

Creating your first Alexa Skill Tutorial

Stevens Software Engineering Club

1. First you need to create an Amazon account. Go to developer.amazon.com and create an Amazon account if you don't already have one. Once signed up, go to the developer console by clicking "Developer Console" in the top right corner.



Services and Technologies

on appstore

on Appstore

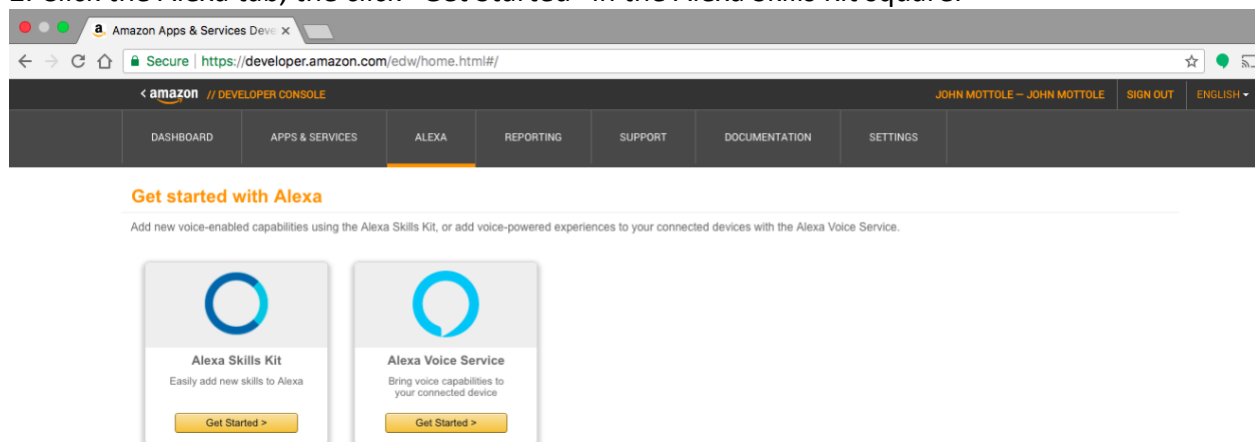
and games for Amazon Fire TV, Fire
mobile platforms



Amazon Web Services

Reliable, scalable, and inexpensive cloud computing services

2. Click the Alexa tab, then click "Get Started" in the Alexa Skills Kit square.



3. Click “Add a New Skill” to create a new skill

amazon

// DEVELOPER CONSOLE

JOHN MOTTOLE -- JOHN MOTTOLE

SIGN OUT

ENGLISH

DASHBOARD

APPS & SERVICES

ALEXA

REPORTING

SUPPORT

DOCUMENTATION

SETTINGS

Building Alexa Skills with the Alexa Skills Kit

Add a New Skill

To learn more about building Alexa skills, see [Getting Started with the Alexa Skills Kit](#). To start building an Alexa skill for free using AWS Lambda, see [Creating an AWS Lambda Function for a Custom Skill](#).

We encourage you to visit the [Alexa Developer Forum](#) to collaborate with Alexa team members and fellow Alexa developers.

4. This is where we will set up our new skill. On this page we can keep most of the settings the way they are, we just need to give it a name and an invocation name. The name will be the official name in the Alexa Skills Store. The invocation name is the name that you will use when talking to Alexa. We are going to put “Hello World” for both of these. Hit save and click on the Interaction Model tab.

Amazon Apps & Services Dev: x

Secure | https://developer.amazon.com/edw/home.html#/skill/create/

Skill Information

Interaction Model

Configuration

SSL Certificate

Test

Publishing Information

Privacy & Compliance

Skill Type

Define a custom interaction model or use one of the predefined skill APIs. [Learn more](#)

Custom Interaction Model

Smart Home Skill API

Flash Briefing Skill API

Video Skill API

Language

Language of your skill

English (U.S.)

Name

Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.

Invocation Name

The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".

For successful Alexa Skills Certification, please review and follow our [Invocation Name Guidelines](#) as well as our [Certification Requirements](#).

Global Fields

These fields apply to all languages supported by the skill.

Audio Player

Does this skill use the audio player directives?

