**language translation bot using Amazon Lex**

**4.2A Create an empty chatbot**

**Create an empty chatbot**

1. Login to your **AWS management console** and navigate to **Amazon Lex** from the search bar.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
2. Click on **‘Create Bot’.**
3. In the Creation method, go with **‘Create a blank bot’.**
4. Give the chatbot a suitable name and description.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
5. For the IAM role, choose **‘Create a role with basic Amazon Lex permissions’** which will automatically create a role for you.
6. Choose **‘No**’ for the option Children’s Online Privacy Protection Act (COPPA), leave the rest as default and click Next.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
7. You can choose the language you want to communicate with the bot and can also choose between different voice interaction modules.
8. The **Intent classification confidence score threshold** is like a confidence bar that the system uses to decide if it's sure enough about what the user meant. For now let’s keep it default and click Done.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
9. Your empty chatbot has been created! You will lead on the Intent page where we will work on the conversational flow of the chatbot.

**4.2B Specify Intents and slots**

**Specify Intent and Slots**

A diagram of a structure

AI-generated content may be incorrect.

1. Let’s first understand some terms used in Amazon Lex:- Intent: The goal or purpose behind what the user is saying.  
   - Utterance: The actual words or phrases spoken or typed by the user.  
   - Slots: Specific pieces of information or variables within the user's utterance that the system needs to extract.- Fulfilment: The action or response that the system takes based on the user's intent and the information provided in the slots.  
     
   Take a look at this example:  
     
   In a pizza ordering chatbot, the intent reflects what the user wants, like ordering pizza or checking an order. Utterances are what users say, such as "I want a large pepperoni pizza." Slots are specific details in these utterances, like size and toppings. Fulfillment is the chatbot's response, confirming the order and providing information.
2. Now that we are familiar with the basic terms, let’s start with building our chatbot.
3. In the Intent details, fill in an appropriate Intent name and description.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
4. In the **sample utterances,** add some inputs that can be asked by the user. Amazon Lex will try to understand the user input by aligning the input with the utterances provided.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.  
     
   We will add some more utterances later on after creation of slots.
5. Go to the slots option and click on **Add Slot.** We need to take the language in which the text to be translated as a slot. For that we need to first create the Slot type. Go back and click on **Save Intent**.
6. Click on back to Intents list and navigate to slot type, **add Slot type**.
7. Choose **‘Add blank slot type’** and give a name to the slot type (here **‘language’**).  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
8. Add some languages as Slot type values similar to given below:  
     
   A screenshot of a computer

   AI-generated content may be incorrect.  
     
   (Let us go with the same slot type values for now instead of adding new ones as there would be modifications required in the Lambda function too)
9. Click on **Save slot type.**
10. Navigate back to the Intents to use this slot type for our slot.  
      
    A screenshot of a computer

    AI-generated content may be incorrect.
11. Click on **Add Slot**. Give the slot name and choose the slot type ‘**language’** previously made. Write the prompt where the chatbot asks for user choice for language translation.
12. Click on **Add.**
13. Create another slot **‘text’** which takes the text to be translated as an input. Choose *AMAZON.FreeFormInput* as the slot type option and enter a suitable prompt asking for the text to be translated. Click on **Add.**  
      
    A screenshot of a computer

    AI-generated content may be incorrect.
14. We can add some more utterances specifying the Slots.  
     **For example :** Instead of ‘I want to translate’ , if user inputs ‘I want to translate to French’ with the language specified in the starting intent itself, it should not ask for the language again by running the language prompt. Instead we can specify Slots like below:  
      
    A screenshot of a computer

    AI-generated content may be incorrect.  
      
    This will automatically understand the language slot type if already specified and ask for the input text to be translated directly.
15. We can also add an **Initial response** to the initial intent as a feedback message.  
      
    A screenshot of a computer

    AI-generated content may be incorrect.

