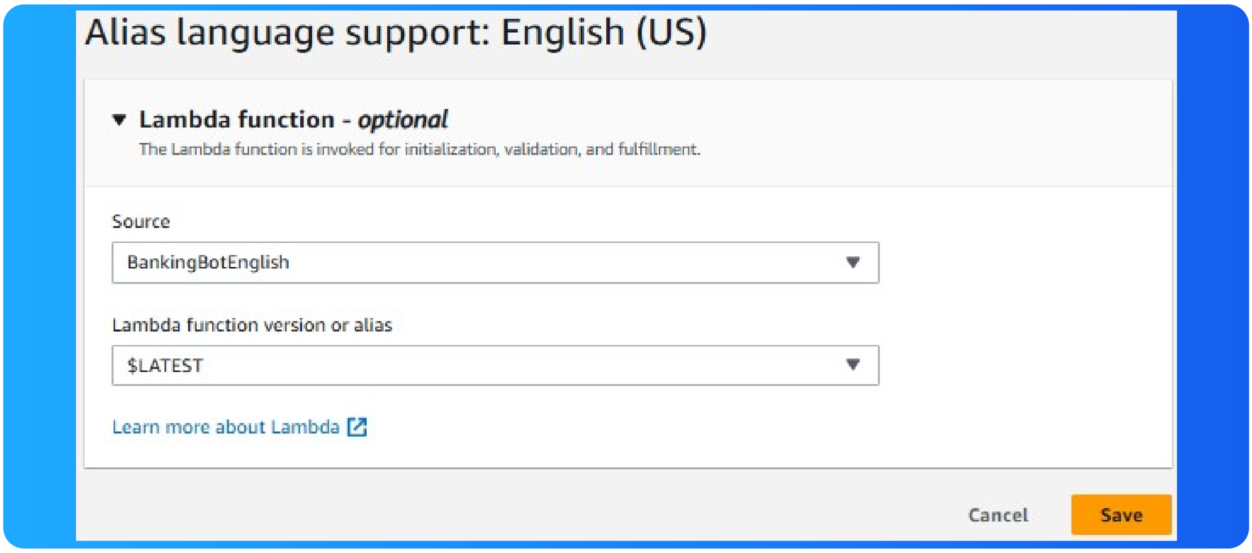
```python
import json
import random
import decimal

def random_num():
    return(decimal.Decimal(random.randrange(1000, 50000))/100)

def get_slots(intent_request):
    return intent_request['sessionState']['intent']['slots']

def get_slot(intent_request, slotName):
    slots = get_slots(intent_request)
    if slots is not None and slotName in slots and slots[slotName] is not None:
        return slots[slotName]['value']['interpretedValue']
    else:
        return None

def get_session_attributes(intent_request):
    sessionState = intent_request['sessionState']
    if 'sessionAttributes' in sessionState:
        return sessionState['sessionAttributes']

    return {}

def elicit_intent(intent_request, session_attributes, message):
    return {
        'sessionState': {
            'dialogAction': {
                'type': 'ElicitIntent'
            },
            'sessionAttributes': session_attributes
        },
        'messages': [ message ] if message != None else None,
        'requestAttributes': intent_request['requestAttributes'] if 'requestAttributes' in intent_request else None
    }

def close(intent_request, session_attributes, fulfillment_state, message):
```

Chatbot Alias

An alias is a pointer to a specific version of a bot in Amazon Lex. It allows developers to manage different stages of a bot, like development or production, without affecting other versions.

TestBotAlias is an alias created for the purpose of testing the chatbot. It allows the bot to invoke the latest configurations and Lambda integrations while undergoing tests without affecting the main bot.

To connect Lambda with my BankerBot, I visited my bot's TestBotAlias and selected the Lambda function 'BankingBotEnglish,' ensuring that the function is invoked for initialization, validation, and fulfillment.



Alias language support: English (US)

▼ **Lambda function** - *optional*

The Lambda function is invoked for initialization, validation, and fulfillment.

Source: BankingBotEnglish

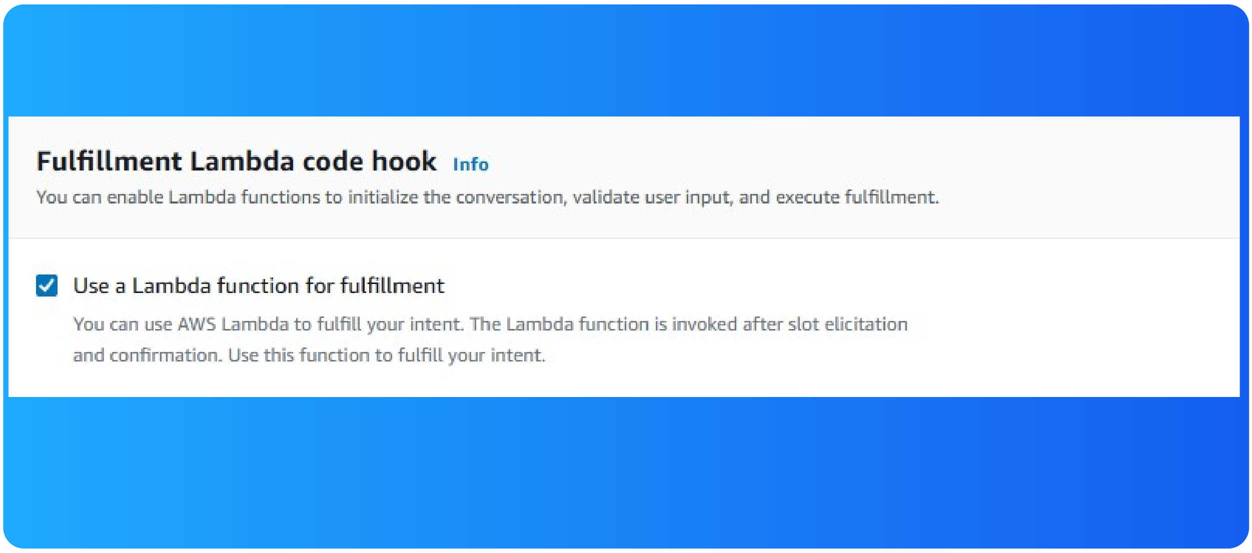
Lambda function version or alias: $LATEST

Code Hooks

A code hook is a function triggered during an intent’s fulfillment process. It allows custom code (like a Lambda function) to be executed based on the user's input.

Even though I already connected my Lambda function with my chatbot's alias, I had to use code hooks because they allow the bot to execute specific code, like retrieving a random bank balance, during the intent's fulfillment.

I could find code hooks at the fulfillment panel of the CheckBalance intent, under Advanced options, where I linked my Lambda function to run on successful intent fulfillment.



Fulfillment Lambda code hook Info

You can enable Lambda functions to initialize the conversation, validate user input, and execute fulfillment.