No

Yes

[Learn more](#)

Video App

Does this skill use the video app directives? [Learn more](#)

No

Yes

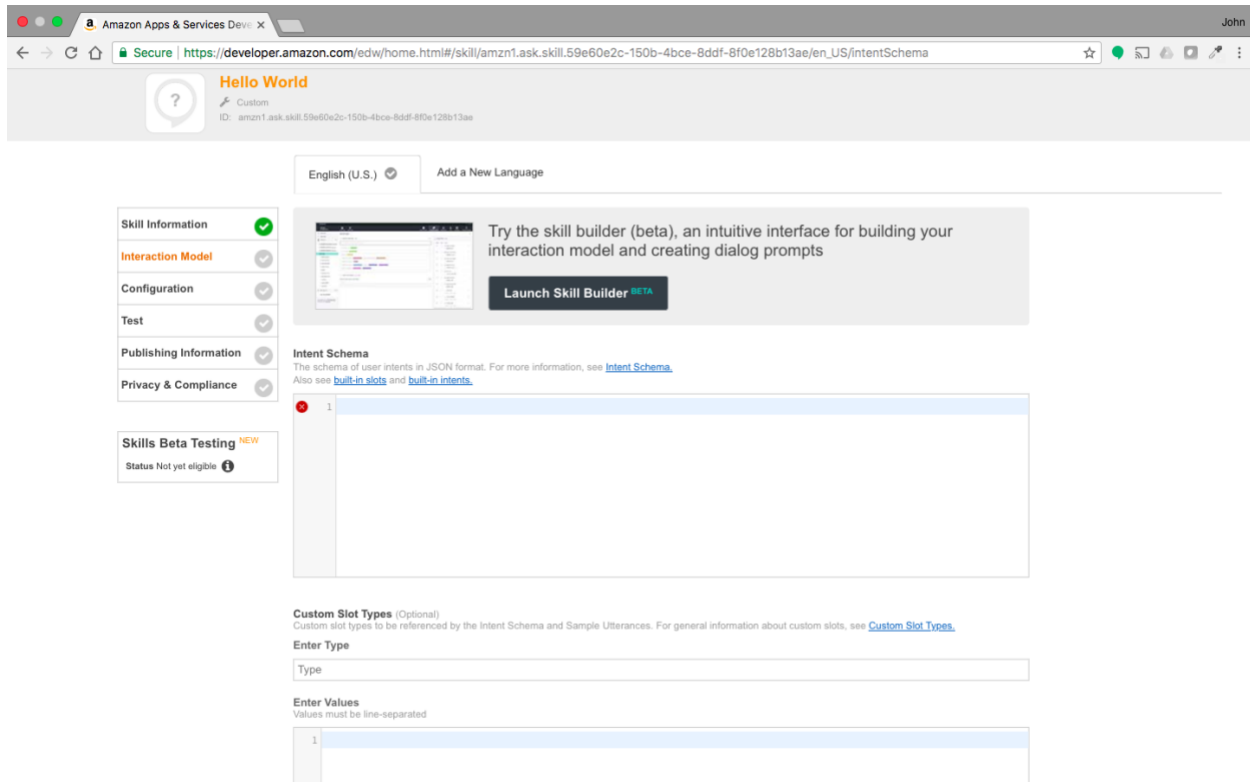
Render Template

Does this skill use the Render Template directives? [Learn more](#)

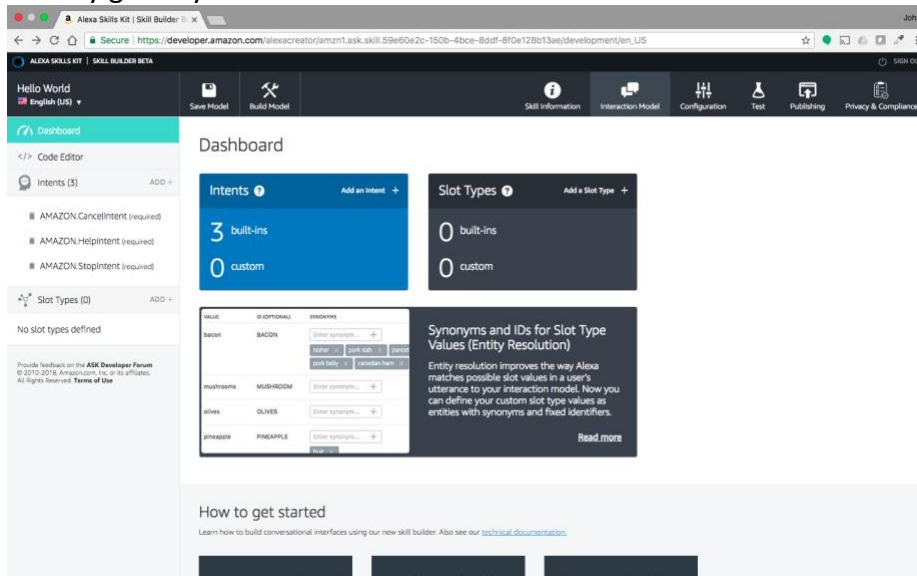
No

Yes

5. Right now the skill builder is in beta mode but will be the standard soon, it is stable enough to use so we will build our skill in this. Click “Launch Skill Builder”.



