coursera AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Introduction to Week 4 Video: Introduction to Week 4 Exercise 4.1: Make Amazon Lex Smarter 2 min Extending Our App Video: Extending Our Exercise 4.1: Make LEX smarter App Part 1 14 min Video: Extending Our We have done well. We have a working server-less data driven website on a content delivery network :) App Part 2 6 min We now are going to leverage the Amazon LEX bot you created in Week 1 and turn our application from Video: Adding a Channel a text app to a voice enabled app. Therefore the API endpoint that we used for the text application will no longer work if we point it to a Video: Exercise 4.1 Lambda function that is expecting a conversation LEX-style "phrase". Introduction So, out with the old and in with the new. Reading: Exercise 4.1: The goal of this exercise is to swap out the Lambda function currently sitting behind API Gateway for a Make Amazon Lex Smarter new one that simply takes a "text phrase" from the browser and passes it over to LEX to interpret. When 45 min LEX replies to this message we pass that messages all the way back to the browser. Week 4 Wrap Up and This new lambda function's code is just passing messages back and forth between the browser and LEX, Resources nothing more We will call this function lavarery AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter We are leveraging the fact that the Chrome browser can listen to your device microphone and convert what Introduction to Week 4 you say to text in real time. This is a JavaScript API, that you don't need to worry about as we wrote the code Video: Introduction to for this already. There is another JavaScript API that converts text back to speech! Super useful. We shall Week 4 be leveraging that too :) 2 min Extending Our App This way we can literally "talk" to the browser, and essentially be communicating with LEX, in what feels like "real time". Video: Extending Our App Part 1 We are going to approach this exercise in multiple steps: 14 min Video: Extending Our 1) We need to make your Lex bot smarter with what we call a **validation hook**. This way it can extract App Part 2 data from DynamoDB and provide weather information. Just like our text app did. 2) Test you smarter LEX bot in the LEX console. You should notice the difference in smarts Video: Adding a Channel 6 min 2) Create a lambda function to proxy text phrases to and from your new smarter LEX bot, and text it. Video: Exercise 4.1 3) Change the API gateway configuration so it points to this new proxy function, and test it. Introduction 4) Go to a NEW secret website (that you upload when you did week 2) and may not have realized it Reading: Exercise 4.1: called voice.html. Make Amazon Lex Smarter Week 4 Wrap Up and 1. Steps to creating a validation Lambda function for the Lex Resources bot. This will make LEX smarter:) Sign in to the AWS Management Console and in the Find Services search box type lambda and AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter For Function name type in getSmartWeather. Introduction to Week 4 Again leave the Runtime as Node.js 8.10. Video: Introduction to For Execution role leave it as Use an existing role. Week 4 For Existing role choose service-role/Get-Weather. **Extending Our App Basic information** Video: Extending Our App Part 1 Enter a name that describes the purpose of your function 14 min getSmartWeather Use only letters, numbers, hyphens, or underscores with no spaces Video: Extending Our App Part 2 Node js 8.10 Video: Adding a Channel Permissions Info 6 min Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add trigger Video: Exercise 4.1 Introduction Execution role Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console 3 min Use an existing role ۳ Reading: Exercise 4.1: Existing role Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs. Make Amazon Lex * C service-role/Get-Weather Smarter View the Get-Weather role on the IAM console. Week 4 Wrap Up and Resources Click Create function. Paste the following into the index.js tab in the inline code editor. 