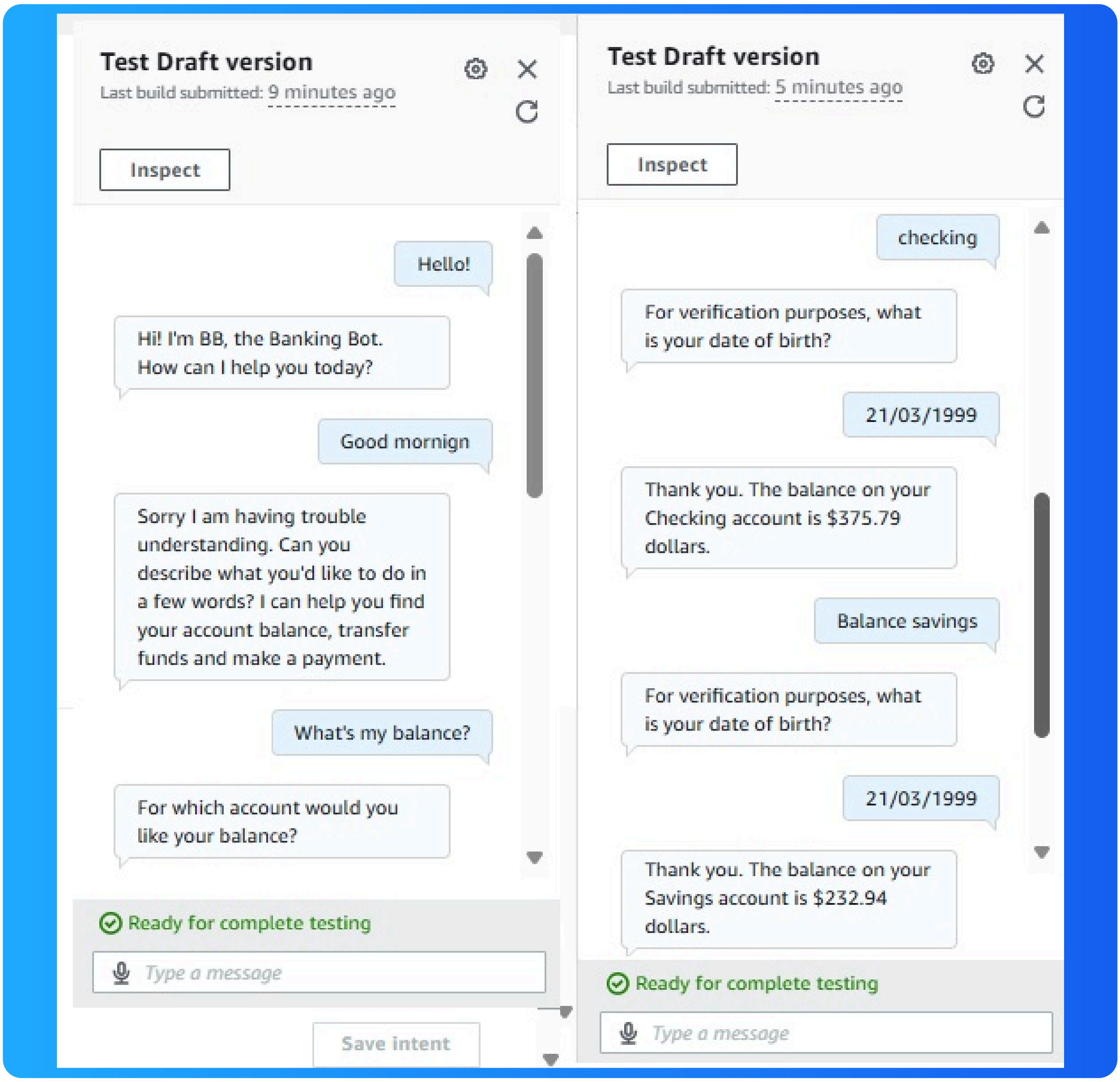
☑ Use a Lambda function for fulfillment

You can use AWS Lambda to fulfill your intent. The Lambda function is invoked after slot elicitation and confirmation. Use this function to fulfill your intent.

Gabriel Mazer

The final result!

I've set up my chatbot to trigger Lambda and return a random dollar figure when the user asks for their account balance. The Lambda function simulates this by generating a random amount for each request.



Test Draft version
Last build submitted: 9 minutes ago
Inspect

Hello!

Hi! I'm BB, the Banking Bot. How can I help you today?

Good mornign

Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment.

What's my balance?

For which account would you like your balance?

Ready for complete testing
Type a message
Save intent
Test Draft version
Last build submitted: 5 minutes ago
Inspect

checking

For verification purposes, what is your date of birth?

21/03/1999

Thank you. The balance on your Checking account is $375.79 dollars.

Balance savings

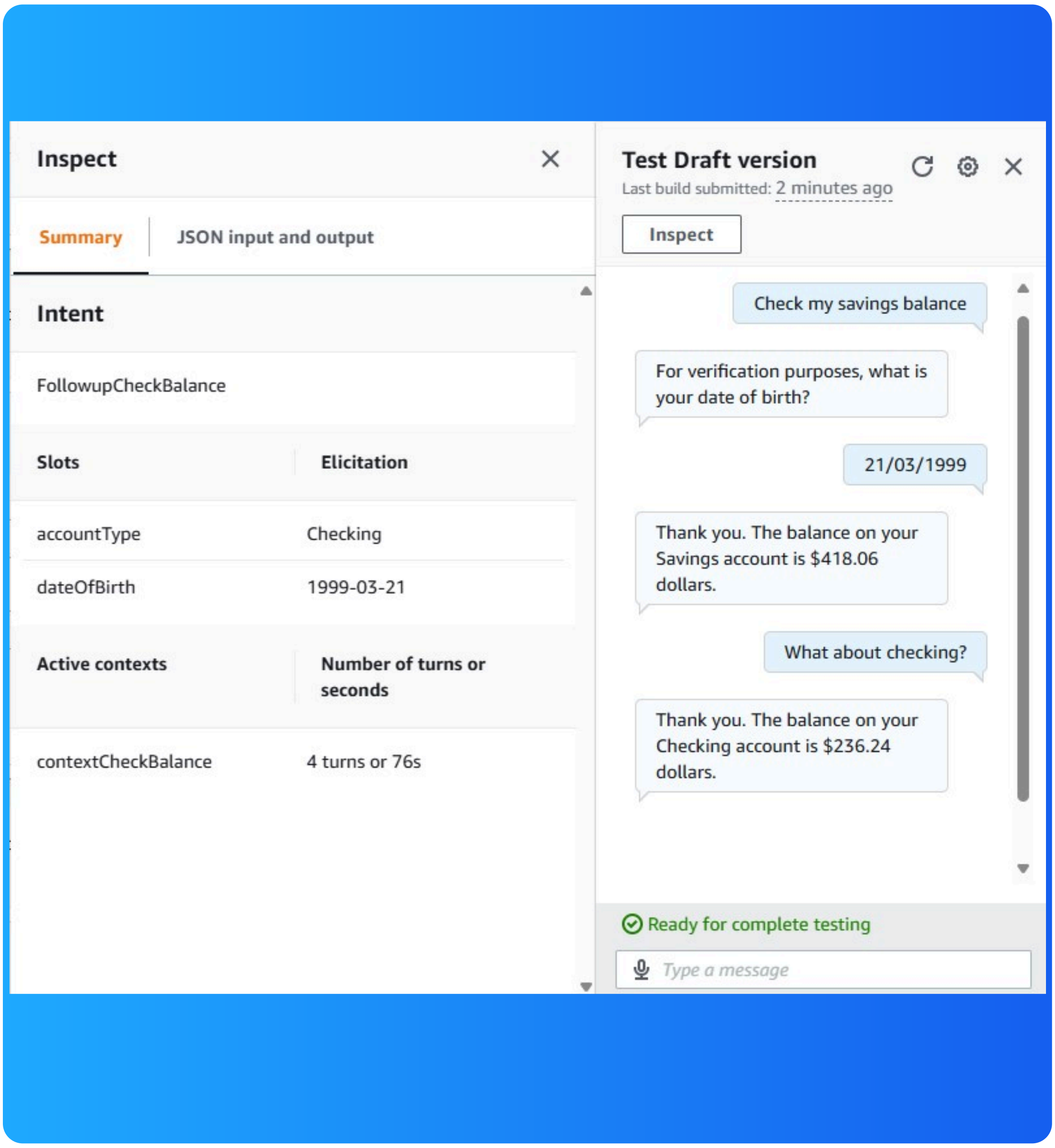
For verification purposes, what is your date of birth?

