**Save User Info with a Lex Chatbot**

Welcome to part-4 of AI x AWS series.

In the first three projects, I’ve learnt how to:

* 💬 Define intents
* 🔀 Provide variations in the bot's responses
* 🌟 Set up a custom slot type
* 🤝 Connect chatbot with AWS Lambda

In this project, I have learnt how to set up **context carryover,** which is a technique that helps the chatbot remember things learnt about the user (like their birthday) from one intent and share them with other intents!

**Step 1:** Set up a new Lex Chatbot.

Complete all the steps in projects 1, 2, and 3.

**Step 2:** Remember information stored in CheckBalance

Now that BankerBot can check a user's bank account balance, my next level up is to remember the juicy information I've collected.

After all, it can get pretty frustrating if I keep asking a user for their birthday—especially within the same chat session!

**In this step, get ready to:**

* Save the user's birthday from the CheckBalance intent.
* In your **CheckBalance** intent page, scroll down to the **Contexts** panel.
* Under the **Output contexts** drop-down, choose **New context tag**.

A screenshot of a computer

AI-generated content may be incorrect.

**What are context tags?**

**Context tags** in Amazon Lex are used to store and check for specific information across different parts of a conversation. They help save the user from having to repeat certain information

There are two types of context tags in Amazon Lex:

1. **Output context tag:** This tells the chatbot to remember certain details after an intent is finished, so other parts of the conversation can use this stored information later. For example, the account type from BalanceCheck could be saved and reused
2. **Input context tag:** This checks if specific details are already available before an intent activates. For example, FollowupCheckBalance will check if this conversation already has the user's date of birth saved somewhere, so it won't need to ask for that information again.

* Name your new context contextCheckBalance

A screenshot of a computer

AI-generated content may be incorrect.

* I'll set up the timeout for 5 turns or 90 seconds. That way, I won’t remember a user's birthday for too long—keeping things secure while still making the experience smoother!
* Choose **Add**.
* Choose **Save intent**.
* Choose **Build** - time for a few quick questions.
* Choose **Test**.
* Check that the bot still operates the same as usual - i.e., no errors have popped from creating the context tag.

A screenshot of a chat

AI-generated content may be incorrect.

**Step 3:** Create the FollowupCheckBalance Intent

I'll set up a new intent to handle follow-up balance check requests without requiring authentication.

Since I've already used **CheckBalance** to give someone their account balance, I need a way to handle follow-up questions like, *"What about my other account's balance?"* without asking for the user's birthday again.

But here's the catch—**CheckBalance** isn’t designed to answer follow-up questions like *"What about XYZ?"* Adding these to its utterance list would just confuse me!

To solve this, I'll create a new intent dedicated to follow-ups. 😎

**Steps I'll take:**

1. Head to the **Intents** page.

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose **Add intent** → **Add empty intent**.

A screenshot of a chat box

AI-generated content may be incorrect.

1. Set up the following properties:
   * **Name**: FollowupCheckBalance
   * **Description**: Intent to allow a follow-up balance check request without authentication.
   * **Input context**: contextCheckBalance
   * **Sample utterances**:
     + - 1. How about my {accountType} account?
         2. What about {accountType} ?
         3. And in {accountType} ?

A screenshot of a computer

AI-generated content may be incorrect.

* Add a new slot:
* Name: accountType
* Prompt: For which account would you like your balance?
* Slot type: **accountType**

A screenshot of a computer screen

AI-generated content may be incorrect.

* Add another new slot:
  + Name: dateOfBirth
  + Prompt: For verification purposes, what is your date of birth?
  + Slot type: **AMAZON.Date**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4:** Finishing Touches for FollowupCheckBalance

I've set up the structure for the **FollowupCheckBalance** intent—nice!

But… something’s missing. 🧐

The **FollowupCheckBalance** intent has an input context tag, but I haven’t actually carried over the user's date of birth from **CheckBalance** yet.

**Time to fix that!**

I'll set up **FollowupCheckBalance**'s dateOfBirth slot to use the saved information:

1. Stay on the **FollowupCheckBalance** intent page.
2. Expand the **dateOfBirth** slot.
3. Choose **Advanced options**.
4. Scroll down to the **Default values** panel.
5. Enter: #contextCheckBalance.dateOfBirth

A screenshot of a computer

AI-generated content may be incorrect.

**What does #contextCheckBalance.dateOfBirth mean?**

This tells Amazon Lex that the input context contextCheckBalance should have the value of dateOfBirth in CheckBalance.

A screenshot of a computer

AI-generated content may be incorrect.

* Choose **Add default value**.
* Choose **Update slot**.
* Head to the **Fulfillment** pane to make sure the Lambda function is also connected to this intent (so random balances are still being returned to the user).
* Expand **On successful fulfillment**.
* Choose **Advanced options**.
* Head to the **Fulfillment Lambda code hook** panel.
* Select the checkbox to enable a fulfillment Lambda for this intent.
* Choose **Update options**.
* Choose **Save intent**.
* Choose **Build** - you know the drill!
* Choose **Test**.
* In the first test, try to trigger the new FollowupCheckBalance intent I've just created **without** triggering CheckBalance first.

e.g. ask your chatbot What about checking?

**Why isn't the intent working?**  
No matter which utterance you use, you will just get an error response. This is because the FollowupCheckBalance intent's input context isn't available yet. The intent doesn't know your birthday!

* For the second test, ask for a balance in your account - activate the **CheckBalance** intent first.
* Then, the context for date of birth will be carried over to the **FollowupCheckBalance** intent.
* Wooo!! the bot now uses the context from the first check balance request, and doesn't ask for your birthday again for your second request.

A screenshot of a chat

AI-generated content may be incorrect.