AI 620 Emerging Topics in Artificial Intelligence

HOS05A Contact Organizer Application

02/20/2023 Developed by Yared Shewarade
09/18/2024 Reviewed by Anh Nguyen
10/25/ Reviewed by Jonathan Koerber

School of Technology and Computing (STC) @City University of Seattle (CityU)

Before You Start

- The directory path shown in screenshots may be different from yours.
- Some steps are not explained in the tutorial**.** If you are not sure what to do:
    1. Consult the resources listed below.
    2. If you cannot solve the problem after a few tries, ask the courses student worker for help.

Learning Outcomes

Students will be able to learn:

- Introduction to Amazon Lex
- Setting up Contact Assistant Architecture

Resources

- Tripuraneni, S., & Song, C. (2019). *Hands-on artificial intelligence on amazon web services: Decrease the time to market for AI and ML applications with the power of AWS* (1st ed.). Packt.

Introduction to Amazon Lex

Amazon Lex is a development platform for building intelligent assistants or chatbots. With Amazon Lex, we are building our own custom intelligent assistant capabilities. Lex itself provides many AI capabilities, including Automatic Speech Recognition (ASR) and Natural Language

Understanding (NLU), that are useful for building conversational interfaces. However, developers must follow Lex's development constructs, conventions, and norms to leverage these underlying AI capabilities.

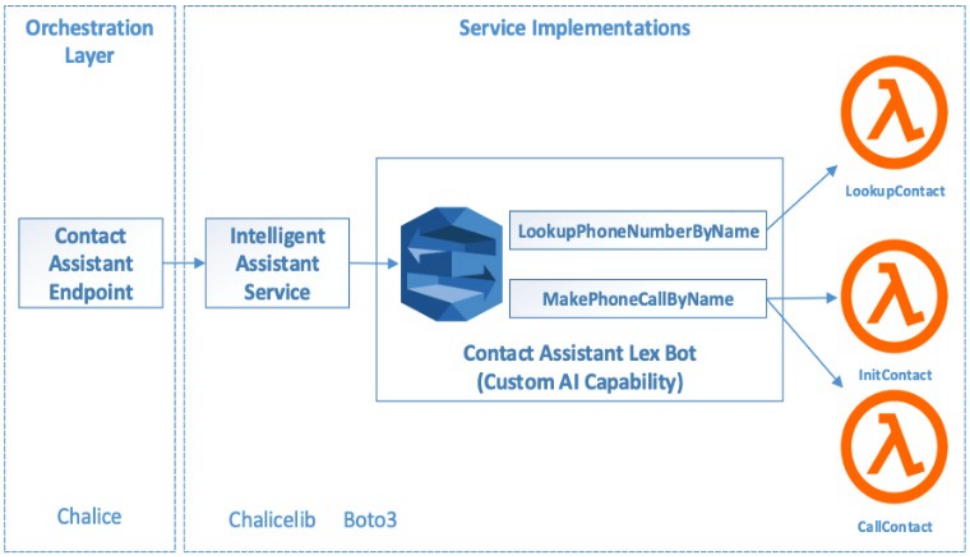
These Amazon Lex conversational interfaces are built from Lex’s specific building blocks:

- **Bot**: A Lex boot can perform a set of related tasks through the custom conversational interfaces. A bot organizes the related tasks into a unit for development, deployment, and execution.
- **Intent**: An intent represents an automated task the users want to perform. AN intent belongs to a specific AWS account rather than specific bot and can be used by different bots in the same AWS account.
- **Sample utterance**: An utterance is a typed or spoken phrase in natural language that the user might say to invoke an automated task. Amazon Lex encourages developers to provide multiple utterances to make the conversational interfaces more flexible for the users.
- **Slot type**: Each slot has a type, and it restricts input space and simplifies verification to make the conversational interface more users friendly
- **Prompt and response**: A prompt is a question in which Lex asks the users to either provide input to a slot, or to confirm the input provided.
- **Session attributes**: Amazon Lex provides mechanisms to keep contextual data that can be shared across intents in the same session data that can be shared across intents in the same session.

The architecture for the contact assistant project includes the following:

- An organization layer
- A service implementation layer

The contact assistant bot will be able to perform two tasks, `LookupPhoneNumberByName` and `MakePhoneCallByName`. This bot leverages Amazon Lex's underlying AI capabilities to interpret the user's verbal commands, and then performs the tasks using AWS Lambda functions. These Lambda functions implement the fulfillment of the tasks, looking up phone numbers and making phone calls.