21/03/1999

Thank you. The balance on your Savings account is $232.94 dollars.

Save User Info

Gabriel Taveira Mazer



Intent

FollowupCheckBalance

Slots	Elicitation
accountType	Checking
dateOfBirth	1999-03-21

Active contexts	Number of turns or seconds
contextCheckBalance	4 turns or 76s

Check my savings balance

For verification purposes, what is your date of birth?

21/03/1999

Thank you. The balance on your Savings account is $418.06 dollars.

What about checking?

Thank you. The balance on your Checking account is $236.24 dollars.

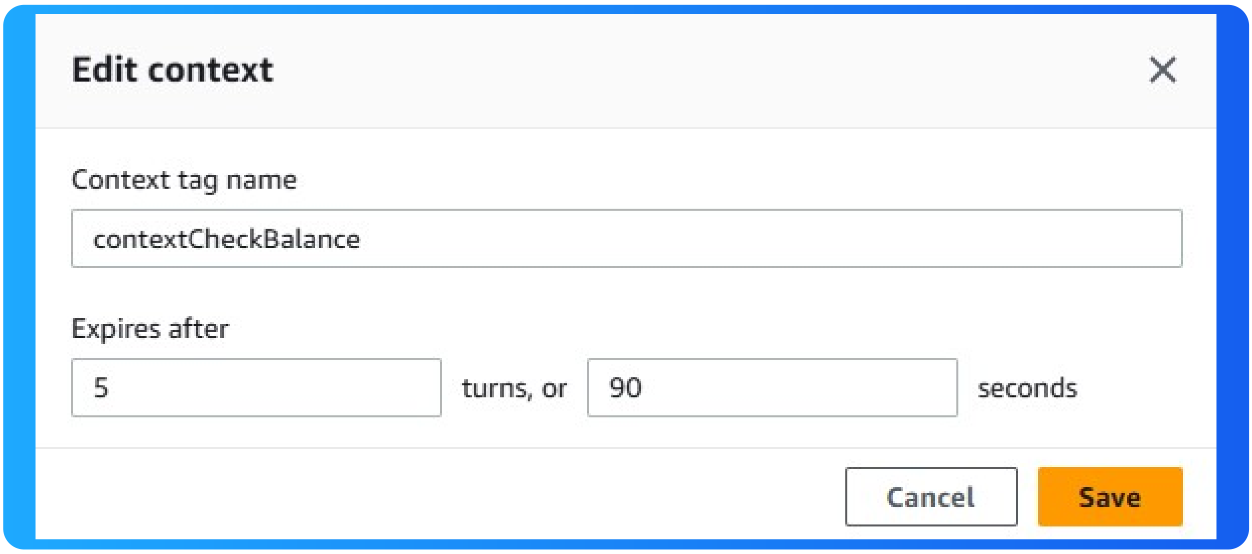
Gabriel Mazer

Context Tags

Context tags are markers used to store information between different user interactions in Amazon Lex. They help the chatbot remember key details, such as account type, across multiple intents or conversation turns, improving the conversation.

There are two types of context tags: output and input. Output tags store information after an intent is triggered, while input tags enable subsequent intents to retrieve that information, allowing the bot to carry details across multiple interactions

I created an output context tag called contextCheckBalance. This context tag was created in the CheckBalance intent.



Edit context

Context tag name

contextCheckBalance

Expires after

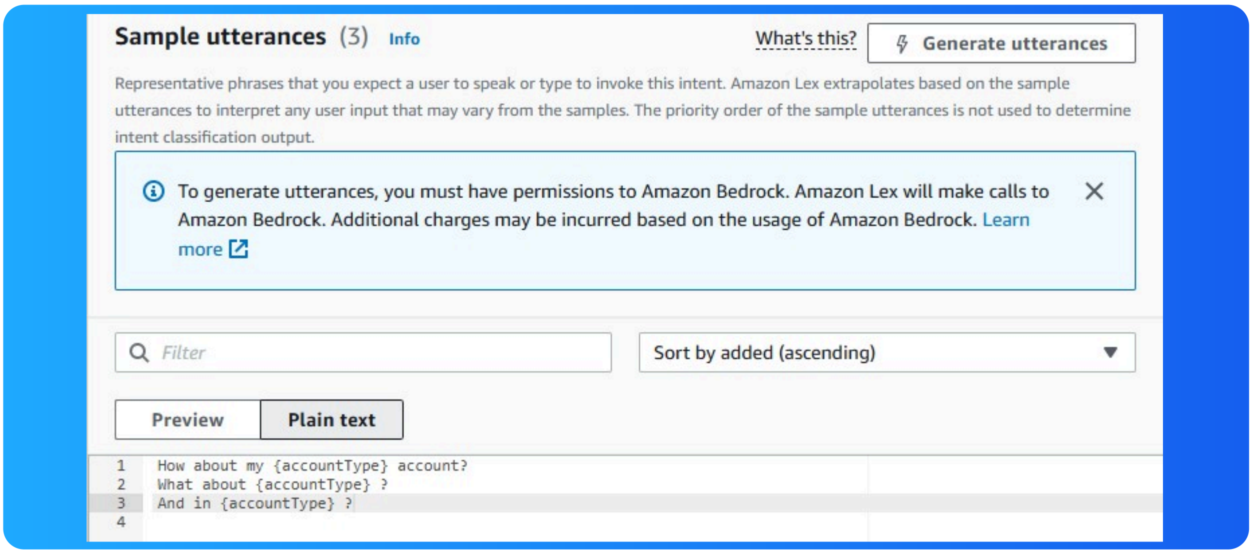
5 turns, or 90 seconds

Gabriel Mazer

FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to let users check their balance again without being asked for their birthday, using the contextCheckBalance tag to carry over the previous verification.