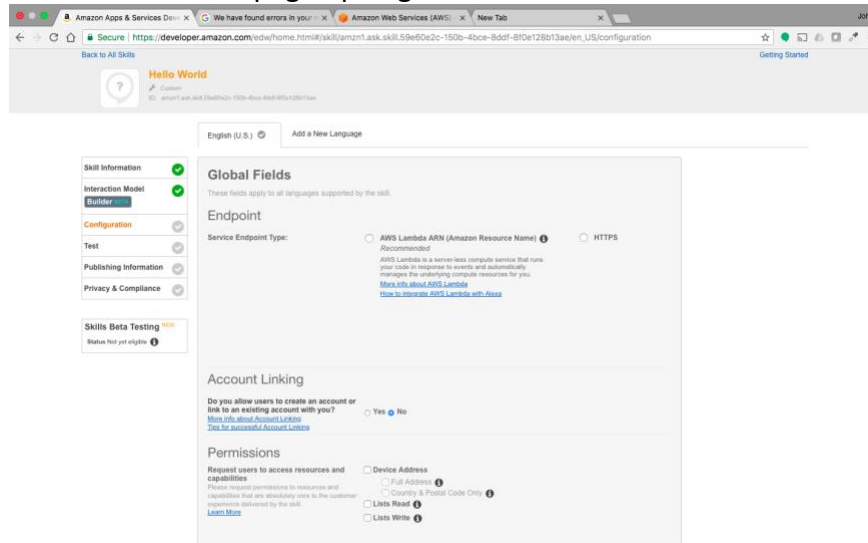
4. On the left side of the skill builder is our intents. Intents can be thought of as functions. They are different actions that users tell your skill to do. Right now there are 3 that are pre-defined by Amazon. We are going to add two more, one that has our skill say hello and one that will say goodbye.



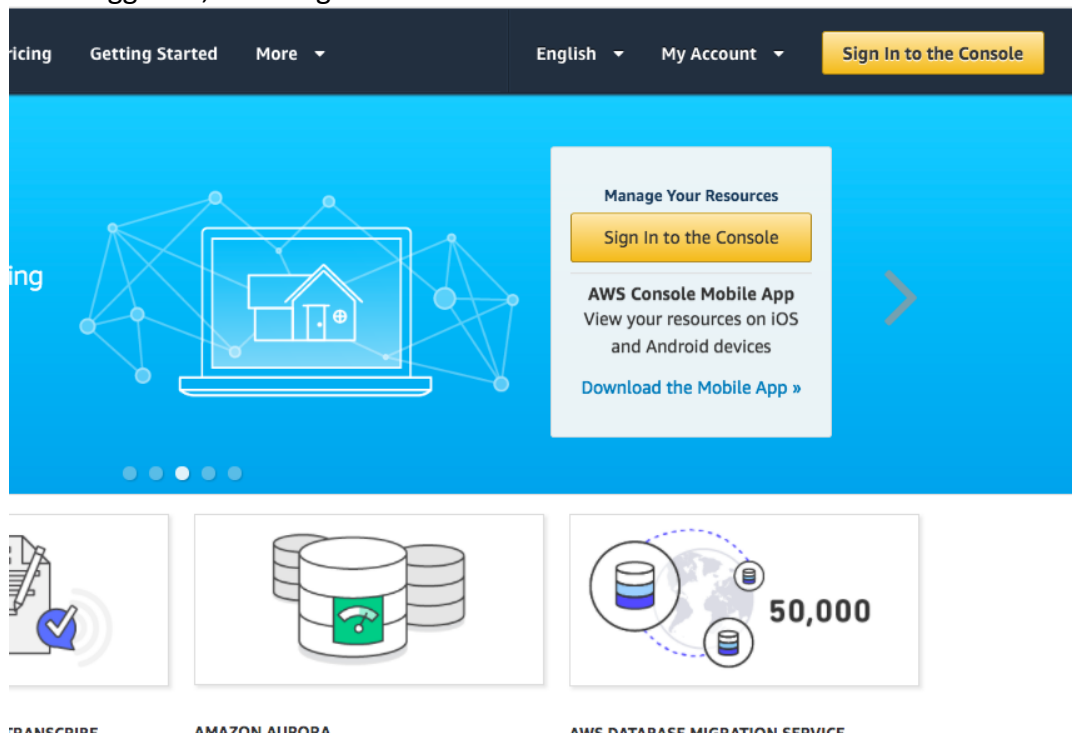
5. Click “Add an intent” on the dark blue Intent box. Name this one “SayHello” with no spaces and click create. We now have to enter sample utterances. These are potential things that users could say to our skill to make Alexa do something. Type “Hi” in the box and press enter. This will add an utterance. People say hi in more than one way, so let’s repeat this with “hello” and “what’s up”. Now that we defined our “SayHello” intent, lets create one for Alexa to say goodbye. Click the “Add” button on the Intents section on the left and repeat the process, this time call it “SayGoodbye” and make the sample utterances things like, “Bye”, “Goodbye” and “So long”.