**4.2C Specify Fulfilment**

**Specify Fulfilment**

1. Fulfilment runs the Lambda function to fulfil the intent and informs users about it’s status once it is complete.
2. Write a suitable prompt on successful fulfilment and in case of a failure.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
3. Click on **‘Advanced Options’** and check **‘Use a Lambda function for fulfilment’.** Click on **update options.**
4. We can also provide a **Closing message** after completion of the intent.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
5. Click on ‘Save Intent’.

We have setup the conversation flow of our chatbot. Next, we will proceed to creating the Lambda function for actual serverless text translation.

**4.2D Create an IAM role**

**Create an IAM role**

1. From your **AWS management console**, navigate to **IAM** from your search bar.
2. Navigate to roles and **Create Role**. This role will be used for the Lambda function to provide basic Lambda execution permissions and access to Amazon Translate.
3. Select the **trusted entity type** as **AWS service** and select **Lambda** as the use case. Click on **Next.**  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
4. Attach the following policies to the role :
   1. *TranslateFullAccess*
   2. *AWSLambdaBasicExecutionRole*
5. Click on **Next**. Enter a suitable name and description for the role and **Create Role.**  
     
   A screenshot of a computer

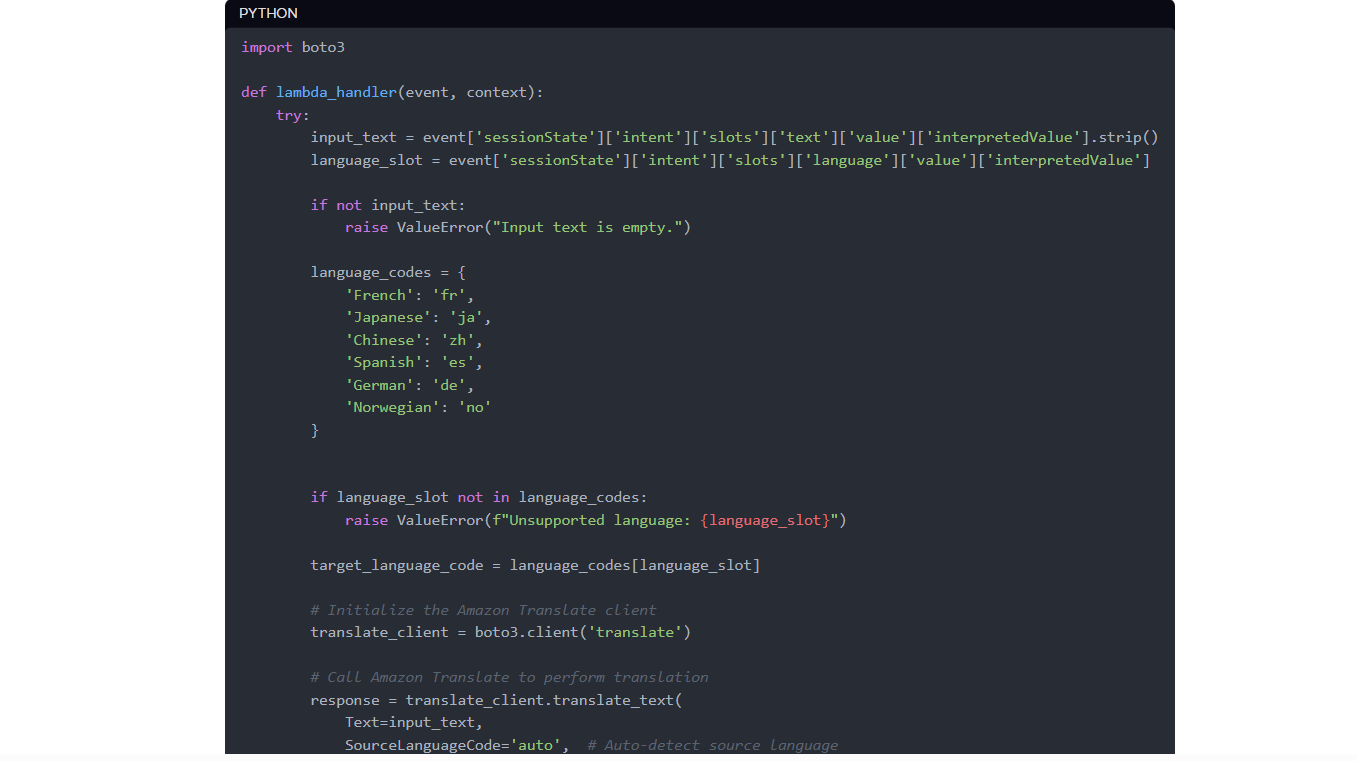
   AI-generated content may be incorrect.
6. This role will be used for the Lambda function permission.