This intent is connected to the previous intent I made, CheckBalance, because both use the same contextCheckBalance tag, allowing the user's date of birth to carry over without needing re-verification.



Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

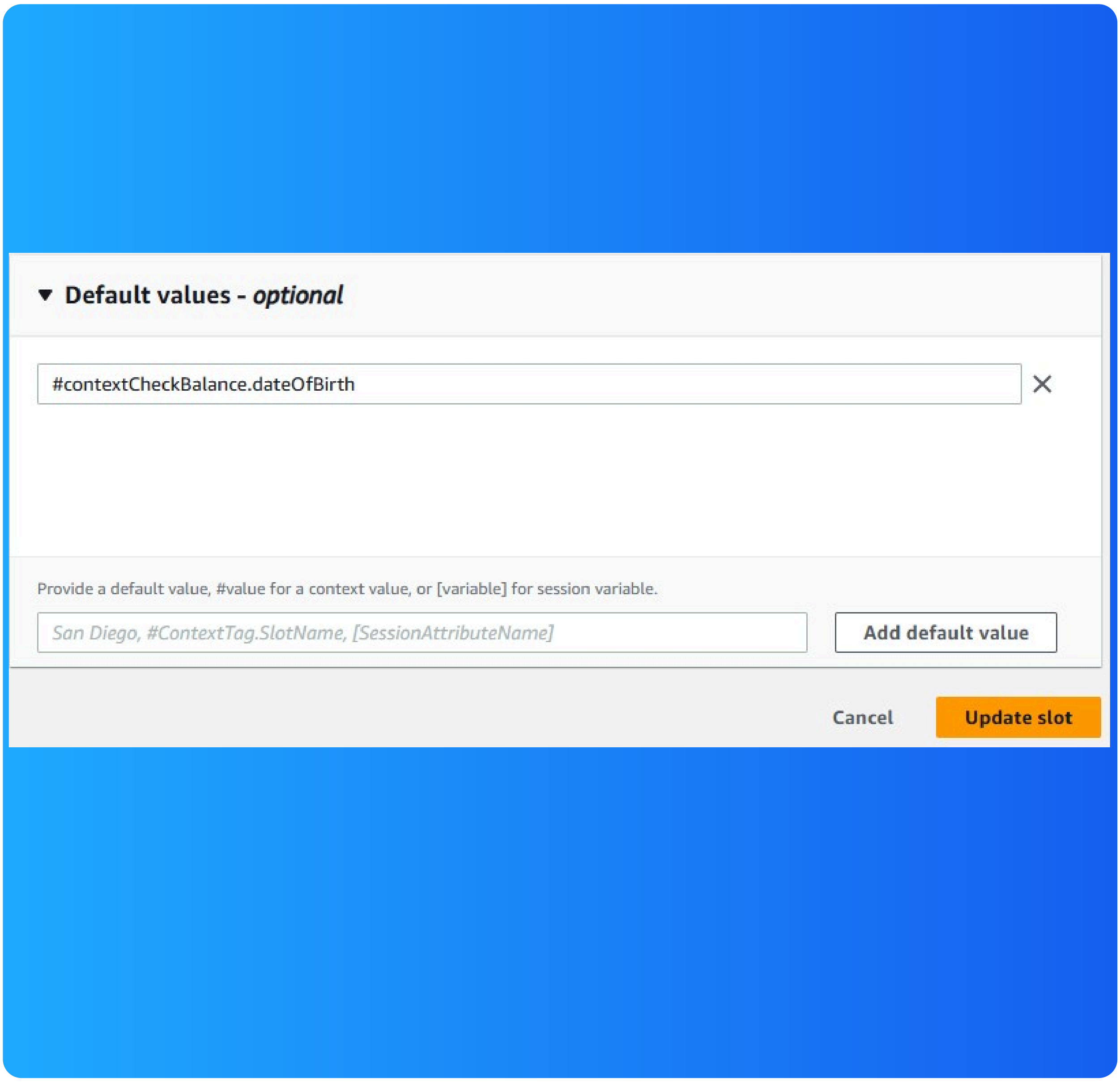
To generate utterances, you must have permissions to Amazon Bedrock. Amazon Lex will make calls to Amazon Bedrock. Additional charges may be incurred based on the usage of Amazon Bedrock. Learn more

```
How about my {accountType} account?
What about {accountType} ?
And in {accountType} ?

```

Gabriel Mazer

Input Context Tag

I created an input context, contextCheckBalance, that carries the dateOfBirth value from CheckBalance intent into FollowupCheckBalance for smooth context transitions.



▼ **Default values - *optional***

#contextCheckBalance.dateOfBirth

Provide a default value, #value for a context value, or [variable] for session variable.

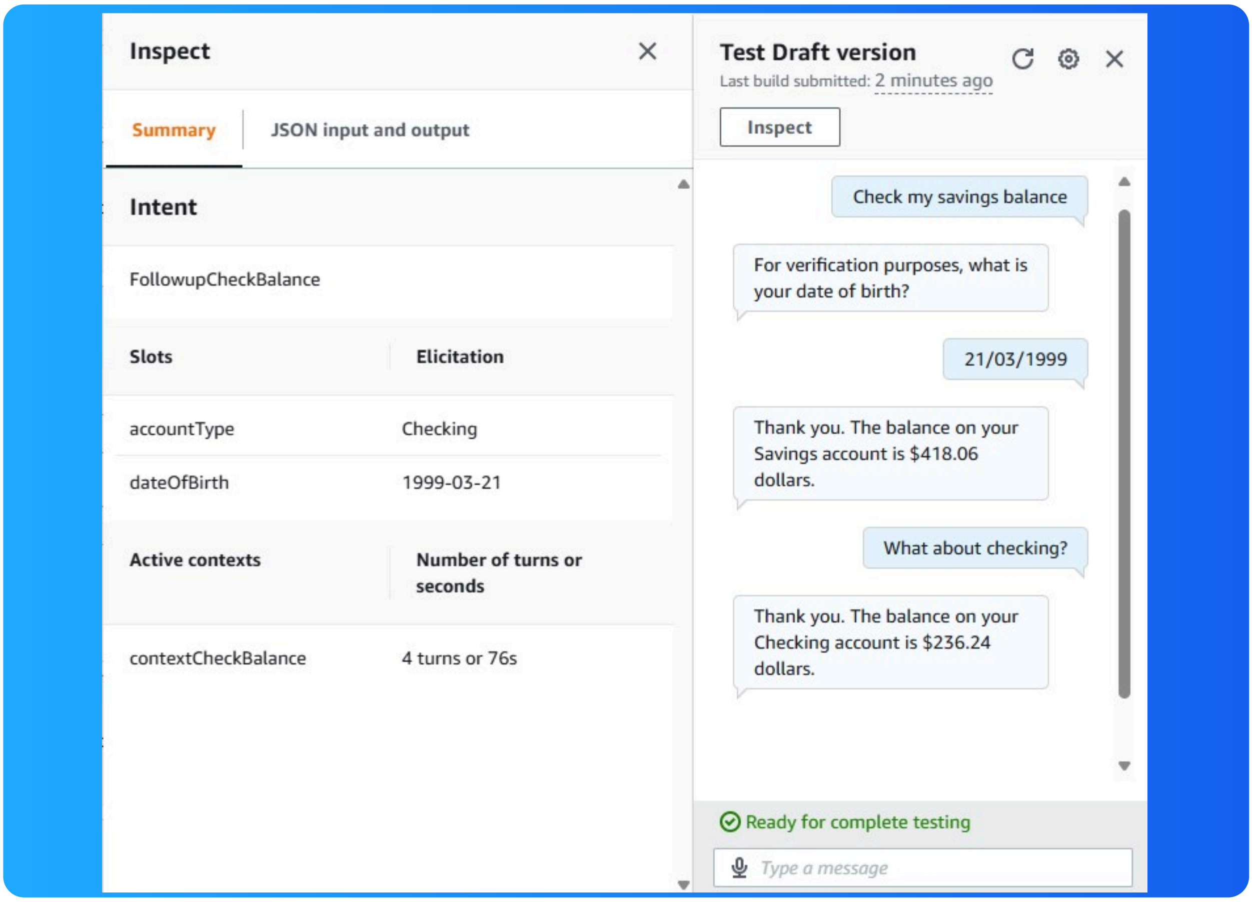
San Diego, #ContextTag.SlotName, [SessionAttributeName]