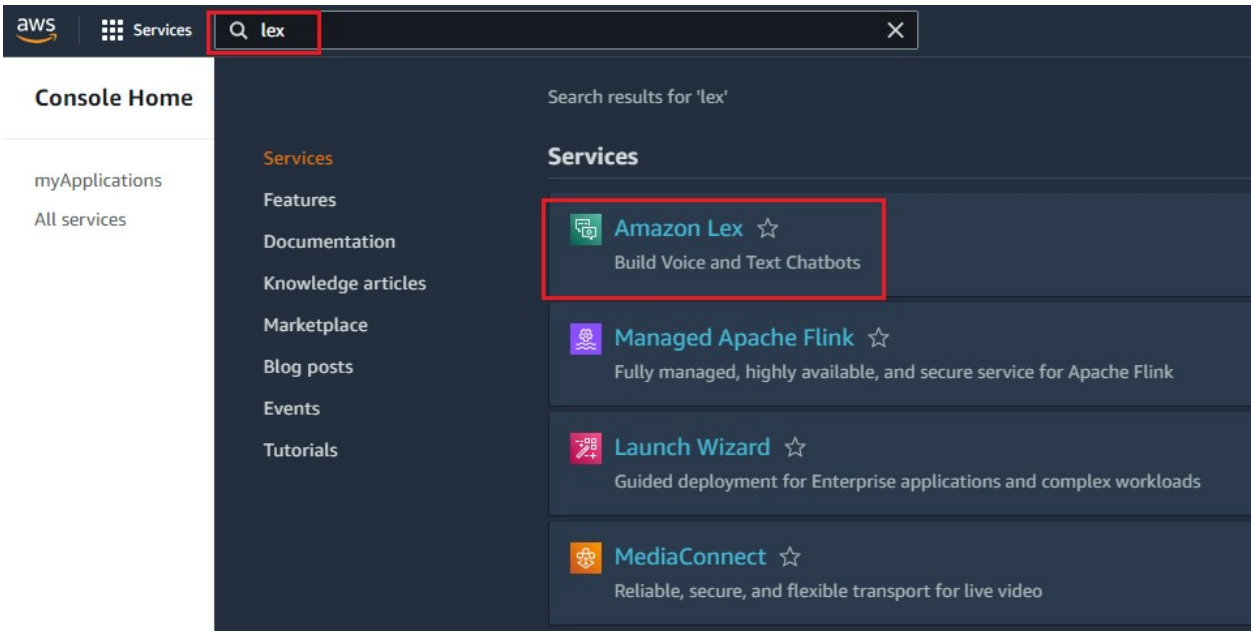
The contact assistant architecture includes the following:

- In the orchestration layer, we will build a **Contact Assistant Endpoint** that provides a RESTful interface to access our contact assistant's capabilities.
- In the service implementation layer, we will build a service, called the **intelligent assistant service**, that shields implementation details of our custom AI capability, including its Amazon Lex implementation details. This way, when we want to reimplement the contact assistant bot with a different chatbot technology, only the intelligent assistant service needs to be modified.

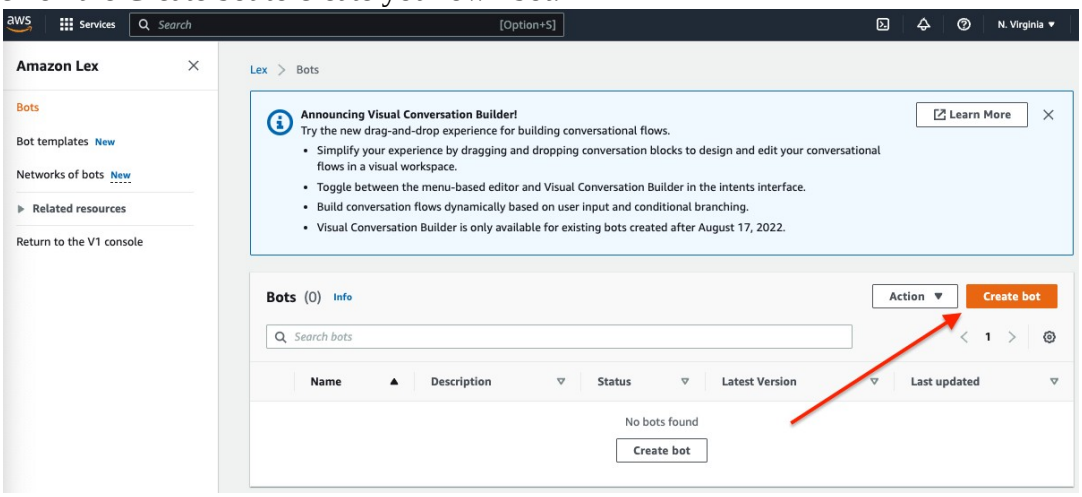
Setting up the contact assistant bot

Note: For submission, take the screenshot for all steps and save it in your local repository along with your code.

