



AWS Builders' Day

Building Chatbots with Amazon Lex

December 2018

Table of Contents

Overview	3
Create the bot	4
Create a Lambda function	11
Link the bot with the Lambda function	12
Chat with your own bot	14
Conclusion	14

Overview

Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to any developer, enabling you to quickly and easily build sophisticated, natural language, conversational bots (“chatbots”). This lab is designed to demonstrate how to create a new bot including defining intents and slots. This lab will walk you through the following:

- Creating a Lex bot
- Adding intents
- Adding slot types
- Using AWS Lambda as the back-end logic for Lex

Create the bot

1. Log in to the [AWS console](#) and navigate to the Amazon Lex service
2. If you have never created a bot, click “Get Started”
3. Choose “Custom bot”, which will display a dialog asking you to defined your bot
4. Our bot name will be “PersonalBanker”
5. Choose your preferred output voice
6. Session timeout should be 5 minute
7. Choose “No” to the Children’s Online Privacy Protection Act (COPPA) question

The form should now look as follows, noting that we’re going to accept the default IAM role.

Create your Lex bot

Amazon Lex enables any developer to build conversational chatbots quickly and easily. With Amazon Lex, no deep learning expertise is necessary—you just specify the basic conversational flow directly from the console, and then Amazon Lex manages the dialogue and dynamically adjusts the response. To get started, you can choose one of the sample bots provided below or build a new custom bot from scratch.

CREATE YOUR OWN

TRY A SAMPLE

Custom bot

BookTrip

OrderFlowers

ScheduleAppointment

Bot name

PersonalBanker

Language

English (US)

Output voice

Kimberly

Type text here to hear a sample

Session timeout

5

min

IAM role

AWSServiceRoleForLexBots

Automatically created on your behalf

COPPA

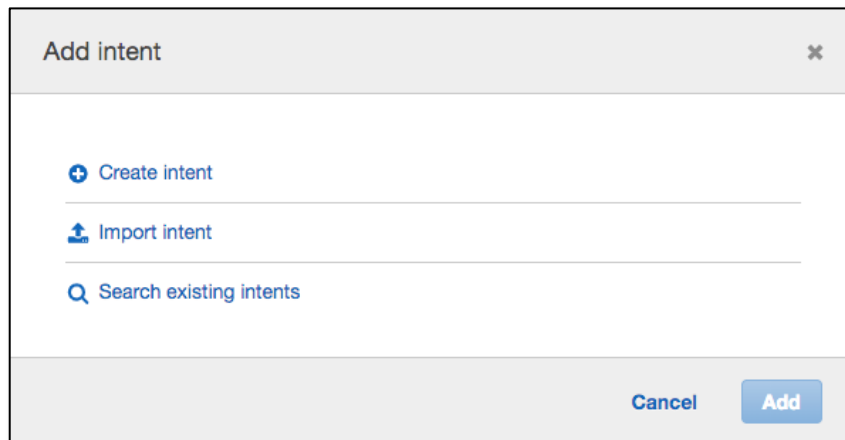
Please indicate if your use of this bot is subject to the [Children's Online Privacy Protection Act \(COPPA\)](#). [Learn more](#)

YesNo

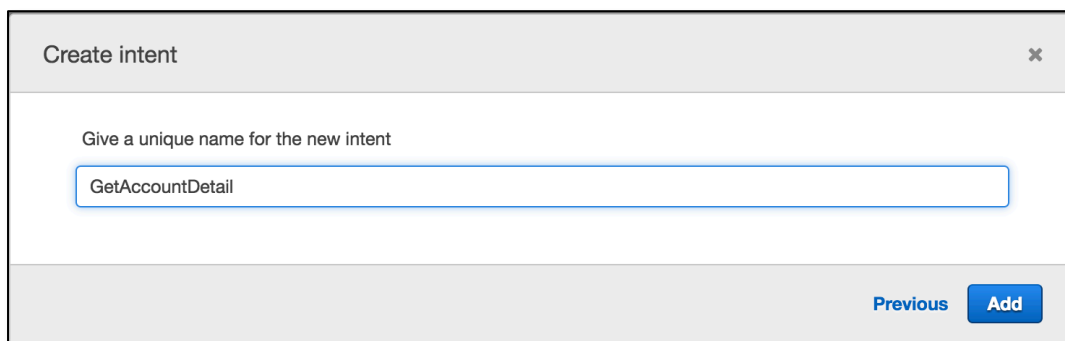
Cancel

Create

8. Click "Create"
9. We will start by creating an intent, which represents an action that the user wants to perform. For example, we're going to create three intents in this lab for different actions a user can perform: Get Account Details; Get Loan Details; Get Loan Products. Click the "Create Intent" button.
10. In the window that pops-up click the "Create intent" link



