

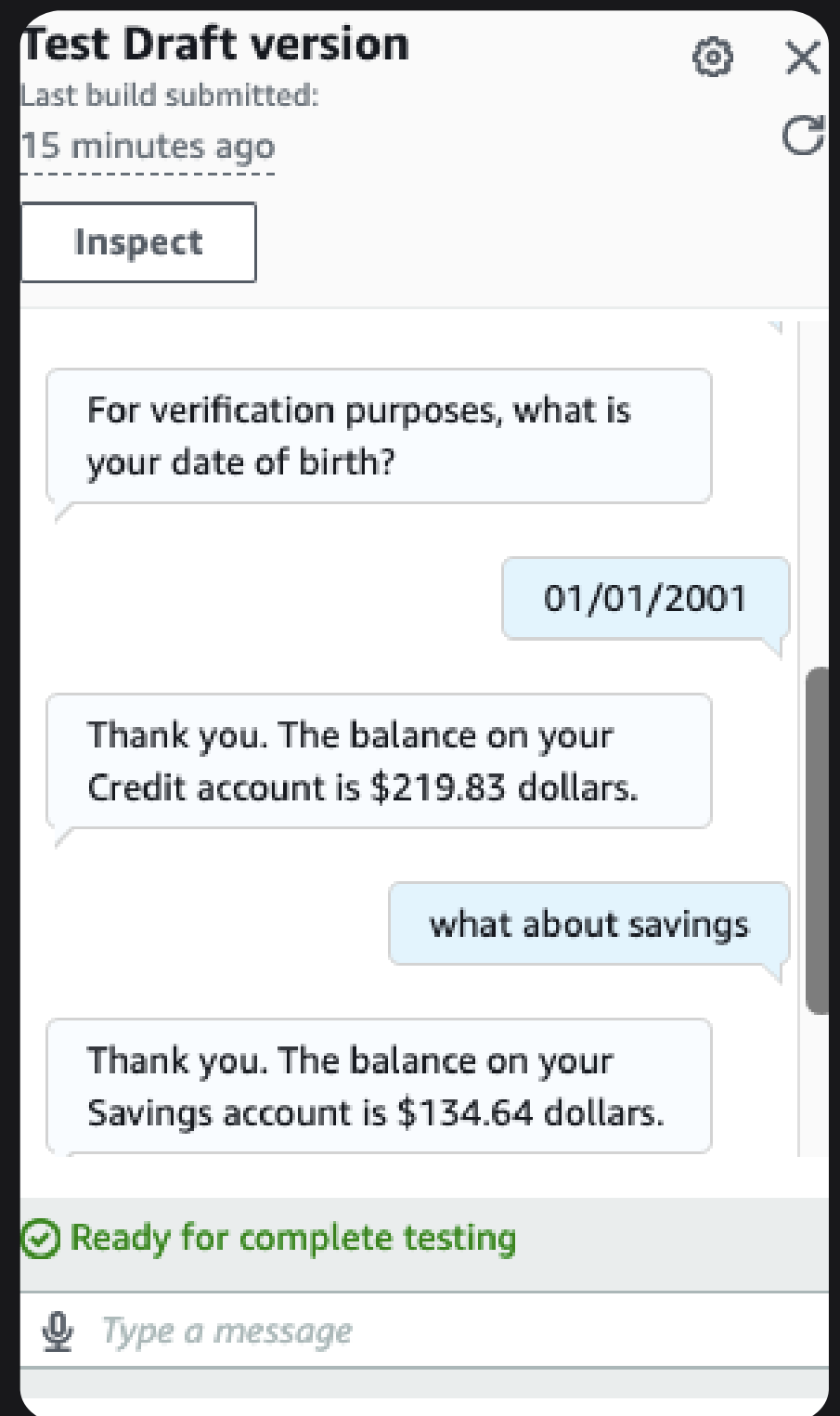
How I built a chatbot with Amazon Lex

that can remember user info!