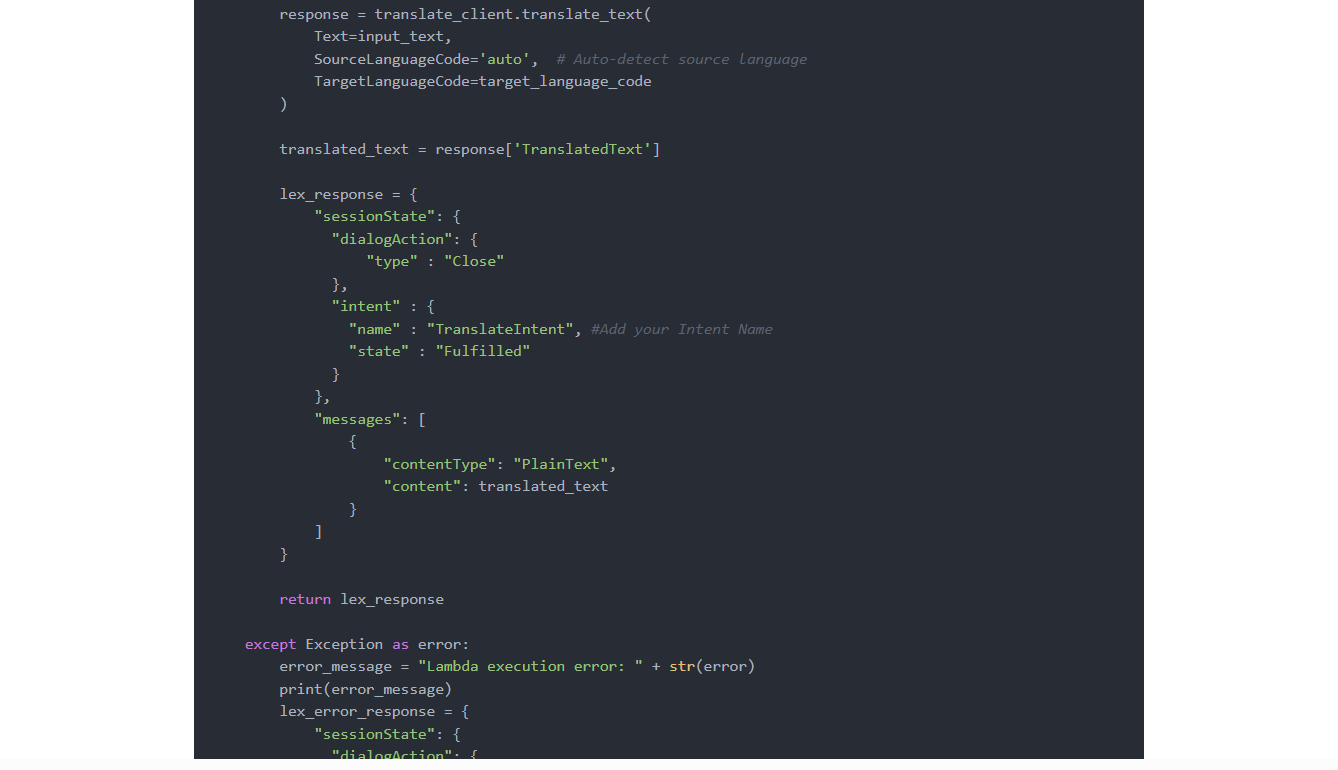
**4.2E Create a Lambda function**

**Creating a Lambda function**

1. From your AWS management console, search for Lambda from your search bar.  
     
   A screenshot of a computer

   AI-generated content may be incorrect.  
     
