Slack Reroute Lambda

The purpose of this function is take all requests from slack and to reroute them to the appropriate Step Function. The 6 Step Functions that slackReroute starts are the App Added Flow, the New Message Posted Flow, the Marked Answer Flow, the Dismiss Button Flow, the Helpful Button Flow, and the Not Helpful Button Flow.

App Added Flow

This flow is triggered when the user adds the app to a Slack channel.

Preliminary Work

This step uses the preliminaryAppAddedWork lambda.

This lambda's purpose is to do the needed preliminary work of adding the Slack Workspace, the Slack Channel, and all the Slack Users in that channel into our Database if they do not exist already.

Once this preliminary work is finished, the next state machine in the step function starts.

Start Processing Messages

This step uses the startProcessingMessages.

This lambda's purpose is to get all messages from the past year in the Slack Channel and to batch insert them into SQS.

When this lambda finished the App Added Flow Step Function is complete.

Read From Slack Channel Added SQS Lambda

This lambda is triggered when there is something in SQS for it to grab. It reads from SQS one message at a time and then moves that message along into the Single Message Processing Step Function.

NOTE: Currently SQS cannot trigger a Step Function so we had to have an intermediary Lambda to read from and then start the Step Function.

Single Message Processing Flow

This flow is triggered by the readFromSlackChannelAddedSQSLambda.

Message Cleaner

This step uses the newMessageCleaner Lambda.

This lambda's purpose is to determine if the message is a possible candidate to be used in our system. We do this by checking to see if this message has a thread associated with it and if the message does not contain any files.

When this lambda finished we either continue to the next step (our natural language processing step) or we end. This is decided on by what is returned from the Message Cleaner lambda

NLP (Natural Language Processing)

This step uses the nlp lambda.

This lambda's purpose is to determine if the now cleaned message is a question or just a regular message. If it is a question we continue to the next step (our Message To Vec step) or we end.

Message To Vec

This step uses the doc2vec lambda.

This lambda's purpose is to take the text from the now cleaned question and convert it into a vector of numbers to represent that text.

When this is finished we move on to the Create Question in DB step.

Create Question in DB

This step uses the endOfMessageProcessing lambda.

This lambda's purpose is to use the data that Slack has given us, the vector we have created, and the UUIDs we generate to create a SlackQuestion in our DB.

When this step is finished the Step Function ends.

New Message Posted Flow

This flow is triggered when the user posts a new message in Slack.

Message Cleaner

This step uses the newMessageCleaner Lambda.

This lambda's purpose is to determine if the message is a possible candidate to be used in our system. We do this by checking to see if this message has a thread associated with it and if the message does not contain any files.

When this lambda finished we either continue to the next step (our natural language processing step) or we end. This is decided on by what is returned from the Message Cleaner lambda.

NLP (Natural Language Processing)

This step uses the nlp lambda.

This lambda's purpose is to determine if the now cleaned message is a question or just a regular message. If it is a question we continue to the next step (our Message To Vec step) or we end.

Message To Vec

This step uses the doc2vec lambda.

This lambda's purpose is to take the text from the now cleaned question and convert it into a vector of numbers to represent that text.

When this is finished we move on to the Find Similar Questions step.

Find Similar Questions

This step uses the cosine_similarity lambda.

This lambda's purposed is to get all messages in our Database for the channel we are in and then to compare each of their vectors to the new message's vector to find the most similar question to the new question.

Once this is finished we pass the most similar question onto the next step.

Send Message To Slack

This step uses the sendMessageToSlack lambda.

This lambda's purpose is to use the most similar question to craft a message that we send to the user who asked the new question.

To do this we get a link to the most similar question's threaded messages, from Slack, and then hyperlink it to some text and three buttons in the message we create. Once created we send the message to User for them to see.

When this is finished, the step function closes.

Marked Answer Flow

This flow is triggered when the user uses the shortcut we created to mark a message as answer.

Is Possible Answer

This step uses the isPossibleAnswer lambda.

This lambda's purpose is to determine if the message marked is a possible answer. To do this we check to see if the message marked is inside of a thread and if it is not the parent message of that thread.

If this message is not a possible answer, we move onto the Bad Answer Marked step. If this message is a possible answer we move onto the Natural Language Processing step.

Bad Answer Marked

This step uses the badAnswerMarked lambda.

This lambda's purpose is to send the user a message letting them know that the message they marked as an answer does not qualify as an answer.

When this step finished the step function closes.

NLP (Natural Language Processing)

This step uses the nlp lambda.

This lambda's purpose is to determine if the message passed in is a question. For the Marked Answer Flow, we pass in the parent of the thread where the message was marked as an answer to determine if the answer marked has an associated question.

If the parent message is not a question, we proceed onto the Bad Answer Marked step. If the parent message is a question, we proceed onto the Thank You step.

Thank You

This step uses the thankYouMessage lambda.

This lambda's purpose is to send a message to the user thanking them for marking an answer.

When this finshes we proceed onto the Doc2Vec step.

Doc2Vec

This step uses the doc2vec lambda.

This lambda's purpose is to take the text of a message and convert it to a vector of numbers. For the Marked Answer Flow, we do this for the parent message of the answer.

When this finishes we move onto the Create Q and A step.

Create Q and A

This step uses the endOfMarkedAnswer lambda.

This lambda's purpose is to use the information from slack, the vector we created, and the UUIDs we generate to create the Answer and the Question in the database. This makes this question and answer pair, useable for when similar questions are asked in the future.

When this step finishes the step function closes.

Dismiss Button Flow

This flow is triggered when the Dismiss Button, attached to the message we sent to the user after posting a question, is pressed.

Dismiss

This step uses the dismissButton lambda.

This lambda's purpose is to use the responseURL from Slack to delete the message that we posted.

When this step completes the step function finishes.

Helpful Button Flow

This flow is triggered when the Helpful Button, attached to the message we sent to the user after posting a question, is pressed

Helpful

This step uses the helpfulButton lambda.

This lambda's purpose is to use responseURL to replace the message that was sent to the user with a thank you message, to post the link, to the answer that was helpful, in a thread of the question that was asked, and to increase the helpful answers upvote count.

Not Helpful Button Flow

This flow is triggered when the Not Helpful Button, attached to the message we sent to the user after posting a question, is pressed.

Not Helpful

This step uses the notHelpfulButton lambda.

This lambda's purpose is to use responseURL to replace the message that was sent to the user with a thank you message and to decrease the not helpful answers upvote count.