The screenshot displays the Alexa Skills Kit Skill Builder interface in a web browser. The top navigation bar includes the 'Hello World' skill name, a language dropdown set to 'English (US)', and buttons for 'Save Model', 'Build Model', 'Skill Information', 'Interaction Model', 'Configuration', 'Test', 'Publishing', and 'Privacy & Compliance'. The left sidebar shows a list of intents: 'Intents (5)' with an 'ADD +' button, followed by 'AMAZON.CancelIntent (required)', 'AMAZON.HelpIntent (required)', 'AMAZON.StopIntent (required)', 'SayGoodbye' (highlighted in teal), 'SayHello', and 'Slot Types (0)' with an 'ADD +' button. The main content area is titled 'SayGoodbye' and contains three sections: 'Sample Utterances (3)' with a search icon and a list of utterances ('So long', 'Goodbye', 'Bye') each with a delete icon; 'Intent confirmation (optional)' with a toggle switch set to 'NO' and the question 'Does this intent require confirmation?'; and 'Prompts' with a text input field containing 'What will Alexa say to ask the user to confirm the intent?'. On the right, the 'Intent Slots (0)' section shows 'No slots defined' and a 'Create a new slot...' button with an 'Add' button next to it. The bottom of the page includes a footer with feedback information and copyright details.

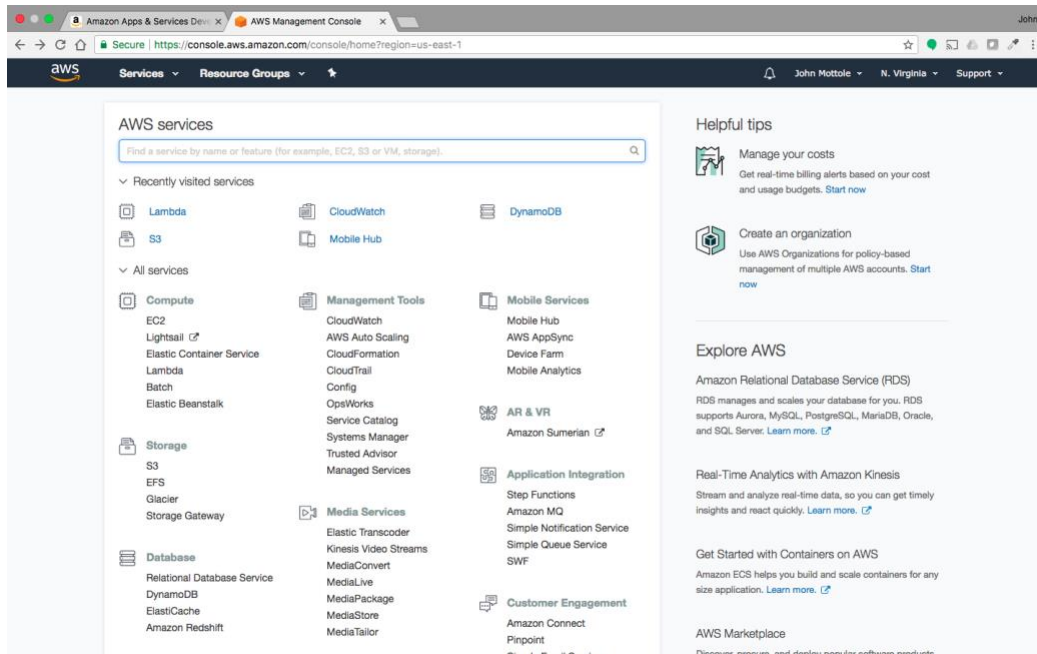
6. Now let's save and build our model that we have just created. Click the "Save Model" button on the top then press "Build Model". This will take a little while, do not refresh the page. Once the build is completed, click on the configuration tab towards the top right corner. We cannot complete this page until we have a Lambda function set up in Amazon Web Services (AWS). The lambda function will act as the logic behind our skill that will tell Alexa what to say to the user. For now leave this page open go to aws.amazon.com in a new tab.