   2. Click on **Create Function.** Give the function a suitable name and choose the runtime as **Python 3.12.**  
     
   3. Choose the previously created role by toggling the **default execution role** option. Keep the rest of the values as default and **‘Create Function’.**  
     
   4. Let us first look at the **Lambda code** and we will understand the code below:

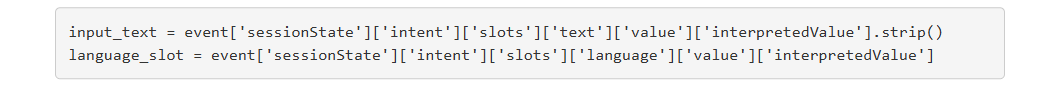




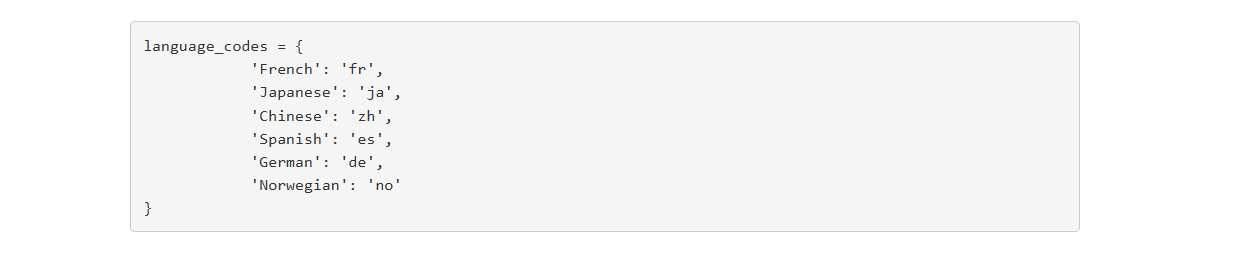


a)**boto3** is used to interact with AWS services through code.

b)**Get Input**:

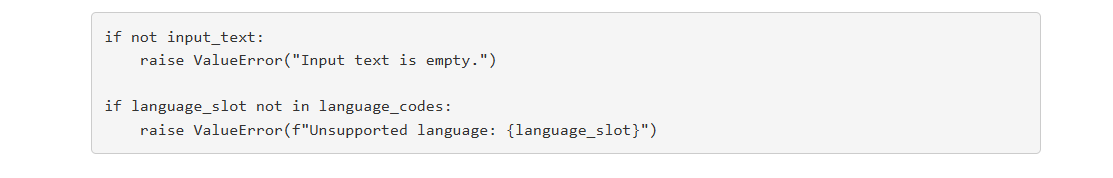
  
Here, the code extracts the user's input text and the target language from the Lex event. If you have kept different slot names from the ones followed in this documentation, remember to change them accordingly.

c) **Language codes:**

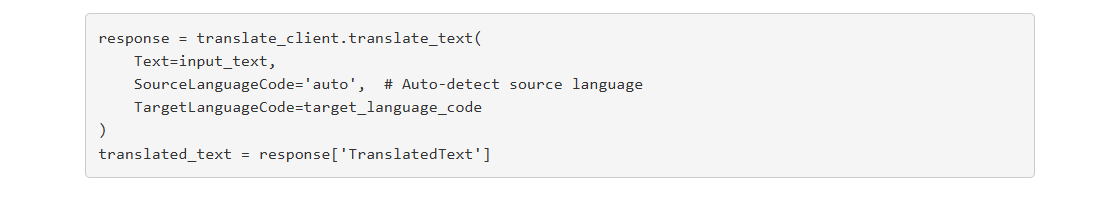
  
These are the language codes format used by Amazon Translate to identify the language to be used. We need to pass this language\_code in the translate\_text function to translate the input according to the mapped value of the language code.