1. Sign in to your AWS Console. In the search box, type "lex" and navigate to the Amazon Lex under Services.



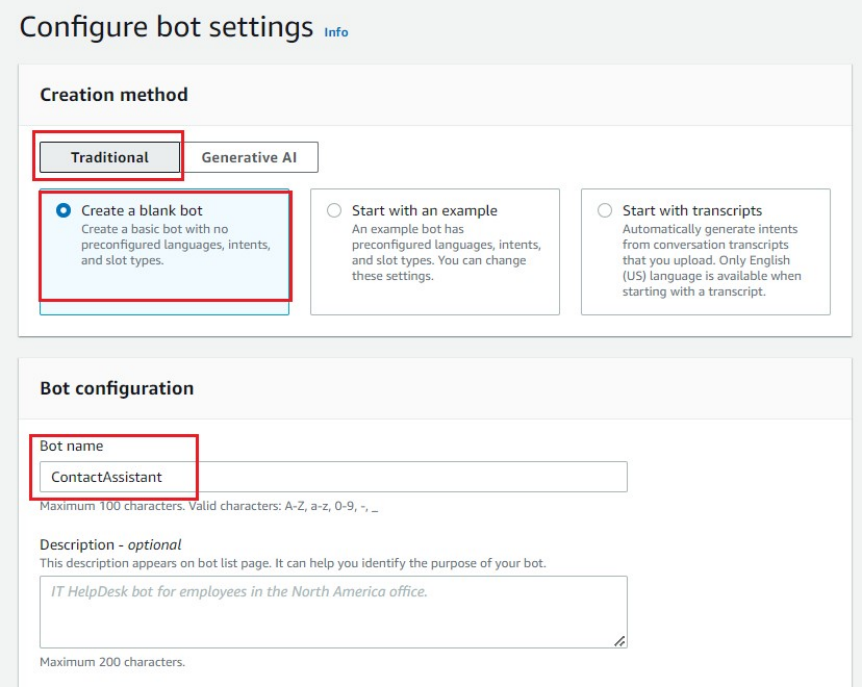
2. Click on the Create bot to create your own bot.