7. If you do not have an AWS account, sign up for one. It is a different from an Amazon account. Once logged in, click "Sign in to the console".



8. This will bring you to list of all the services that AWS offers. At the top right corner, next to your name, there should be a location. Click the drop down and change it to N.Virgina. Under the compute section click Lambda.



9. Click the orange create function button to create our lambda function. We can name this HelloWorld. For runtime leave it as Node.js 6.10. For role choose “Create a new role from template”, make the role name “MyRole” and for policy templates choose “Basic Edge Lambda Permissions”. Click “Create Function” when done.

The screenshot shows the AWS Lambda Management Console 'Create function' page. The 'Author from scratch' tab is selected. The form fields are as follows:

- Name***: HelloWorld
- Runtime***: Node.js 6.10
- Role***: Create new role from template(s)
- Role name***: MyRole
- Policy templates**: Basic Edge Lambda permissions

The 'Create function' button is orange and located at the bottom right of the form.

10. On the left side under Add Triggers, click “Alexa Skills Kit”. It will ask you for a skill ID. This is from your Alexa developer page. The ID can be found under the name of the skill (Hello World). After copying the ID click add, then press save at the top.

The screenshot shows the Alexa Skills Kit configuration page for the skill 'Hello World'. The 'Global Fields' section is visible, showing the 'Endpoint' configuration with 'Service Endpoint Type' set to 'AWS Lambda ARN (Amazon Resource Name)'. The skill ID is displayed as amzn1.ask.skill.59e60e2c-150b-4bce-8ddf-8f0e128b13ae.

Global Fields

These fields apply to all languages supported by the skill.

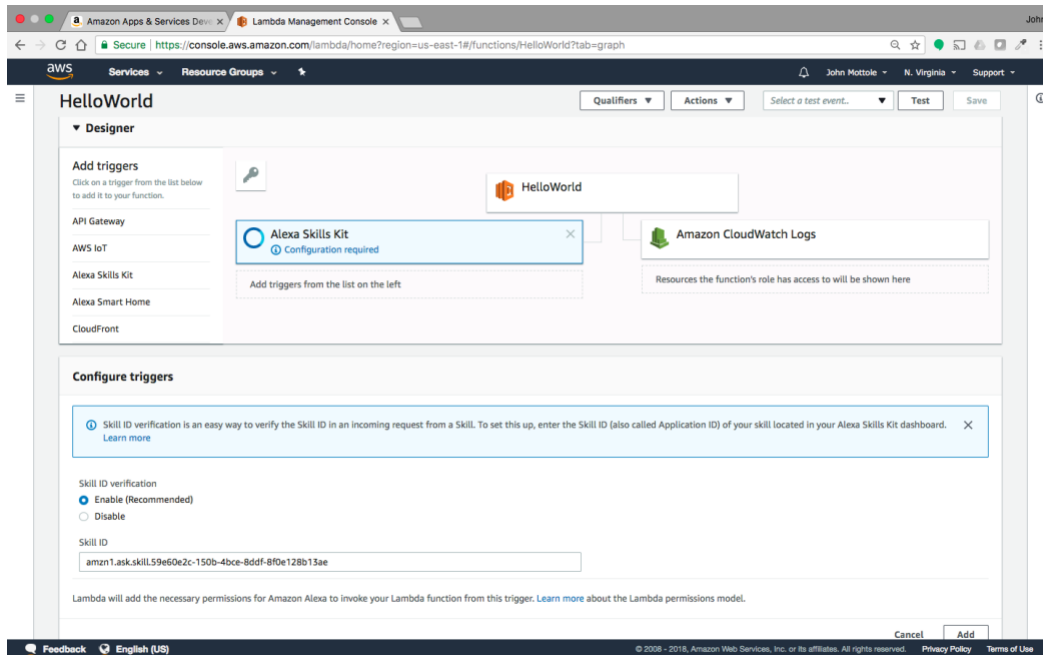
Endpoint

Service Endpoint Type:

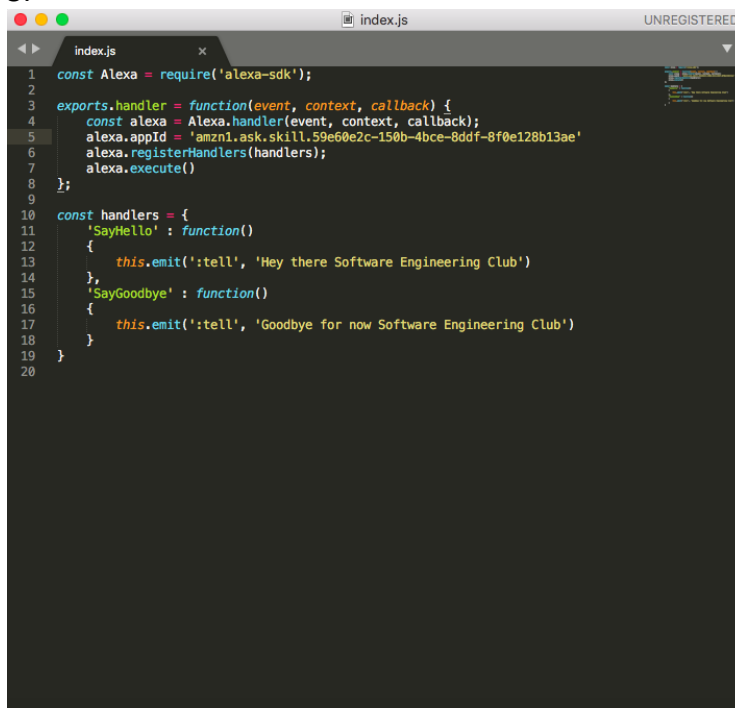
- ☐ AWS Lambda ARN (Amazon Resource Name) **Recommended**

AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.

[More info about AWS Lambda](#)



11. Click on the white “HelloWorld” rectangle on the tree, this will bring us back to the code of our function. There is the ability to write code for lambda functions directly in the browser but for this we will writing code in a text editor and zipping it up to send to AWS. To start create a folder for this project. Open your favorite text editor and copy the code in the index.js file on our github. You will need to change the appId located on line 5 to the same ID that you copy and pasted in step 10. The names of the functions in the handlers, ‘SayHello’ and ‘SayGoodbye’, can stay the same as long as you used the ‘SayHello’ and ‘SayGoodbye’ intent names from step 5.



12. Save this file as “index.js” in your project folder. Open up a terminal and change the directory to your folder. My folder is in Desktop/SEC/Github/Alexa, so to change directories in the terminal I’m going to write “cd Desktop/SEC/Github/Alexa”. The next step will require Node JS, if you do not have it you can install it at <https://nodejs.org/en/download/>.

```
Alexa — -bash — 80x24
Last login: Fri Feb  9 18:05:21 on ttys000
[Johns-MacBook-Air:~ John$ cd Desktop/SEC/Github/Alexa/
Johns-MacBook-Air:Alexa John$
```