Gabriel Mazer

The final result!

To see the context tags and the follow-up intent in action, I said, 'What about checking?' without being asked for my date of birth again.

If I had tried to trigger FollowUpCheckBalance without setting context, my chatbot wouldn't have the context needed. As a result, it would trigger FallbackIntent, informing the user that the request isn't understood.



Inspect

Summary | JSON input and output

Intent

FollowupCheckBalance

Slots	Elicitation
accountType	Checking
dateOfBirth	1999-03-21

Active contexts	Number of turns or seconds
contextCheckBalance	4 turns or 76s

Test Draft version

Last build submitted: 2 minutes ago

Check my savings balance

For verification purposes, what is your date of birth?

21/03/1999

Thank you. The balance on your Savings account is $418.06 dollars.

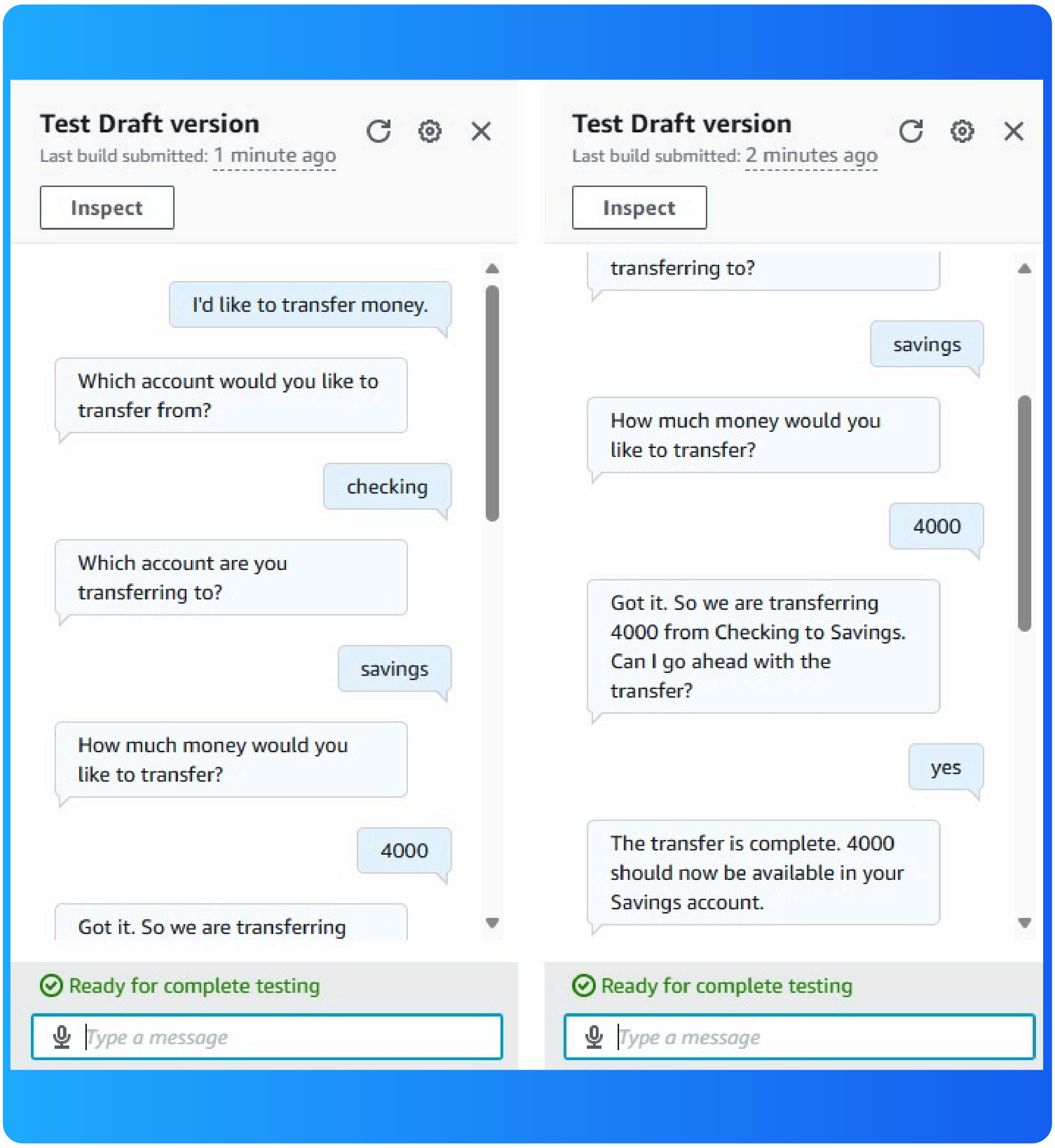
What about checking?

Thank you. The balance on your Checking account is $236.24 dollars.

Ready for complete testing

Build Multiple Slots

Gabriel Taveira Mazer



Test Draft version
Last build submitted: 1 minute ago
Inspect

I'd like to transfer money.

Which account would you like to transfer from?

checking

Which account are you transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring

Ready for complete testing
Type a message
Test Draft version
Last build submitted: 2 minutes ago
Inspect

transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring 4000 from Checking to Savings. Can I go ahead with the transfer?