1 ▼ exports.handler = function(event, ctx, cb){ var mv response = {}: AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter type . tiltitoiot , 12 -"message": { "contentType": "PlainText", 13 Introduction to Week 4 "content": "Name the city your cat lives in, thanks" 15 Video: Introduction to "intentName": "CatWeather", 16 Week 4 17 -"slots": { 18 "city_str": null 19 "slotToElicit" : "city_str" 20 **Extending Our App** 21 22 }; 23 return cb(null, my_response.body); Video: Extending Our 24 App Part 1 25 14 min 26 city_str = event.currentIntent.slots.city_str, 27 AWS = require("aws-sdk"), DDB = new AWS.DynamoDB({ 28 -Video: Extending Our 29 apiVersion: "2012-08-10", App Part 2 region: "us-east-1" 30 31 32 lookup_name_str = city_str.toUpperCase(), 33 + Video: Adding a Channel 34 TableName: "weather", 35 KeyConditionExpression: "sc = :v1", 36 + ExpressionAttributeValues: { 37 ▼ ":v1":{ Video: Exercise 4.1 "S": lookup_name_str 38 Introduction 39 40 3 min 41 ProjectionExpression: "t" 42 }; Reading: Exercise 4.1: 43 Make Amazon Lex 44 console.log(params); Smarter 45 + DDB.query(params, function(err, data){ 46 if(err){ 47 48 Week 4 Wrap Up and 49 50 + if(data.Items && data.Items[0] && data.Items[0].t){ Resources console.log("city weather found"); 51 52 console.log(data.Items[0]); my_response.statusCode = 200; 53 54 my_response.body = { 55 + "sessionAttributes": { 56 "temp_str": data.Items[0].t.N, 57 "city_str": event.currentIntent.slots.city_str 58 "dialogAction":{ 59 + AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter 68 -69 console.log("city weather not found for " + lookup_name_str); Introduction to Week 4 70 my_response.statusCode = 200; 71 my_response.body = { 72 -"dialogAction": { Video: Introduction to 73 "type": "ElicitSlot", Week 4 74 -"message": { 2 min 75 "contentType": "PlainText", "content": "Please try another city, we couldn't find 76 the weather for that city' Extending Our App 77 78 "intentName": "CatWeather", Video: Extending Our "slots": { 79 -App Part 1 80 "city_str": null 81 14 min 82 "slotToElicit" : "city_str" 83 Video: Extending Our 84 App Part 2 85 86 return cb(null, my_response.body); 6 min 87 }); 88 }; Video: Adding a Channel 6 min Video: Exercise 4.1 This code checks for an existence of a slot (the city), if it's there, wonderful. Look it up in Dynamo and Introduction return the temperature. 3 min If not, LEX will need to ask for a city. Once LEX has established the city, meaning there is existence of the Reading: Exercise 4.1: Make Amazon Lex slot, it can just do a look up. Smarter You get the idea. Week 4 Wrap Up and Scroll down to Basic settings and change the timeout to 1 min and 5 sec. Resources Click Save. Let's test it with a dummy LEX payload before we tie it into Lex and publish it. Click Test. Leave the Event template as Hello World. For Front name type in actsmartweathermost AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter invocacionsource : Dialogcodenook , "userId": "1012602", "sessionAttributes": { Introduction to Week 4 6 "bot": { 7 + Video: Introduction to "name": "WeatherCatBot", 8 Week 4 9 "alias": "\$LATEST", 10 "version": "\$LATEST" 2 min 11 "outputDialogMode": "Text", 12 Extending Our App 13 -"currentIntent": { "name": "CatWeather", 14 Video: Extending Our "slots": { 15 + "city_str": "CHICAGO" 16 App Part 1 17 14 min "confirmationStatus": "None" 18 19 } 20 } Video: Extending Our App Part 2 6 min Click Create. Video: Adding a Channel 6 min Click Test. Video: Exercise 4.1 We should see the following output: Introduction 3 min Reading: Exercise 4.1: "sessionAttributes": { 3 "temp_str": "42", Make Amazon Lex 4 "city_str": "CHICAGO" Smarter "dialogAction": { "type": "Close", 8 "fulfillmentState": "Fulfilled", Week 4 Wrap Up and 9 + "message": { Resources 10 "contentType": "PlainText", 11 "content": "42" 12 13 14 AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter "messageVersion": "1.0", Introduction to Week 4 "invocationSource": "DialogCodeHook", "userId": "1012603", Video: Introduction to "sessionAttributes": { Week 4 6 2 min "name": "weather", 8 9 "alias": "\$LATEST" **Extending Our App** "version": "\$LATEST" 10 11 Video: Extending Our "outputDialogMode": "Text", 12 13 + "currentIntent": { App Part 1 "name": "catWeather", 14 15 + "slots": { "city_str": "BANANA" 16 17 Video: Extending Our "confirmationStatus": "None" 18 App Part 2 19 20 } Video: Adding a Channel 6 min Click Save. Video: Exercise 4.1 