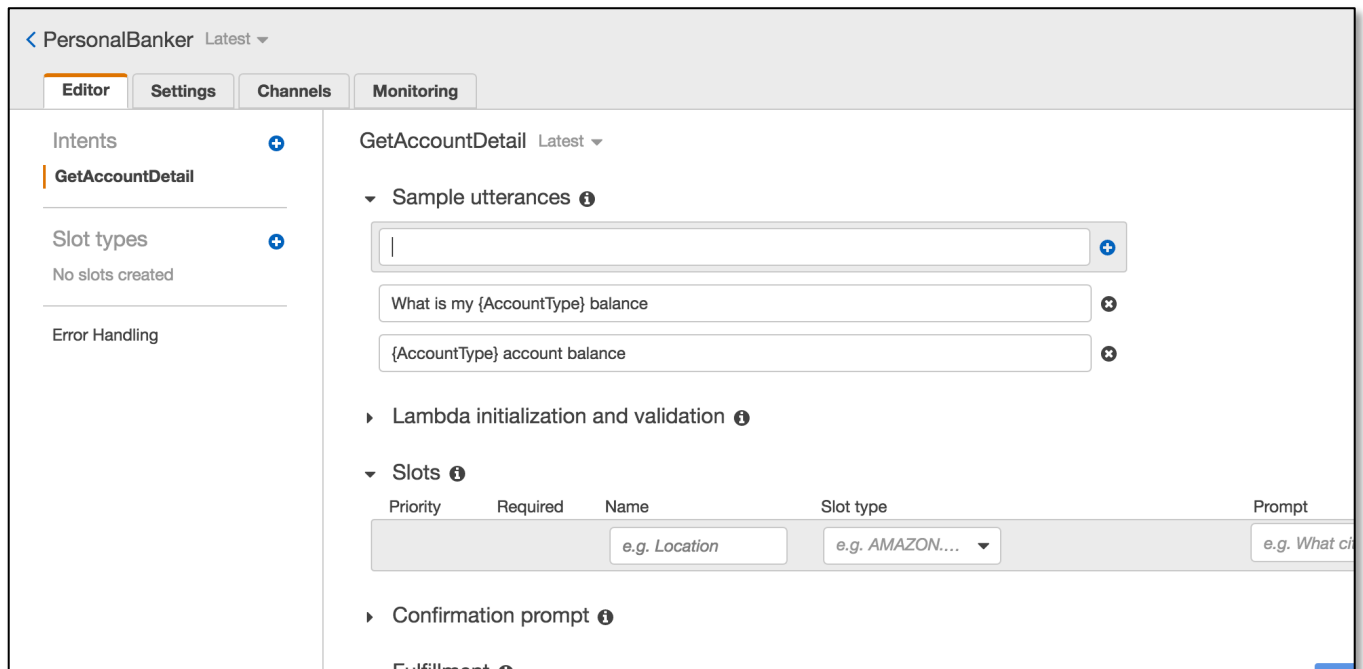
11. Our first intent enables the user to get account details, so name this intent "GetAccountDetail" then click "Add".



12. We now want to provide samples of what our user would type or say to perform this action (i.e. to activate this intent). Under "Sample utterances", type the below phrases and hit [enter] or click the blue "+" sign after each phrase. Make sure you do not add a question mark at the end of the phrase as this will cause build issues later on.
 - *What is my {AccountType} balance*

- `{AccountType} account balance`

NOTE: `{AccountType}` represents a 'slot' which is information that we need to process the users request. Type it exactly as shown above with the braces, and in the next step we will define the 'AccountType' slot and list the acceptable values (checking, savings). Once the lab is complete, the user will interact with our bot by saying something like "What is my checking balance".



13. Next we define a slot which is information we need to process the users request. This information can be included in the utterance (query) that the user types or says, and if not included, Lex will prompt the user for the information. While Lex includes many built-in slot types (such as number, colour, city, food, etc), in this case we want to define a custom slot to get the account type that the user is referring to.