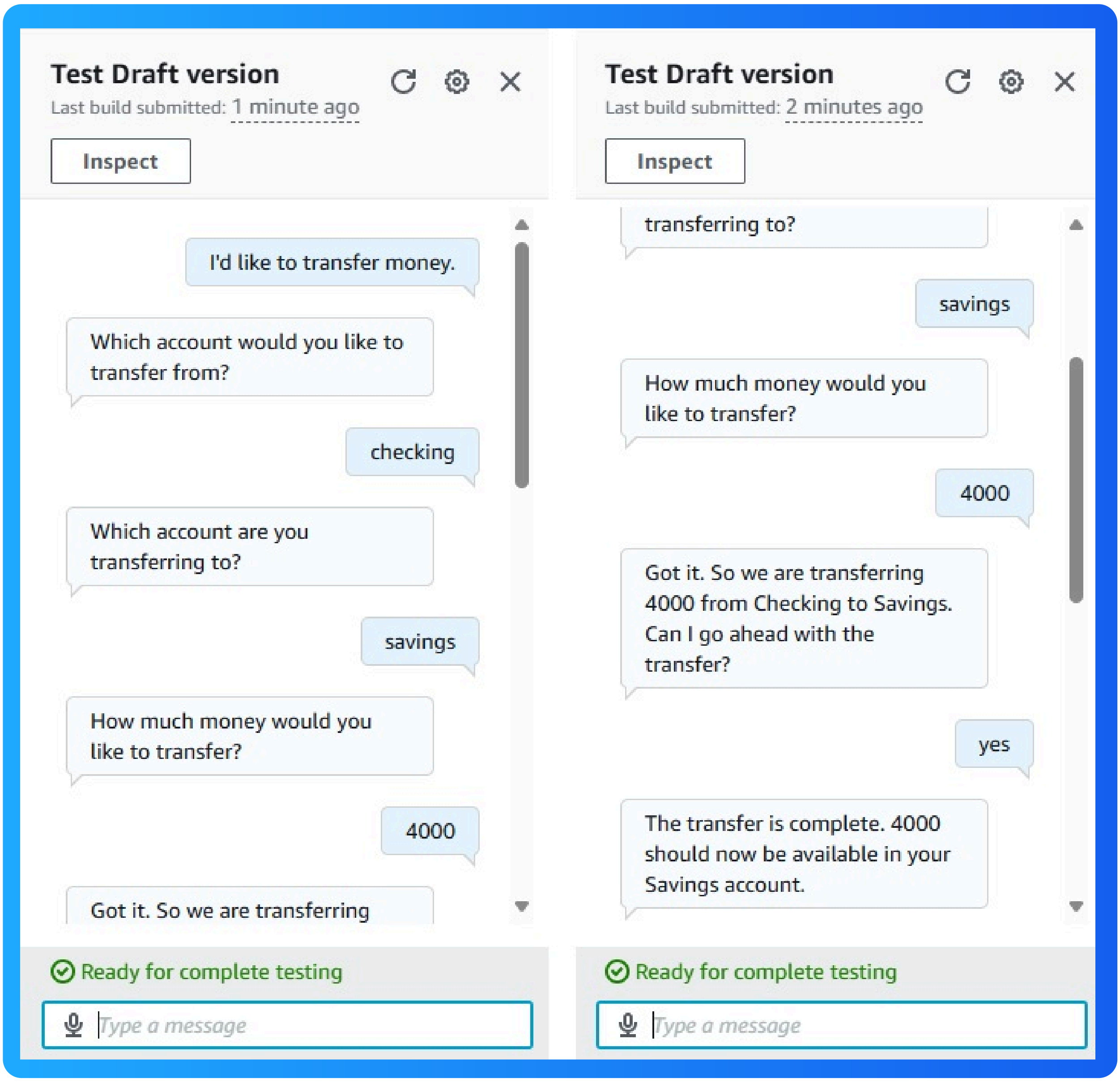
yes

The transfer is complete. 4000 should now be available in your Savings account.

Gabriel Mazer

TransferFunds

An intent I created for my chatbot was TransferFunds, which helps users transfer funds between different bank accounts by specifying source and target accounts and the amount to transfer.


Test Draft version
Last build submitted: 1 minute ago
Inspect
Ready for complete testing
Type a message
Test Draft version
Last build submitted: 2 minutes ago
Inspect
Ready for complete testing
Type a message

I'd like to transfer money.

Which account would you like to transfer from?

checking

Which account are you transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring

transferring to?

savings

How much money would you like to transfer?

4000

Got it. So we are transferring 4000 from Checking to Savings. Can I go ahead with the transfer?