Abdigani Awil Mohamed

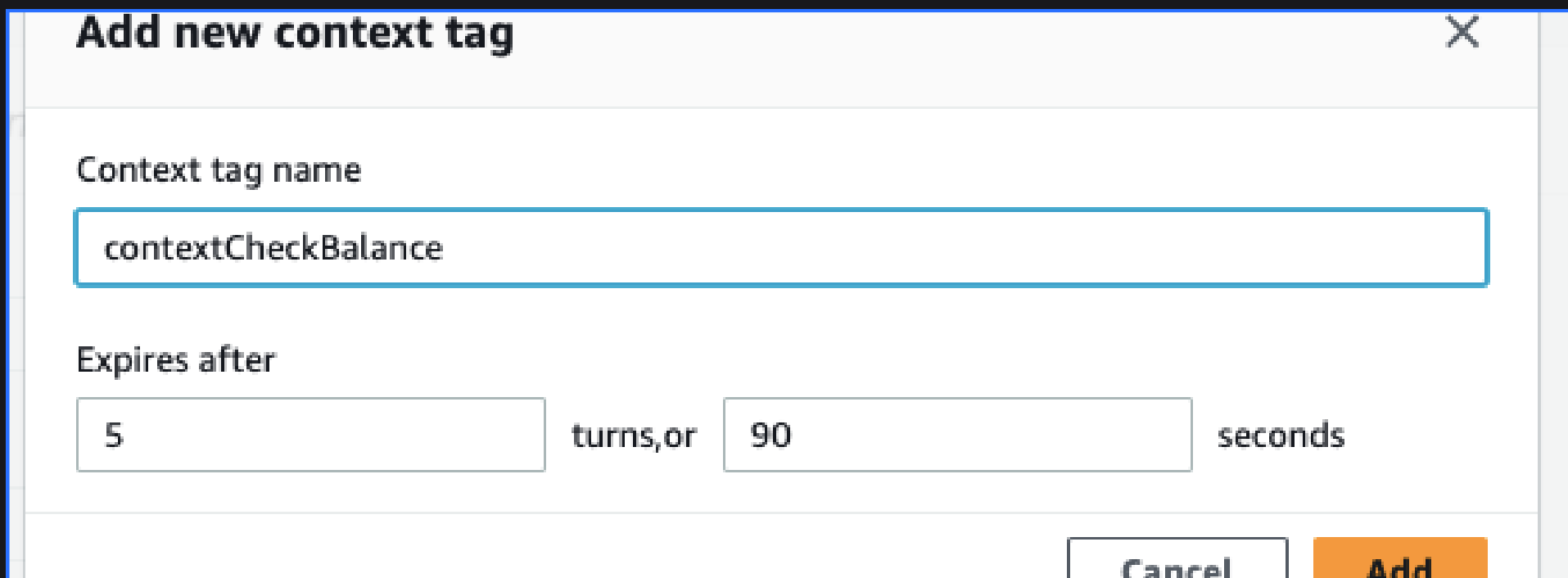
@abdiganiawil10



Context Tags

- Context tags are tools for Amazon Lex to remember specific pieces of information gathered from a conversation, and reuse that information throughout the session with its user.
- There are two types of context tags, they are output context tags and input context tags.
- I created an output context tag called contextCheckBalance, and I created this in the intent CheckBalance.

A look at output contexts



Add new context tag

Context tag name

contextCheckBalance

Expires after

5 turns, or 90 seconds

Cancel Add



Abdigani Awil Mohamed

@abdiganiawil10

A Follow-Up Intent



- I created a new intent called **FollowupCheckBalance**. The purpose of this intent is to let the user check another account's balance without having to provide their date of birth again.
- This intent is related to the previous intent I made, **CheckBalance**, because FollowUpCheckBalance will only get triggered after the user has checked their balance once already(e.g. triggered Check Balance).
- I created an input context, **contextCheckBalance**, using the same tag as the output context tag I have set up in the CheckBalance intent. This means that the input information we are looking for in this intent FollowupCheckBalance can now be retrieved from the CheckBalance intent through this tag.