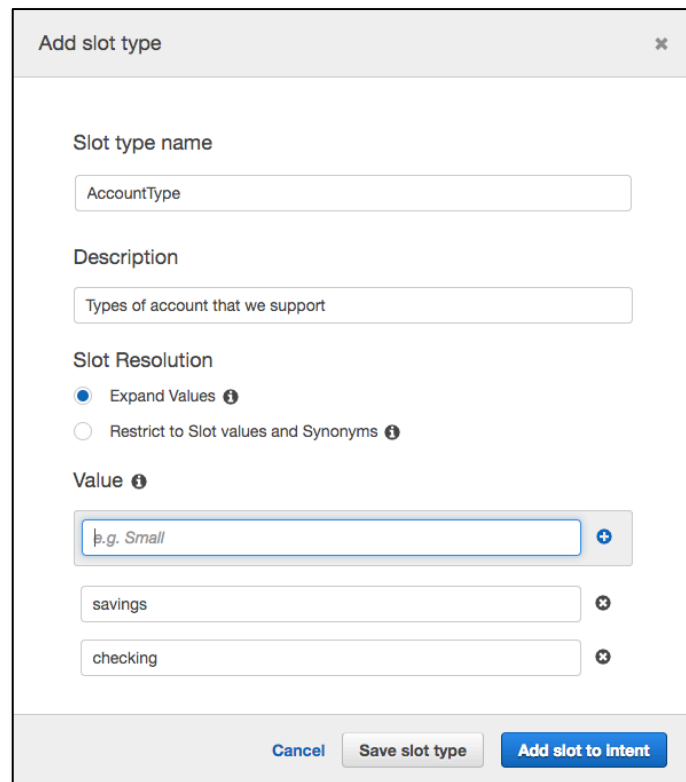
Click on the blue "+" sign next to "Slot types" on the left hand side of the screen and select the "Create slot type" link

AWS Builders' Day - 2018
Building Chatbots with Amazon Lex



14. For 'Slot type name' enter "AccountType" and optionally enter a description (although description is not required)

15. For Value, we want to allow the user to make queries against either their "checking" or "savings" account so enter those as values, clicking the blue "+" sign after each word.



16. Click "Add slot to intent"

17. Now that we have defined what the user is going to ask via an utterance, and also defined our customer "vocabularly" via a slot type, we now have to additionally link the slot types to the

intent with additional information such as whether it is required or not in the prompt to use if Lex has ask the user to provide it.

In the existing Slot list change the “Name” field from “slotOne” to “AccountType” so that it matches the slot name that we specified when we created the sample utterances – note that these do not have to match, but this example will be clearer if we keep them the same.

18. Specify “What type of account?” for the “Prompt” field. This prompt will be used by our bot if the user does not specify an account type when asking a question.

GetAccountDetail Latest ▾

▼ Sample utterances ⓘ

e.g. I would like to book a flight. +

What is my {AccountType} balance ✕

{AccountType} account balance ✕

► Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Prompt
		e.g. Location	e.g. AMAZON.... ▾	e.g. What city? ⚙️ +
1.	<input checked="" type="checkbox"/>	AccountType	AccountType ▾	What type of account? ⚙️ ✕

19. Scroll down and click “Save Intent”

If at any point you made a mistake in the steps above, selecting the “Latest” version of the intent at the top, next to the intent name, will allow you to edit your choices.

20. Let's build this simple Bot: Hit the grey Build button at the top right corner. You will be asked for confirmation to build. Click “Build”.

The build process takes approximately a minute. Once complete, you can ask your bot a question as a way to test it. For example, you could type “What is my checking balance?” in the chat window, or click the microphone symbol, speak your request and client it again to have Lex translate your speech to text. At this stage since we have not added in the backend Lambda function, the response will be that the bot is ready for fulfillment.

Let's add 2 more intents and one more slot type.

21. Click the blue “+” sign next to “Intents” on the left hand side of the page, then click “Create intent”.
22. This intent will allow users to get information on their outstanding home or car loan balance, so set the name to “GetLoanDetail” then click “Add”
23. For sample utterances (the things we expect out users to type/say to our bot to trigger a specific intent/action), add the following phrase then click the blue “+” at the end of the sample utterance box.
 - “Get my {LoanType} loan balance”
24. Now we’ll create a new slot type which we’ll use to store a response from the user as to whether they are wanting the intent to access their *car* loan balance or their *home* loan balance. Click the blue “+” sign next to ‘Slot types’ on the left hand side of the screen
 - For ‘Slot type name’ enter “LoanType” and optionally provide a description
 - Enter the following two options as values:
 - car
 - home

The screenshot shows the 'Add slot type' dialog box in the Amazon Lex console. The dialog has a title bar with 'Add slot type' and a close button. It contains the following fields and options:

- Slot type name:** A text input field containing 'LoanType'.
- Description:** A text input field containing 'The type of loan that we're going to query'.
- Slot Resolution:** Two radio button options:
 - ☒ Expand Values ⓘ
 - ☐ Restrict to Slot values and Synonyms ⓘ
- Value ⓘ:** A list of values with input fields and remove buttons (✕).
 - The first value is 'e.g. Small' in a light blue box, with a '+' button to its right.
 - The second value is 'home'.
 - The third value is 'car'.