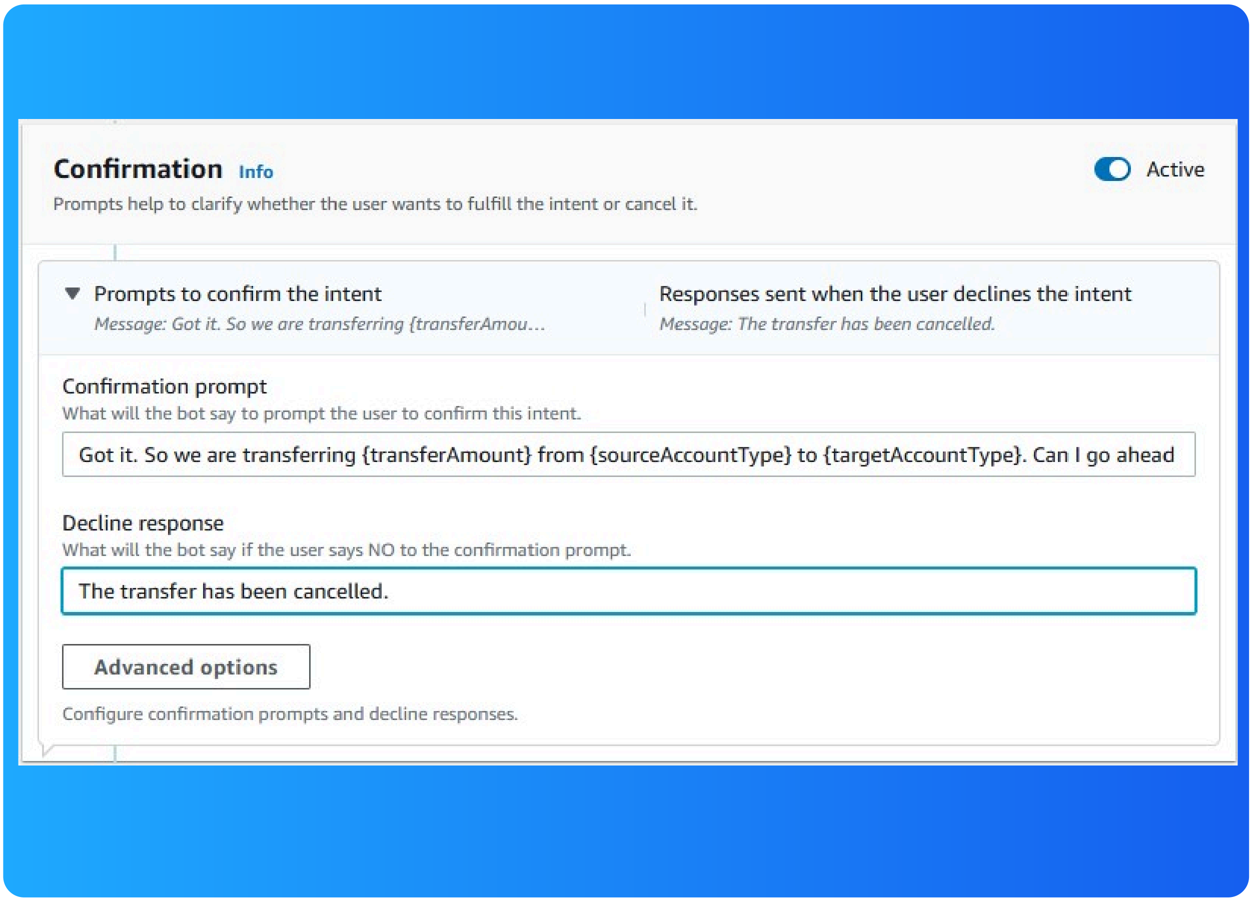
yes

The transfer is complete. 4000 should now be available in your Savings account.

Using multiple slots

For this intent, I had to use the same slot type twice because both the source and target account types share the same accountType slot for selecting accounts.

I also learned how to create confirmation prompts, which are used to confirm user actions before the chatbot processes them, ensuring accuracy.



Confirmation

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 {transferAmou…

Responses sent when the user declines the intent

Message: The transfer has been cancelled.

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.

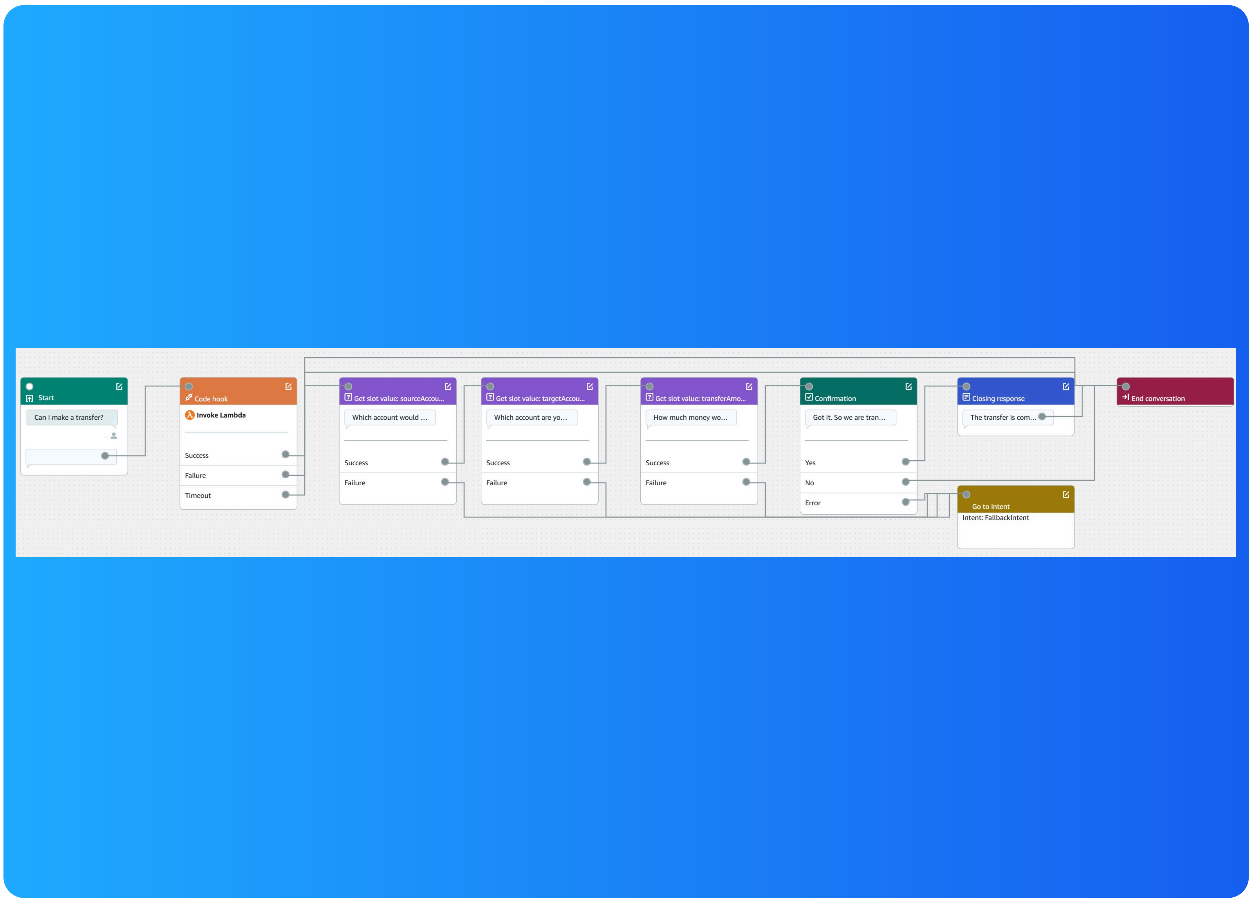
Configure confirmation prompts and decline responses.

Gabriel Mazer

Exploring Lex features

Lex has a conversation flow feature that allows users to define the sequence of prompts, actions, and responses in a clear visual format. This makes it easier to manage the interactions between the bot and the user.