13. Type “npm init” into terminal, keep pressing enter to accept the suggestions from npm. After npm asks if everything is ok and you have pressed enter, type “npm install --save alexa-sdk”.

```
Alexa — -bash — 90x49
Last login: Fri Feb  9 18:05:21 on ttys000
[Johns-MacBook-Air:~ John$ cd Desktop/SEC/Github/Alexa/
Johns-MacBook-Air:Alexa John$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

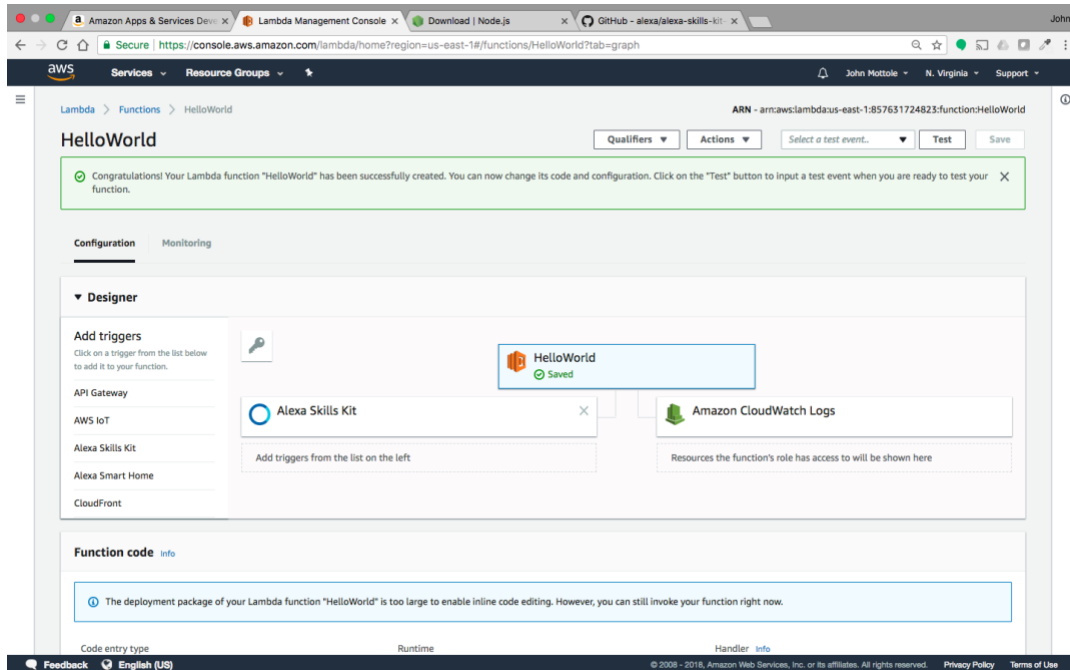
Press ^C at any time to quit.
[package name: (alex)
[version: (1.0.0)
[description:
[entry point: (index.js)
[test command:
[git repository:
[keywords:
[author:
[license: (ISC)
About to write to /Users/John/Desktop/SEC/Github/Alexa/package.json:

{
  "name": "alex",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

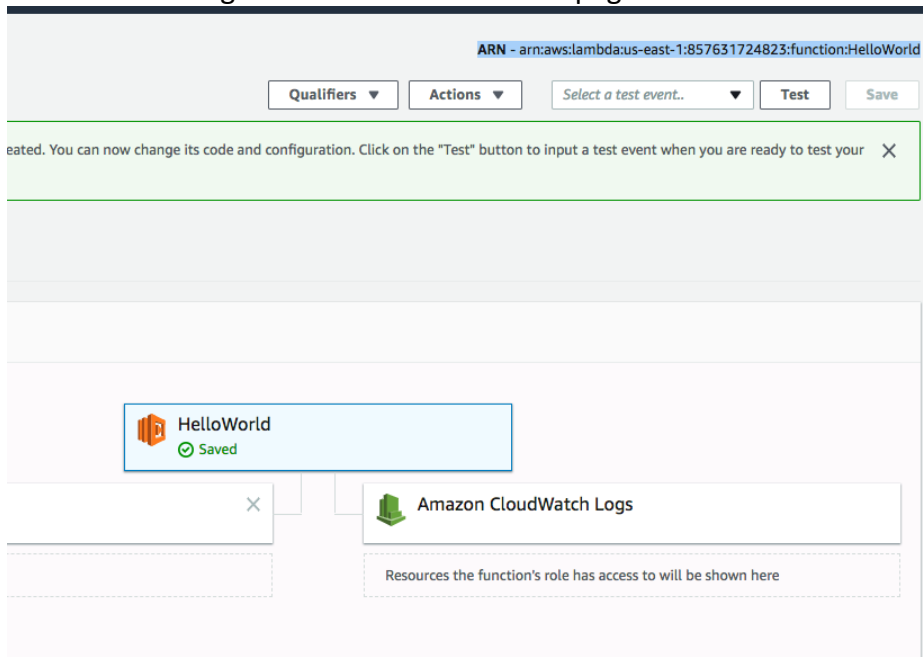
[Is this ok? (yes)
[Johns-MacBook-Air:Alexa John$ npm install --save alexa-sdk
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN alexa@1.0.0 No description
npm WARN alexa@1.0.0 No repository field.

added 18 packages in 6.891s
Johns-MacBook-Air:Alexa John$
```

14. Zip up the index.js file, package.json file and node_modules folder. Do not zip up the entire project folder, you will need to highlight these 3 files and zip them. After you have your zip file go back to AWS and choose "Upload a .Zip file" under the "Code entry type" drop down in the "Function code" block. Click the upload button and select your zip file. Click save on the top right corner.