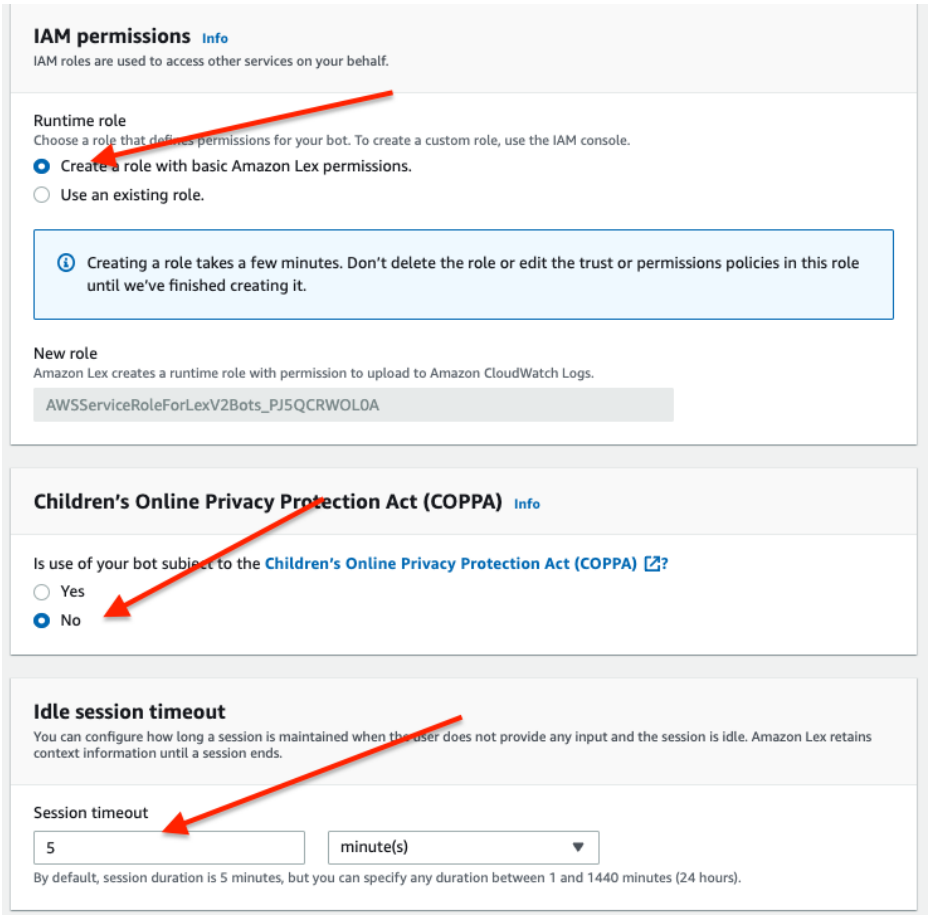


3. Enter the following information:

- Creation method: Traditional > Create a blank bot
- Bot name: ContactAssistant
- IAM permission: Create a role with basic Amazon Lex permissions
- Children's Online Privacy Protection Act (COPPA): No
- Idle session timeout: 5 minute(s).

Then select the **Next** button.

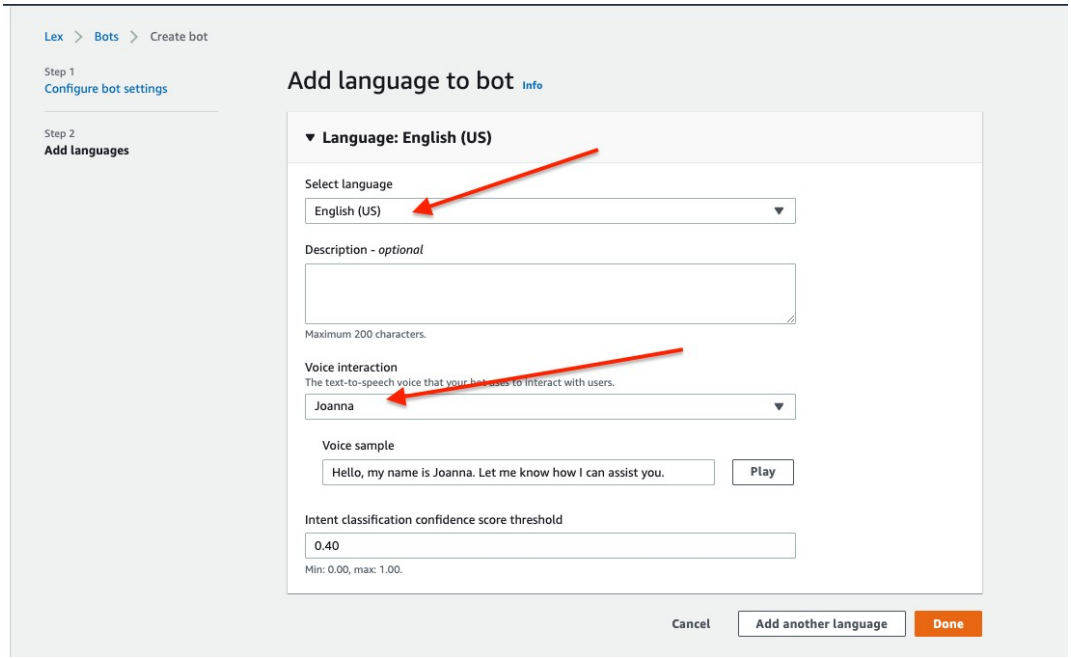




4. In the Add language to bot section:

- Select languange: English (US)
- Voice interaction: Joanna. Currently, Lex only supports US English

Then select **Done**.