A look at input contexts

▼ Default values - optional

No default values
You haven't added any default values yet.

Provide a default value, #value for a context value or [variable] for session variable.

#contextCheckBalance.dateOfBirth

Add default value

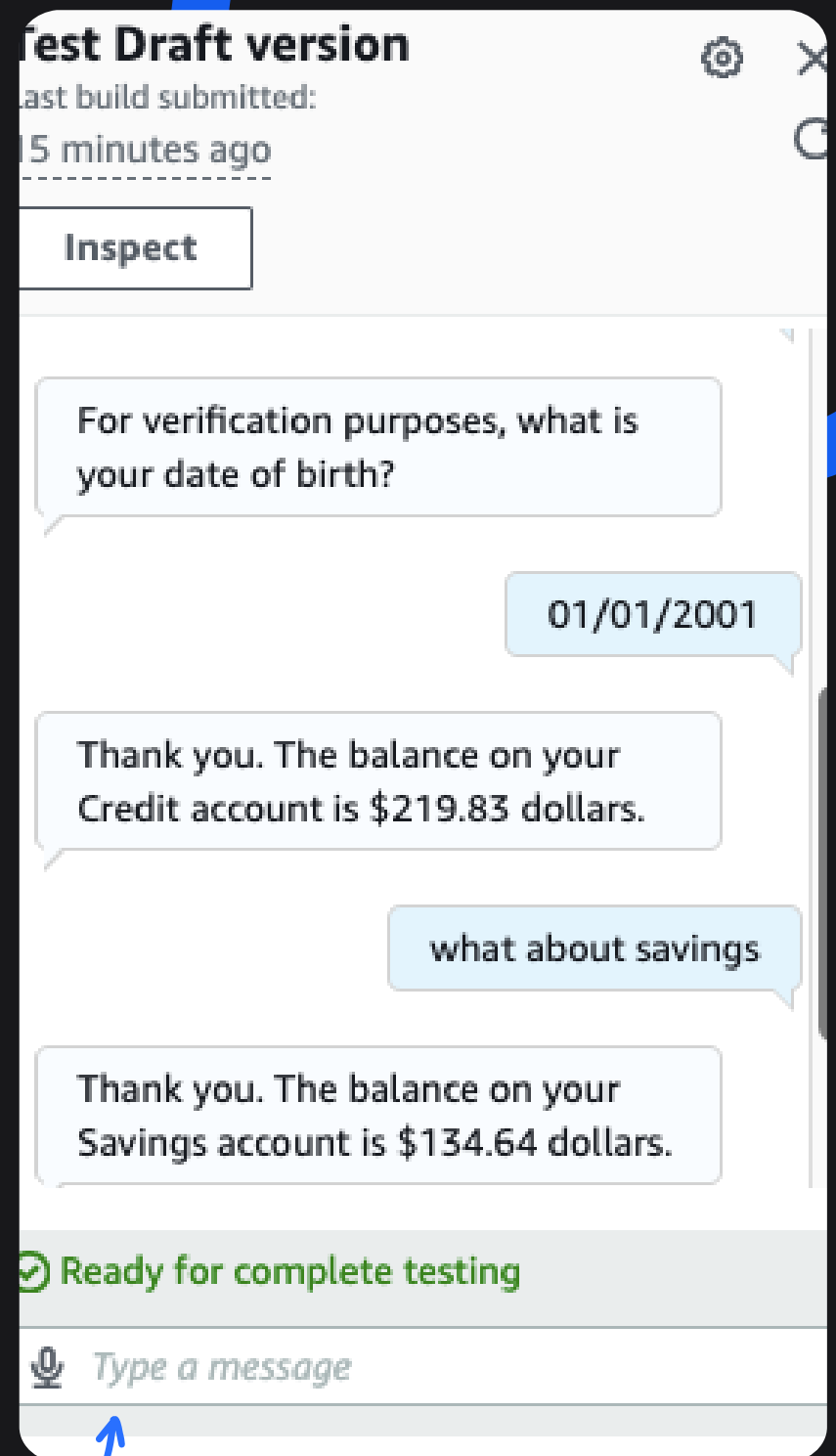


Abdigani Awil Mohamed

@abdiganiawil10

Context Tags in Action

- Conversation time! I built and tested my bot after creating the context tags and new intent.
- To see the context tags and the follow up in intent in action, I first triggered the CheckBalance intent, and then I followed up with the utterance. "What about savings" to trigger FollowUpCheckBalance.
- If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, my chatbot would not have the context needed to fulfil the conversation. As a result, it will return the fallbackintent i.e. let the user know it doesn't understand the request being made.



My chatbot now carries
over the user's date of
birth to the next intent!



Abdigani Awil Mohamed

@abdiganiawil10

My Key Learnings

01

Context tags are used in Amazon Lex to store and retrieve specific information across different conversation parts. They help users avoid repeating information by remembering details from previous interactions.

02

The difference between input and output context tags is that input context tags check for existing information before an intent, while output context tags store information for later use in the conversation.

03

I created the input context in FollowupCheckBalance by specifying 'contextCheckBalance' as the input context tag, ensuring the intent retrieves previously stored user details, such as the date of birth.

04

FollowupCheckBalance requires the presence of the 'contextCheckBalance' output context from a previous CheckBalance intent to retrieve stored user details such as the date of birth, thus it cannot function successfully without prior context.

05

I've learned the importance of designing conversation flows that anticipate user needs and preferences, ensuring a smoother and more personalised interaction with the chatbot.