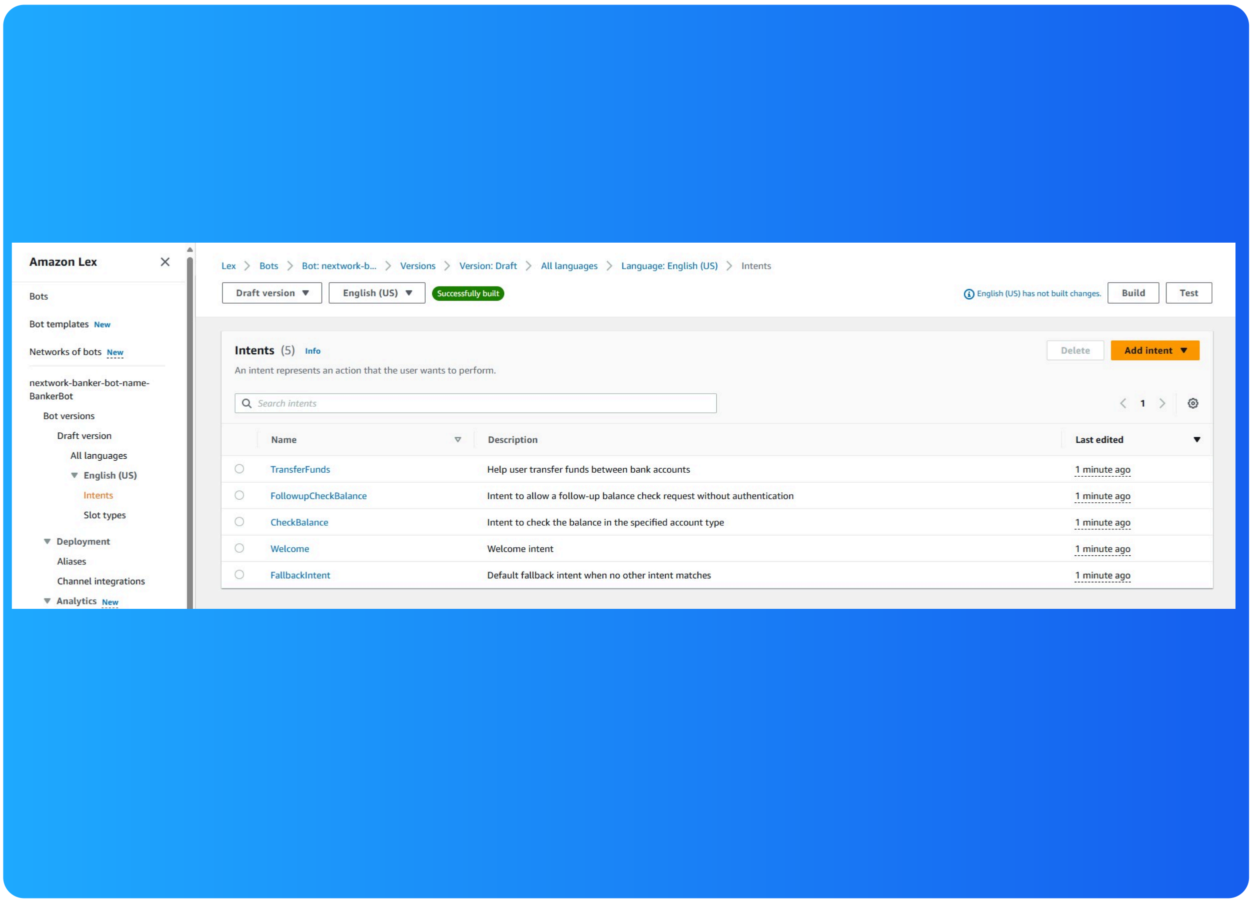
You could also set up your intent using a visual builder! A visual builder offers an intuitive drag-and-drop interface that simplifies creating and organizing different components in your chatbot's workflow.



AWS CloudFormation

AWS CloudFormation is a service that allows you to model and set up your AWS resources using a template. It automates the provisioning and management of resources like Amazon Lex, Lambda, and more.

I used CloudFormation to automate the creation and setup of my chatbot, including the intents, Lambda functions, and roles, saving time and ensuring consistent deployment.



English (US) has not built changes.

Intents (5) Info

An intent represents an action that the user wants to perform.

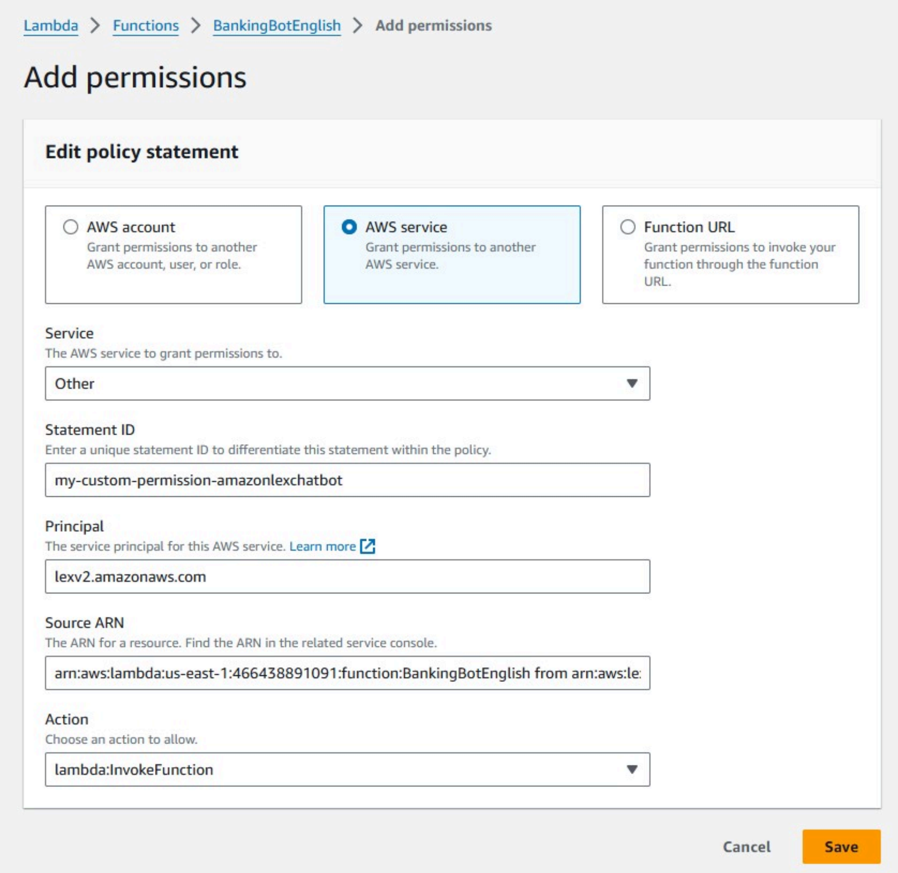
	Name	Description	Last edited
○	TransferFunds	Help user transfer funds between bank accounts	1 minute ago
○	FollowupCheckBalance	Intent to allow a follow-up balance check request without authentication	1 minute ago
○	CheckBalance	Intent to check the balance in the specified account type	1 minute ago
○	Welcome	Welcome intent	1 minute ago
○	FallbackIntent	Default fallback intent when no other intent matches	1 minute ago

Gabriel Mazer

The final result!

Re-building my bot with CloudFormation took me around 15 minutes, as the stack creation process automatically configured all resources.

There was an error after I deployed my bot! The error was an “Access Denied” issue when invoking the Lambda function. I fixed this by adding the necessary permissions for the Lex bot to invoke the Lambda function.



Add permissions

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.
Other

Statement ID
Enter a unique statement ID to differentiate this statement within the policy.
my-custom-permission-amazonlexchatbot

Principal
The service principal for this AWS service. Learn more
lexv2.amazonaws.com

Source ARN
The ARN for a resource. Find the ARN in the related service console.
arn:aws:lambda:us-east-1:466438891091:function:BankingBotEnglish from arn:aws:le

Action
Choose an action to allow.
lambda:InvokeFunction