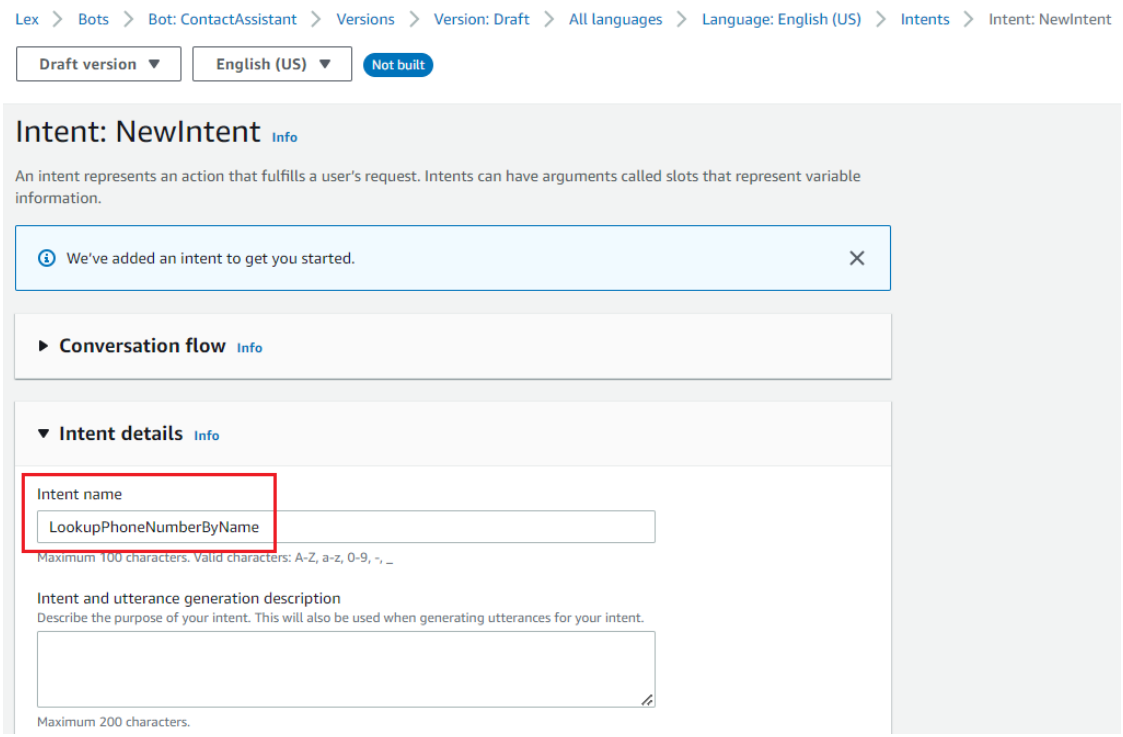
Once the bot is created successfully, you will create an intent for the bot. Follow the next section to create a new intent.

LookupPhoneNumberByName intent

Our first intent allows the user to look up a contact phone number by stating the contact's first and last names. This intent is essentially a search feature built on top of the contact store, but with a conversational interface.

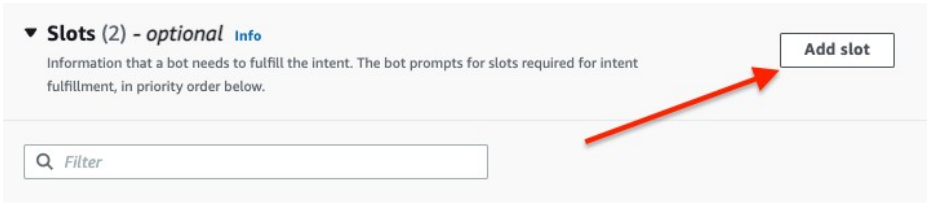
The `LookupPhoneNumberByName` intent has very focused inputs and outputs, but we can build many related intents, such as `LookupAddressByName`, and `LookupContactNamesByState`. Even though we can consider the `LookupPhoneNumberByName` intent as a search feature to a data source, it requires a different design thinking.

1. Change the Intent name to `LookupPhoneNumberByName`



2. Scroll down to Slots section. Add {FirstName} and {LastName} to look up a phone number.

For `LookupPhoneNumberByName`, we need both the `{FirstName}` and `{LastName}` to look up a phone number as they are both required.

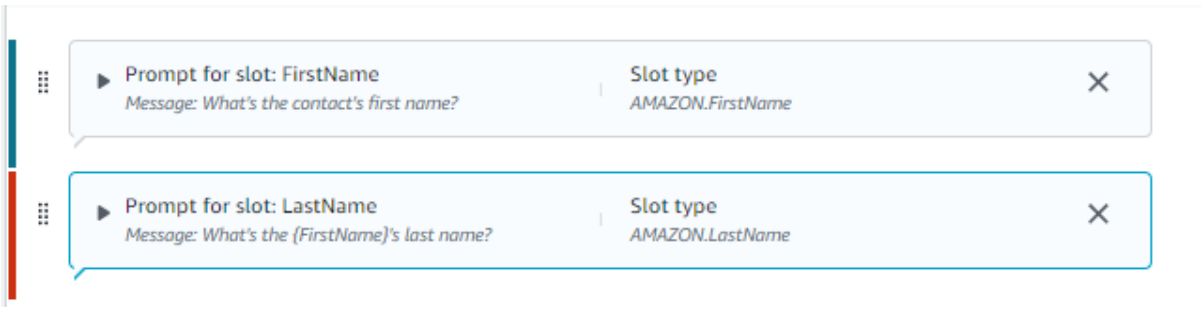


For slot FirstName:

- Name: FirstName
- Slot type: AMAZON.FirstName
- Prompts: What's the contact's first name?