Abdigani Awil Mohamed

 @abdiganiawil10

ERROR! 🙄 OMG!

An error I ran into was...

My FollowupCheckBalance intent still asking me for my birthday even though that information was already saved in an output context tag after triggering CheckBalance.

- I ran into this error because I had forgotten to save the default value of my FollowupCheckBalance's date of birth slot.
- ☒ I solved this error by returning to my slot's default values panel and adding the input context.

Screenshot of error here

The screenshot displays the 'Inspect' interface for a 'Test Draft version' of a Lex bot. The left sidebar shows the 'Summary' and 'JSON input and output' tabs. The main area is divided into sections: 'Intent' (CheckBalance), 'Slots' (Elicitation), and 'Active contexts' (Number of turns or seconds). The 'Slots' section shows a table with columns 'Slots' and 'Elicitation'. The 'dateOfBirth' slot is listed with a value of '-'. The 'Active contexts' section shows 'contextCheckBalance' with a value of '4 turns or 83s'. The right sidebar shows the chat history with messages: '01/02/2001', 'Thank you. The balance on your Credit account is \$111.49 dollars.', 'what about my savings', and 'For verification purposes, what is your date of birth?'. At the bottom, there is a green status bar indicating 'Ready for complete testing' and a text input field with a microphone icon.

| Slots | Elicitation |
|-------------|-------------|
| accountType | Savings |
| dateOfBirth | - |

| Active contexts | Number of turns or seconds |
|---------------------|----------------------------|
| contextCheckBalance | 4 turns or 83s |



Abdigani Awil Mohamed

@abdiganiawil10



Final thoughts...

- This part of the project took me 30 minutes. Writing documentation took me around 25 minutes.
- Delete EVERYTHING at the end! Let's keep this project free :)
- In the next phase of this project, I'm excited to level up my Lex bot one more time by creating an intent that can help users transfer funds between accounts. I'm also using AWS CloudFormation to recreate my bot in seconds!



Abdigani Awil Mohamed

 @abdiganiawil10