At the bottom of the dialog are three buttons: 'Cancel', 'Save slot type', and 'Add slot to intent'.

25. Click “Add slot to intent”

26. Change the name of the slot on the intent detail screen from “slotOne” to “LoanType”

27. Provide a prompt (such as “Which loan account?”)

28. Make sure that the “Required” column is selected

GetLoanDetail Latest ▾

▼ Sample utterances ⓘ

e.g. I would like to book a flight. +

Get my {LoanType} loan balance ✕

► Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt
		e.g. Location	e.g. AMAZON.US_CITY ▾		e.g. What city? ⚙️ +
1.	<input checked="" type="checkbox"/>	LoanType	LoanType ▾	1 ▾	Which loan account? ⚙️ ✕

29. Click ‘Save Intent’

Now we’ll add a final intent to allow users to get more information on the loan products that we have.

30. Click the blue “+” sign next to “Intents” on the left hand side of the page, then click “Create new intent”.

31. Set the name of the intent to “GetLoanProducts” then click ‘Add’

32. For sample utterances (the things we expect our users to type/say to our bot to trigger a specific intent/action), add the following two phrases. Make sure **not** to include punctuation (such as comma’s or question marks).

- “What is the rate on the {LoanType} loan”
- “More loan info”

The “More loan info” utterance will be used as a follow-up after a user has asked for information on their outstanding loan balance. We’re not going to create a new slot type for this intent, as we’re going to re-use the LoanType slot from the intent that we defined previously.

33. As we haven’t defined a new slot type for this intent there isn’t a slot entry yet under on the

intent detail panel. Enter the following values into the empty slot line:

- Name: LoanType
- Slot Type: LoanType (select from the drop-down list)
- Prompt: Which loan type?

34. Click the blue “+” button to the right of the Slots information to add this slot to the intent

GetLoanProducts Latest ▾

▼ Sample utterances ⓘ

e.g. I would like to book a flight. +

More loan info ✕

What is the rate on the **LoanType** loan ✕

▶ Lambda initialization and validation ⓘ

▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt		
		e.g. Location	e.g. AMAZON.US_CITY ▾		e.g. What city?	⚙️	+
1.	<input checked="" type="checkbox"/>	LoanType	LoanType ▾	1 ▾	Which loan type?	⚙️	✕

35. Click 'Save Intent'

36. Click Build, and click Build again when prompted for confirmation.

Our bot is almost ready, and you can test it like before with the new utterances, but it is still quite dumb – all it needs now is a smart backend.

Create a Lambda function

Here we will create a Lambda function that has some Python code to detect the intent name ('GetAccountDetail', 'GetLoanDetail' or 'GetLoanProducts') and to return static values based on the AccountType (checking, saving) or LoanType (car, home) included in the intent. In a real world example we would have already authenticated the user and would write Python code to do a database lookup for the account balances.

1. Use the AWS Console to navigate to Lambda.
2. Click on the orange 'Create a function' link under the 'Getting Started' section
3. On the "Create function" page, click the "Author from scratch" button
4. Let's give our function the name of "myPersonalBanker" and optionally provide a

description

5. Choose Python 2.7 as the Runtime
6. We will “Create new role from template – give it a Lex-style role name (such as “LexRole”) and select “Test Harness permissions” as the policy template.

Author from scratch [Info](#)

Name
myPersonalBanker

Runtime
You can select a supported AWS Lambda runtime or provide your own runtime as part of the function deployment package or Lambda layer after creating the function.
Python 2.7

Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.
Create a new role from one or more templates.

Lambda automatically creates a role with permissions from the selected policy templates. Basic Lambda permissions (such as logging to Amazon CloudWatch) are automatically added. If your function accesses a VPC, the required permissions are also added.

Role name
Enter a name for your new role.
LexRole

This new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Policy templates
Choose one or more policy templates. A role will be generated for you before your function is created. [Learn more](#) about the permissions that each policy template will add to your role.
Test harness permissions X