For slot LastName:

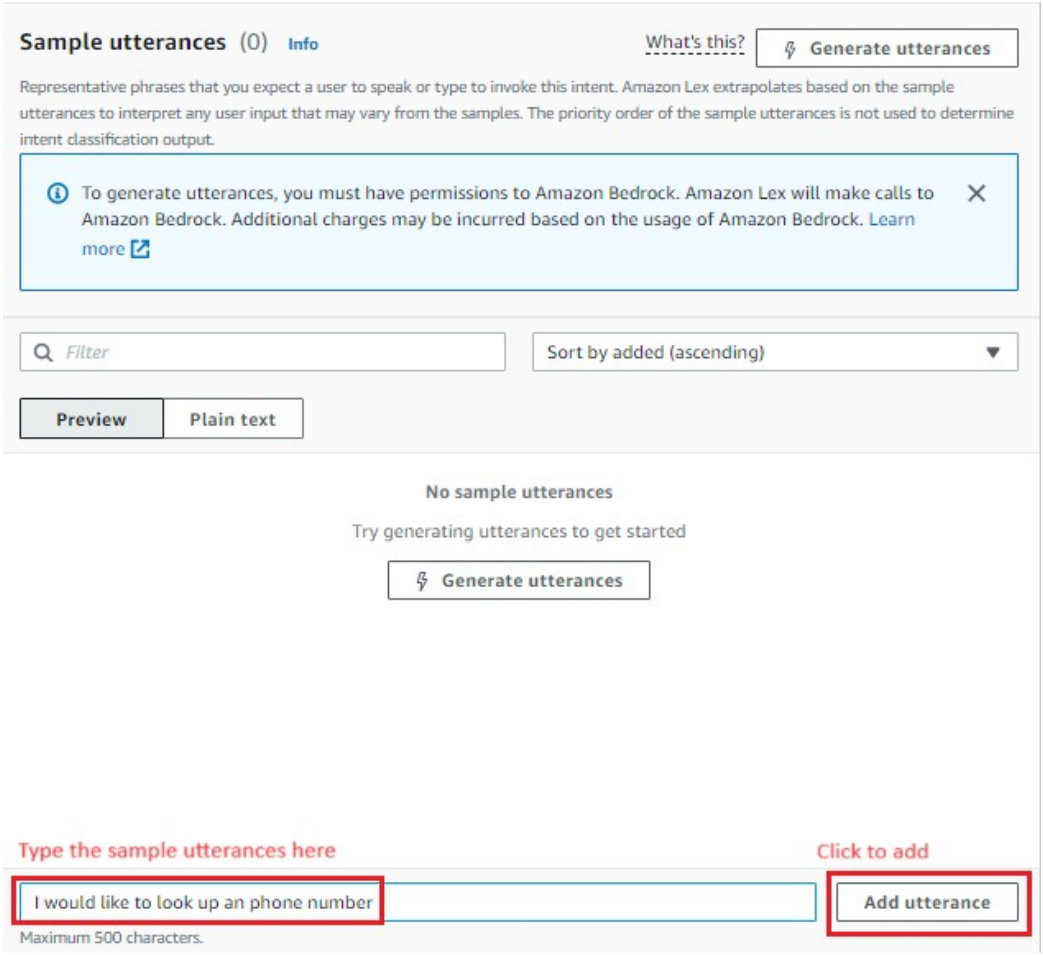
- Name: LastName
- Slot type: AMAZON. LastName
- Prompts: What's the {FirstName}'s last name?