Click Test. Introduction 3 min You should see the following output: Reading: Exercise 4.1: Make Amazon Lex 1 - { Smarter 2 + "dialogAction": { "type": "ElicitSlot", 4 + Week 4 Wrap Up and "contentType": "PlainText", "content": "Please try another city, we couldn't find the weather for that 6 Resources 8 "intentName": "CatWeather", "slots": { 9 + 10 "city_name_str": null 11 12 "slotToElicit": "city_str" 13 14 } AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter ■ Office again increment the city_ta to 1012004 and set the city_set to muit. As if we didn't pass in a city in our phrase. i.e "is it cold out?" Introduction to Week 4 Video: Introduction to Week 4 "messageVersion": "1.0", "invocationSource": "DialogCodeHook", "userId": "1012604", Extending Our App 5 + "sessionAttributes": { 6 7 + "bot": { Video: Extending Our "name": "weather", App Part 1 "alias": "\$LATEST", 9 14 min "version": "\$LATEST" 10 11 "outputDialogMode": "Text", 12 Video: Extending Our 13 -"currentIntent": { App Part 2 14 "name": "catWeather", "slots": { 15 + "city_str": null 16 17 Video: Adding a Channel 18 "confirmationStatus": "None" 6 min 19 20 } Video: Exercise 4.1 Introduction 3 min Click Save. Reading: Exercise 4.1: Make Amazon Lex Click Test. Smarter You should see the following output: 45 min Week 4 Wrap Up and Resources "dialogAction": { "type": "ElicitSlot", "message": { "contentType": "PlainText", "content": "Name the city your cat lives in, thanks" "intentName": "CatWeather", AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Introduction to Week 4 Awesome it is all working, we now have a function that we know works Video: Introduction to We just need to wire it up to LEX and try it in the LEX console. Week 4 2 min Lets give LEX it's new brain:) Extending Our App 2. Wire it into Lex - (Turbo charge LEX) Video: Extending Our App Part 1 Click Services type in lex in the search box or select Amazon Lex from the History list. Video: Extending Our Click our WeatherCatBot App Part 2 Scroll down to Lambda initialization and validation and select Initialization and validation 6 min code hook. Video: Adding a Channel Select getsmartWeather for our function and Latest for Version or alias. Video: Exercise 4.1 Lambda initialization and validation 6 Introduction Initialization and validation code hook Reading: Exercise 4.1: Make Amazon Lex Lambda function getSmartWeather • Smarter View in Lambda console C Week 4 Wrap Up and Latest Version or alias Resources At the Add permission to Lambda Function click OK. Expand the Confirmation prompt and remove the check for Confirmation prompt. Click Save Intent at the bottom. AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Introduction to Week 4 Click Test Chatbot at the right if it is not already expanded. Video: Introduction to Test out talking to your bot. Week 4 **Extending Our App** User 🙋 Chatbot @ Video: Extending Our App Part 1 Can my cat go out in banana? 14 min Please try another city, we couldn't find the weather for that city Video: Extending Our App Part 2 Can my cat go out in DENVER 38 Video: Adding a Channel Can my cat go out? Video: Exercise 4.1 Name the city your cat lives in, thanks Introduction 3 min alto Reading: Exercise 4.1: Make Amazon Lex 47 Smarter Will my cat be OK outside? Week 4 Wrap Up and Name the city your cat lives in, thanks Resources Tempe 31 I have you see now that I EV is much smarter and feels more human. Voc it just shite out a temperature AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter old text based app and switch us to the realm of voice! Introduction to Week 4 ..we will do service tests as we go. Video: Introduction to Week 4 2 min 4. Steps for creating the lambda messaging proxy. Extending Our App Video: Extending Our Click Services type in lambda in the search box or select Lambda from the History list. App Part 1 Click Create function. Make sure Author from scratch is selected. Video: Extending Our App Part 2 For Function name type in lex_proxy. For Runtime leave it as Node.js 8.10. Video: Adding a Channel For Execution role select Use an existing role. Video: Exercise 4.1 For Existing role select service-role/Get-Weather. Introduction Click View the Get-Weather role on the IAM console. 