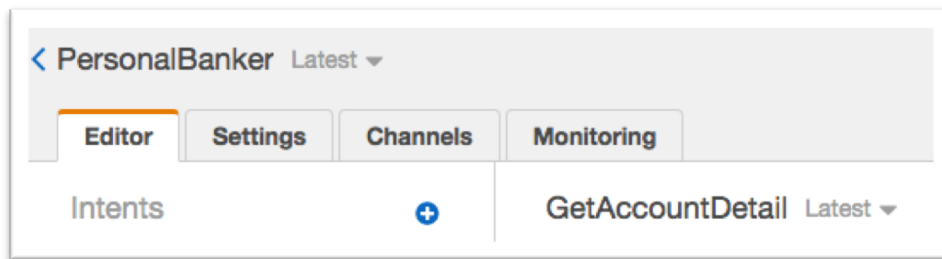
7. Hit “Create function” on the bottom right and you’ll be take to the “Configuration” window. We are not adding any additional triggers, nor are we using Lambda Layers, so scroll down to the “Function code” section
8. Open the lambda function code you will find [here](#) (myPersonalBanker_v1.py). Copy and paste the code into the inline editor – make sure that you overwrite any template code that is already in the code box
9. Scroll down to the “Execution role” section and ensure that the role you created previously is selected in the “Existing role” drop-down – if not then please select it
10. Leave the rest unchanged, then hit the orange “Save” button at the top of the screen

Link the bot with the Lambda function

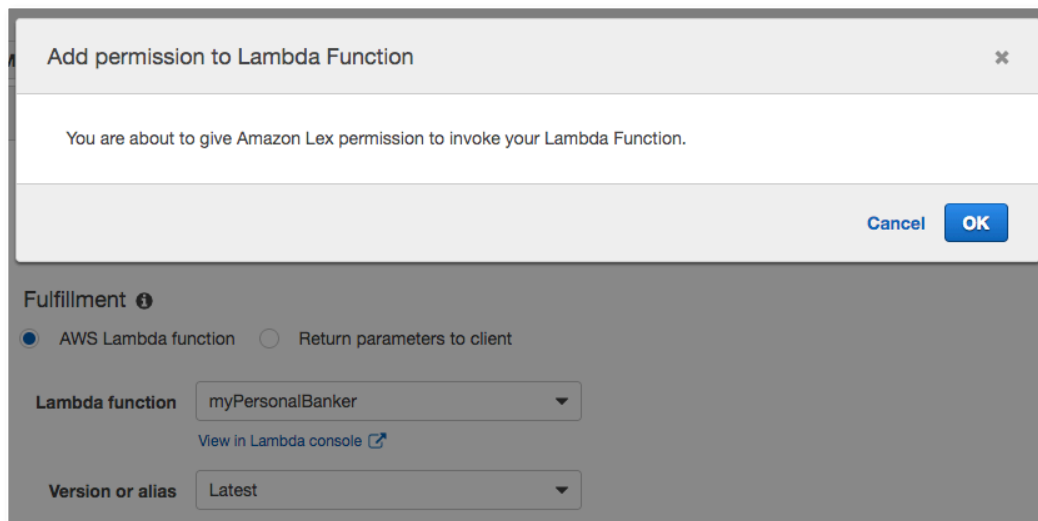
In this step we will link the three intents we created to the Lambda function. We do this by providing the Lambda function as the method that contains the business logic used to 'fulfill' the users requests. Once this is done (and the bot rebuilt), when a user specifies an intent (such as 'what is my checking account balance'), Lex will call our Lambda function and pass it the intent name ('GetAccountDetail') and the slot value ('checking').

To do this, we go back to the [Lex Console](#).

1. Click on Personal Banker
2. Ensure the 'GetAccountDetail' intent is selected
3. Make sure that the 'Latest' version is selected for both the bot and the intent



4. Scroll down to "Fulfillment", select "AWS Lambda function", choose "myPersonalBanker" and click "OK" in the popup warning window which opens. It indicates you are giving Lex the permission to run this Lambda function.



5. Click "Save intent"

6. Repeat the above steps **3, 4 and 5** for intents “GetLoanDetail” and “GetLoanProducts”
7. Click “Build” and then click “Build” again on the confirmation screen.

It is time we give this bot a run!

Chat with your own bot

1. Start by typing “What is my checking account balance?” (or press the microphone button and ask your question using your computer mic).
2. You should get an answer.
3. Then type “What is my home loan balance?” (or ask using your mic)
 - Notice that Lex is able to recognize that you are wanting to trigger the GetLoanDetail intent even though what you typed, “**What is** my home loan balance?”, did not exactly match the sample utterance that you configured the intent with which was “**Get my** {LoanType} loan balance”.
4. Type ‘more loan info’ and see how Lex returns information on the current, new home loan rate. In this case, because we didn’t set the ‘slot’ to be required, we didn’t need to specify whether we were looking for more information on *car* or *home* loans ... Lex returned information on the loan type (in this case, home) that we had just asked about.

Conclusion

In this lab you have learned the basic operations to manage a Lex bot. First, you created a bot, then you defined intents and slot types. Finally you defined a Lambda function and attached it to your chatbot.