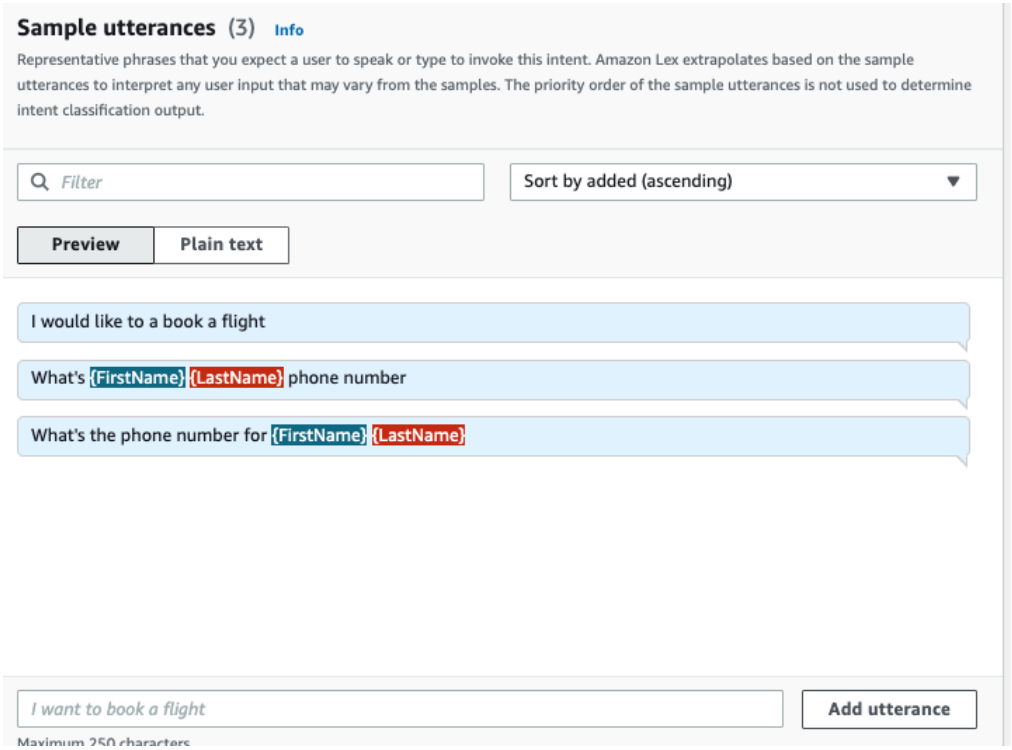
3. Let's extract the following contact information to leverage Amazon Comprehend.

Sample utterances are phrases that invoke the intent to perform an automated task.

Scroll up a bit to the Sample utterances. Add the following sample utterances for our `LookupPhoneNumberByName` intent:

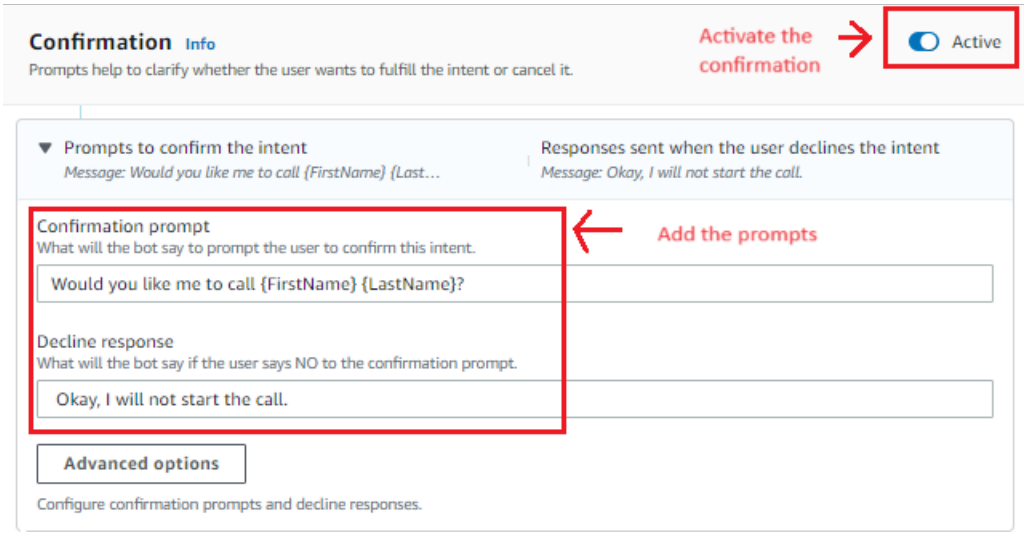


Your sample utterances should look like this:

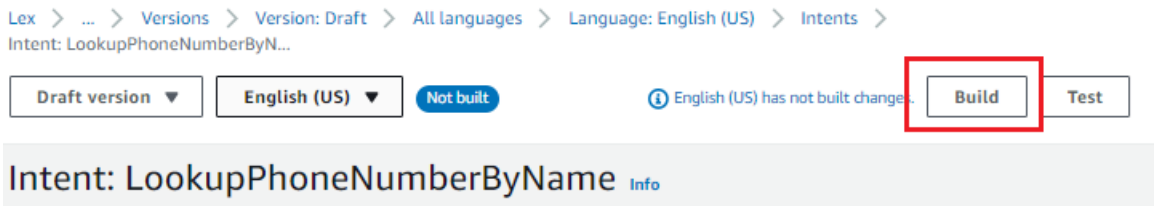


4. Create confirmation prompt and response for LookupPhoneNumberByName.
   A confirmation prompt is an opportunity to inform the user about the action about to be taken.

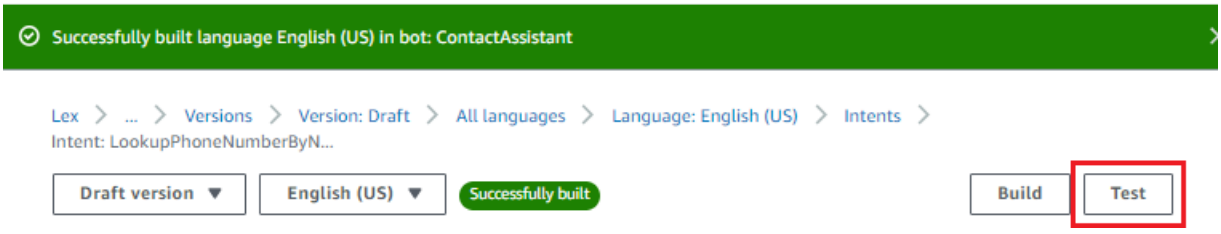
- Activate the Confirmation section
- Add the following prompts:
  - Confirmation prompt: Would you like me to call {FirstName} {LastName}?
  - Decline response: Okay, I will not start the call.



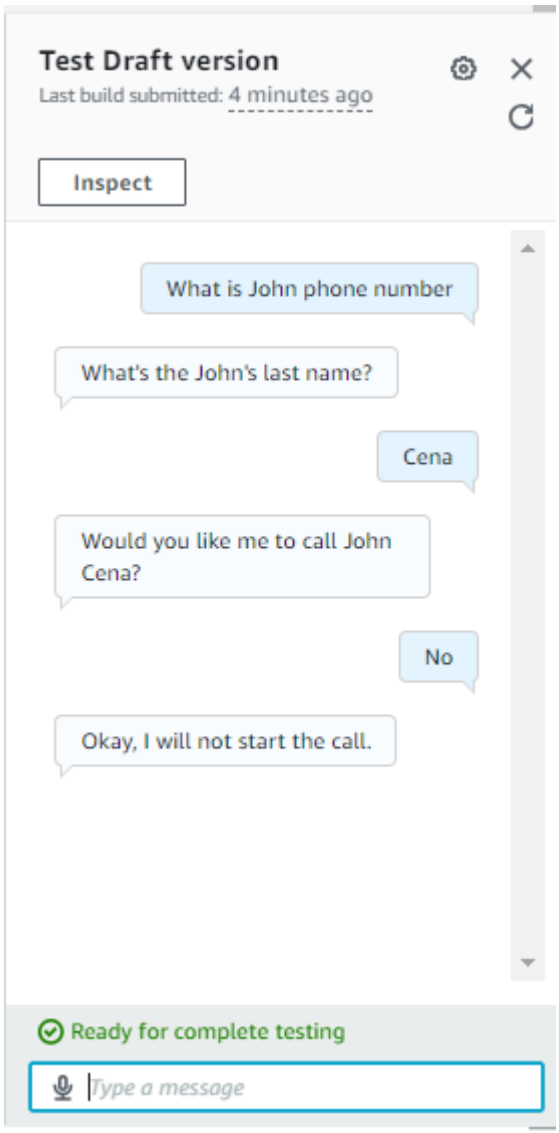
5. Once the above steps are completed, press **Save intent** button in the down right corner. Then build your bot using the Build option on the top right of the screen.



3.1 Once the bot is successfully built, use the Test option on the top of the screen to test the bot you just built.



Testing the bot



HOS submission instructions:

1. Please install the GitHub Desktop: https://cityuseattle.github.io/docs/git/github_desktop/

2. Clone, organize, and submit your work through GitHub Desktop: https://cityuseattle.github.io/docs/hoporhos