3 min Click the drop-down arrow next to the Policy name. Reading: Exercise 4.1: Make Amazon Lex Smarter Policy name ▼ 45 min AWSLambdaEdgeExecutionRole-81930794-1b39-4ad1-967a-ec21bce29a78 Week 4 Wrap Up and Resources Click Edit policy Click Add additional permissions. Under Service click Choose a service and search for lex. Select List and Read and for Write choose PostText. AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Read (8 selected) ▼ Write (1 selected) Introduction to Week 4 Video: Introduction to CreateBotVersion ? DeleteBotVersion ? PostContent ? Week 4 CreateIntentVersion ? DeleteIntent ② ✓ PostText ?? CreateSlotTypeVersion ? DeleteIntentVersion ③ PutBot ? Extending Our App DeleteBot ② DeleteSlotType ?? PutBotAlias ? Video: Extending Our App Part 1 DeleteBotAlias ? ■ DeleteSlotTypeVersion ② PutIntent ③ 14 min PutSlotType ?? DeleteBotChannelAssociation ③ DeleteUtterances ③ Video: Extending Our App Part 2 Video: Adding a Channel Under Resources select All resources. Click Review policy. Video: Exercise 4.1 Introduction Click Save changes. 3 min Go back to your Lambda tab. Reading: Exercise 4.1: Make Amazon Lex Click Create function. Smarter Scroll down to Basic settings and set the Timeout to 1 min and 5 sec. 45 min Week 4 Wrap Up and Basic settings Resources Description Memory (MB) Info AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Introduction to Week 4 Paste the following into the index.js tab: Video: Introduction to Week 4 function handler(event, context, callback){ MESSAGE_STR = event.message_str, **Extending Our App** USER_ID_STR = event.user_id_str, 5 AWS = require("aws-sdk"), LEXRUNTIME = {},
BOT_NAME_STR = "WeatherCatBot", Video: Extending Our App Part 1 BOT_ALIAS_STR = "\$LATEST", 9 + sessionAttributes = { 10 11 Video: Extending Our params = {}; 12 App Part 2 13 6 min 14 + AWS.config.update({ region: "us-east-1" 15 16 Video: Adding a Channel 17 6 min 18 LEXRUNTIME = new AWS.LexRuntime(); 19 20 + Video: Exercise 4.1 botAlias: BOT_ALIAS_STR, 21 Introduction 22 botName: BOT_NAME_STR, inputText: MESSAGE STR, 23 3 min 24 userId: USER_ID_STR, sessionAttributes: sessionAttributes 25 Reading: Exercise 4.1: 26 Make Amazon Lex 27 -LEXRUNTIME.postText(params, function(error, data){ Smarter 28 var response = {}; 29 if(error){ 30 console.log(error, error.stack); response = "problem with lex"; 31 Week 4 Wrap Up and 32 callback(null, response); 33 ₹ }else{ Resources 34 console.log(data); 35 response = data; callback(null, response); 36 37 38 }); 39 40 exports.handler = handler; AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter 5. Steps for testing it in the Lambda console Introduction to Week 4 Video: Introduction to Still in our Lambda console. Click Configure Test. Week 4 Leave Event template as Hello World. For Event name type in askwithcity. **Extending Our App** Paste the following code: (the user_id_str is to maintain the session for that user) Video: Extending Our App Part 1 2 "message_str": "can my cat go out in alto?", Video: Extending Our "user_id_str": "10126023" 3 App Part 2 4 6 min Video: Adding a Channel Click Create. 6 min Video: Exercise 4.1 Click Test. Introduction You should see a similar response: Perfect! 3 min Reading: Exercise 4.1: Make Amazon Lex Response: 2 + Smarter "intentName": "CatWeather", 4 + "slots": { "city_str": "alto" Week 4 Wrap Up and "sessionAttributes": { Resources "city_str": "alto", "temp_str": "47" 9 10 "message": "47", 11 "messageFormat": "PlainText", 12 "dialogState": "Fulfilled", 13 14 "slotToElicit": null 15 } AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Click Configure test events. Introduction to Week 4 Change the event: (new user id - bump) Video: Introduction to Week 4 2 min "message_str": "can my cat go out in alto?", Extending Our App 3 "user_id_str": "10126023" 4 Video: Extending Our App Part 1 To Video: Extending Our App Part 2 "message_str": "can my cat go out?", Video: Adding a Channel 3 "user_id_str": "10126024" 6 min 4 Video: Exercise 4.1 Introduction Click Save. 3 min Click Test. Reading: Exercise 4.1: Make Amazon Lex You should see the following output: Smarter Week 4 Wrap Up and Resources "intentName": "CatWeather", 3 + "slots": { "city_str": null "sessionAttributes": {}, "message": "Name the city your cat lives