15. Our lambda function is now ready. Go back to the Alexa developer page for your skill and make sure you are on the configuration tab. Choose AWS Lambda RN and enter the ARN located on the right of the lambda function page. Press next when this is filled out.



16. In the test section select “Go To Test Simulator” to test your skill. Type “Tell Hello World hi”, if everything worked it should say “Hey there Software Engineering Club”. Type “tell hello world goodbye” and it should say “Goodbye for now Software Engineering Club”. If you have an Alexa device registered with your Amazon account, your new skill will automatically be enabled on your device.

The screenshot shows the Amazon Alexa Developer Console's Test Simulator. The browser tabs include 'Amazon Apps & Services', 'Lambda Management Console', 'Download | Node.js', and 'GitHub - alexa/alexa-skills-kit'. The URL is https://developer.amazon.com/alexa/console/simulator/edw/amzn1.skill.59e60e2c-150b-4bce-8ddf-8f0e128b13ae/development/en_US/. The interface has a top bar with 'Test is enabled for this skill' and checkboxes for 'Skill I/O', 'Echo Show Display', and 'Device Log'. Below this are tabs for 'Alexa Simulator', 'Manual JSON', and 'Voice & Tone'. The 'Alexa Simulator' tab is active, showing a conversation history on the left and a 'Skill I/O' section on the right. The conversation history includes: 'tell hello world hi' (user input) leading to 'Hey there Software Engineering Club' (Alexa response); 'Tell hello world bye' (user input) leading to 'Sorry, I don't know that.' (Alexa response); and 'tell hello world goodbye' (user input) leading to 'Goodbye for now Software Engineering Club' (Alexa response). The 'Skill I/O' section shows the JSON input and output for the last interaction. The JSON input is a standard Alexa request with session, application, user, and context information. The JSON output is a response with an outputSpeech object containing the text 'Goodbye for now Software Engineering Club' and a sessionAttributes object.

Test is enabled for this skill

Skill I/O

Echo Show Display

Device Log

Alexa Simulator

Manual JSON

Voice & Tone

English (US)

Type or click and hold the mic

Goodbye for now Software Engineering Club

Hey there Software Engineering Club

Sorry, I don't know that.

Goodbye for now Software Engineering Club

JSON Input

```
1- {
2  "version": "1.0",
3  "session": {
4    "new": true,
5    "sessionId": "amzn1.echo-api.session.3aed7a32-a9-
6    "application": {
7      "applicationId": "amzn1.ask.skill.59e60e2c-1-
8    },
9    "user": {
10     "userId": "amzn1.ask.account.AF6XN7INIKHYQBL
11   },
12   "context": {
13     "AudioPlayer": {
14       "playerActivity": "IDLE"
15     },
16     "Display": {},
17     "System": {
18       "application": {
19         "applicationId": "amzn1.ask.skill.59e60e
20       },
21       "user": {
22         "userId": "amzn1.ask.account.AF6XN7INIKH
23       },
24       "device": {
25         "deviceId": "amzn1.ask.device.AEGHZLRNZW
26       },
27       "supportedInterfaces": {
28         "AudioPlayer": {},
29         "Display": {
30           "templateVersion": "1.0",
31           "markupVersion": "1.0"
32         }
33       }
34     }
35   }
36 }
```

JSON Output

```
1- {
2  "body": {
3    "version": "1.0",
4    "response": {
5      "outputSpeech": {
6        "type": "SSML",
7        "ssml": "<speak> Goodbye for now Softwa
8      },
9      "shouldEndSession": true
10    },
11    "sessionAttributes": {},
12    "userAgent": "ask-nodejs/1.0.25 Node/v6.10.3"
13  }
14 }
```

© 2010-2018, Amazon.com, Inc. or its affiliates. All Rights Reserved. Alexa Developer Blog Alexa Skills Kit

This is only the surface of what Alexa can do. There are a lot of resources online with all of the things that can be done with Alexa. <https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs> is a good start for using the Alexa SDK.