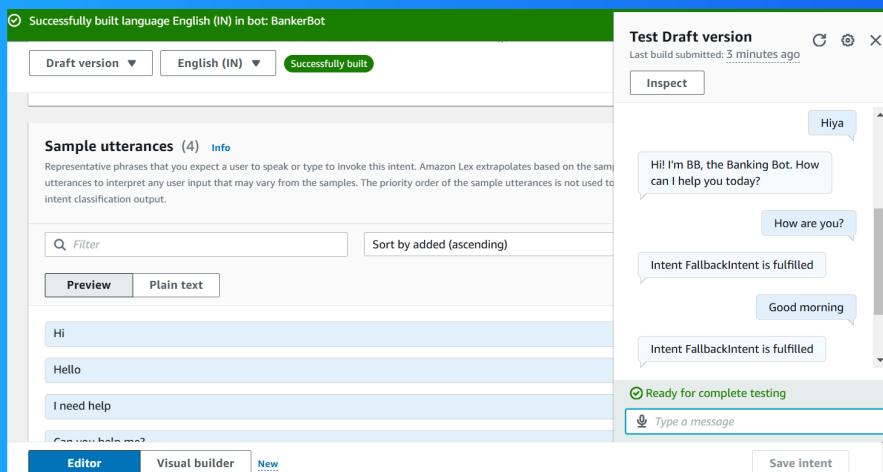




Build a Chatbot with Amazon Lex



Kunal Parkhade





Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is an AWS service to build chatbot that will take the user input in text or voice format and it will give response accordingly. It is useful because it provides end-to-end solution to build, publish, deploy and monitor chatbots.

How I used Amazon Lex in this project

I have used Amazon Lex in today's project for welcoming the user before doing conversation regarding balance account.

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

The one thing I didn't expect in this project is that I can test also after just building the chatbot.

This project took me...

This project takes time of 30 minutes



Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me around 5 minutes.

While creating my chatbot, I also created a role with basic permissions because Amazon Lex will need to call other AWS services.

In terms of the intent classification confidence score, I kept the default value of 0.40. This means chatbot will be 40% confident of understanding user prompt and act accordingly.

The screenshot shows the Amazon Lex console interface for a bot named 'BankerBot'. The top navigation bar includes 'Lex', 'Bots', 'Bot: BankerBot', 'Versions', 'Version: Draft', 'All languages', and 'Language: English (IN)'. Below the navigation, there are buttons for 'Draft version' (selected), 'English (IN)' (selected), and 'Not built'. A status message 'English (IN)' is shown with a 'Details' link. The main content area is titled 'English (IN)' and contains two sections: 'Conversation structure' and 'Language details'. The 'Conversation structure' section has tabs for 'Intents' and 'Slot types'. Under 'Intents', it says 'An intent is an action that fulfills a user's request.' and shows 'Intents: 2' with a 'View intents' button. Under 'Slot types', it says 'A slot type is a list of values used to train the machine learning model to recognize values for a slot.' and shows 'Slot types: 0' with a 'View slot types' button. The 'Language details' section includes fields for 'Description' (empty), 'Voice' (set to 'Kajal'), and 'Confidence threshold' (set to '0.40'). An 'Edit' button is located at the top right of this section.



Kunal Parkhade
NextWork Student

NextWork.org

Intents

Intents are like what do you want from chatbot conversation to do.

I created my first intent, WelcomeIntent, to welcome a user when they say hello.

The screenshot shows the Amazon Lex console interface. At the top, a green banner indicates "Successfully built language English (IN) in bot: BankerBot". Below this, there are dropdown menus for "Draft version" and "English (IN)", and a "Successfully built" button. On the left, a sidebar lists "Sample utterances (4)" with a "Info" link. A detailed description follows: "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 intent classification output." Below this are two buttons: "Preview" and "Plain text". Under "Preview", there are four sample utterances: "Hi", "Hello", "I need help", and "Can you help me?". On the right, the "Test Draft version" panel shows a conversation log. The user says "Hiya", the bot responds with "Hi! I'm BB, the Banking Bot. How can I help you today?", the user says "How are you?", the bot responds with "Intent FallbackIntent is fulfilled", the user says "Good morning", and the bot responds with "Intent FallbackIntent is fulfilled". A green checkmark at the bottom indicates "Ready for complete testing". A text input field at the bottom right says "Type a message". At the very bottom of the interface are buttons for "Editor", "Visual builder", and "New".



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter 'Hello'.

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered 'How are you?'. The error message occurred because I didn't include 'How are you' in my sample utterances.

The screenshot shows the Amazon Lex console interface. At the top, a green banner indicates 'Successfully built language English (IN) in bot: BankerBot'. Below this, there are tabs for 'Draft version' and 'English (IN)', with 'Successfully built' highlighted. On the left, a sidebar lists 'Sample utterances (4)' with a 'Info' link. It includes a 'Filter' input field and buttons for 'Preview' and 'Plain text'. The list of utterances includes 'Hi', 'Hello', 'I need help', and 'Can you help me?'. On the right, a 'Test Draft version' window is open, showing a conversation log. The log starts with 'Hiya' from the user, followed by responses from the bot: 'Hi! I'm BB, the Banking Bot. How can I help you today?', 'How are you?', 'Intent FallbackIntent is fulfilled', 'Good morning', and 'Intent FallbackIntent is fulfilled'. A green banner at the bottom of the test window says 'Ready for complete testing' with a 'Type a message' input field and a 'Save intent' button.



Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when your defined utterances does not match with user prompts.

I wanted to configure FallbackIntent because I wanted to know what will happen If I type something random then what will happen



Kunal Parkhade
NextWork Student

NextWork.org

Variations

To configure FallbackIntent, I defined Closing response for FallbackIntent and also added variations in it for making them sound more conversational.

I also added variations! What this means for an end user is that they will get dynamic range of responses.

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, the main workspace displays a configuration for a bot named 'BankerBot'. It includes sections for 'Response sent to the user after the intent is triggered' (with a message template), 'Message group' (with a message template), 'Variations - optional' (with two additional message templates), and 'More response options' (with a link to add custom payloads, SSML, and card groups). At the bottom, there are tabs for 'Editor' (which is selected) and 'Visual builder', along with a 'Save intent' button. On the right, a 'Test Draft version' window is open, showing a transcript of a conversation. The user says 'can i help you today?' and the bot responds with 'Good morning'. The user then says 'Hmm could you try rephrasing that? I can help you find your account balance, transfer funds and make a payment.' and the bot replies with 'What'. Finally, the user asks 'Can you describe again in few words?' and the bot is shown as 'Ready for complete testing'. A text input field at the bottom of the test window says 'Type a message'.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