in, thanks", "messageFormat": "PlainText", "dialogState": "ElicitSlot", "slotToElicit": "city str" 10 11 AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Time contractoring arra do not trans to otal tid metrocopio. Introduction to Week 4 Video: Introduction to 2 "message_str": "DENVER", Week 4 "user_id_str": "10126024" 4 **Extending Our App** Video: Extending Our App Part 1 You should see the following output: 14 min Video: Extending Our App Part 2 "intentName": "CatWeather", 3 ▼ "slots": { "city_str": "DENVER" 4 Video: Adding a Channel "sessionAttributes": { "city_str": "DENVER", "temp_str": "38" Video: Exercise 4.1 "message": "38", 10 Introduction "messageFormat": "PlainText",
"dialogState": "Fulfilled", 11 12 3 min "slotToElicit": null 13 14 Reading: Exercise 4.1: Make Amazon Lex Smarter 45 min Since the user_id_str didn't change it will keep that same session open and return the correct data. Week 4 Wrap Up and Resources Awesome, you are nearly done. We now just point API Gateway to this new proxy, disabling the old text (text.html) API. That webpage will no longer work. Once you follow these next few steps you can visit the /voice.html version of the website and test it. AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter function. Introduction to Week 4 Sign in to the AWS Management Console and in the Find Services search box type api and choose Video: Introduction to Week 4 API Gateway or choose API Gateway from the History list. Under APIs click CatWeather. Extending Our App Click POST
 POST Video: Extending Our App Part 1 Click Integration Request. Click the pencil icon next to get_weather Video: Extending Our Type in I and select lex_proxy and click the check to update it. App Part 2 Click OK on the Add Permission to Lambda Function pop-up. Video: Adding a Channel Go back to the root - // under Resources. 6 min Click Actions and Enable CORS. Click DEFAULT 4XX and DEFAULT 5XX. Video: Exercise 4.1 Introduction Click Enable CORS and replace existing CORS headers. Click Yes, replace existing values on the Confirm method changes pop-up. Reading: Exercise 4.1: Make Amazon Lex Click Actions and Deploy API. Choose test for the Deployment stage and click Deploy. Smarter Pull up your CloudFront URL appending /voice.html at the end Week 4 Wrap Up and Resources Ensure you have your microphone enabled if you see a pop up in the browser. Click **Push to talk** and utter a phrase like: "Can my cat go out in Denver?" Your bot will chat back with you :) AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Too cold for my cat? Introduction to Week 4 Video: Introduction to Week 4 can my cat go out in 2 min Extending Our App Ok, so you want to Video: Extending Our know if your cat can go App Part 1 outside today in Denver. Let me check the tubes...one sec Video: Extending Our App Part 2 The weather in denver is 38 degrees. I think this maybe a bit cold for Video: Adding a Channel your cat. Video: Exercise 4.1 Introduction 3 min Reading: Exercise 4.1: Make Amazon Lex Smarter Refresh page to try Week 4 Wrap Up and Resources AWS Fundamentals: Building Serverless Applications > Week 4 > Exercise 4.1: Make Amazon Lex Smarter Congrats you are done. Introduction to Week 4 BEFORE you close this out and head over to the next video, please tear down any applications you no Video: Introduction to Week 4 longer want running. Outside of free tier some of these services are not free.** 2 min Check the forums if you need help tearing down the services. **Extending Our App** Video: Extending Our App Part 1 **Exercise goal checklist** Video: Extending Our App Part 2 Create a simple chatbot using the lex console. Upload our website to S3. Video: Adding a Channel 3. Create a content delivery network and lock down S3. 6 min 4. Build an API gateway mock with CORS. Video: Exercise 4.1 Introduction 5. Build a Lambda mock, use IAM, push logs to CloudWatch. 3 min 6. Create and seed a database with weather data. Reading: Exercise 4.1: Make Amazon Lex 7. Enhance the lambda, so it can query the database. 8. Play with your new text based data driven application. Week 4 Wrap Up and 9. Create a LEX proxy using Lamba. Resources Enhance API gateway to use the LEX proxy. 11. Play with your new voice web application. Mark as completed