You can further add more languages as you prefer, however remember to also add them in the slot type language.

d) **Check Input**:

  
These lines ensure that the input text is not empty and that the target language is supported.

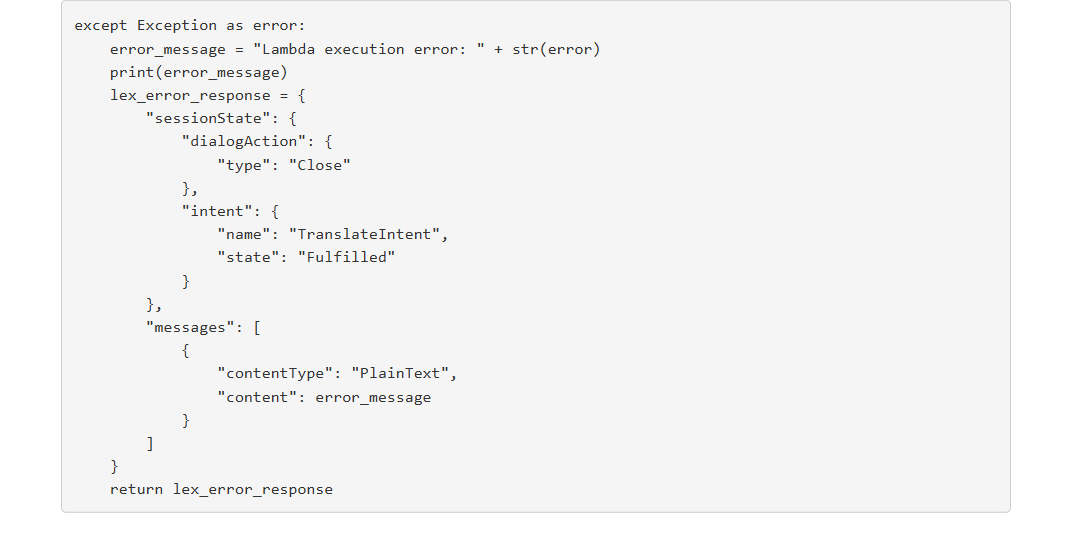
e) **Translate**:

  
This section uses Amazon Translate to translate the input text into the target language specified by the user.

f) **Prepare Response**:

  
After translation, the code constructs a response for the chatbot with the translated text.

g)**Handle Errors**:

  
This part catches any errors that occur during the process and sends an appropriate error message back to the chatbot.

1. Click on **‘Deploy’** to deploy the Lambda function.

**4.2F Test the Lambda function**

**Testing the Lambda function**

1. To test the Lambda function, click on **Test** and configure the **Test event.**

2. Give a suitable name to the test event and keep the Event JSON in the format below:

A screenshot of a computer

AI-generated content may be incorrect.



**Testing the Lambda function**

The reason that we are providing this format is because your Lambda function gets an input from the Amazon Lex in this specified format.

3. Save the test event and click on Test again to Invoke the event.

A screenshot of a computer

AI-generated content may be incorrect.

The function gives an output in a JSON format which is the required format to give an input for Amazon Lex. It will get the message content and display it to the user.

Next, let’s test out the chatbot working.

**4.2G Test the chatbot**

**Testing the chatbot**

1. Navigate to the **Intent page** of your previously created Chatbot. Before testing it, we need to specify the Lambda function to be used by Lex.
2. Click on the **settings** option present on your top right of the chatbot window and choose the **Source of Lambda function** as your previous created function.
3. Now you are ready to go! Test out the chatbot, here are some conversations:

A screenshot of a chat

AI-generated content may be incorrect.. A screenshot of a chat

AI-generated content may be incorrect.