Build Your First Skill for Amazon Alexa

Objective: Get yourself ready to host this workshop on creating a skill with Amazon's Alexa.

About this Workshop

During this workshop, you'll teach participants about Voice User Interfaces, Amazon's Alexa voice service, Amazon Lambda functions, and ultimately - how to create a new Alexa Skill.

The skill that you'll create is an "About Me" skill that returns random facts about the author whenever it is invoked.

How to Prepare

Prepare for this workshop by familiarizing yourself with the slides and this instructor guide. During the workshop, you'll be able to use this guide as a reference for what to say and how to direct participants.

For this workshop specifically, you should:

- **Sign up for an Amazon Developer account and an AWS account** You'll need both accounts in order to create Alexa Services and AWS Lambda functions.
- Follow the slides and create your own "About Me" skill. This will help you prepare for participant questions, and gives you a working example that you can demo during the workshop.
- Read over this manual 2-3 times so that you are familiar with the material

---

Introduction to Alexa and Voice User Interfaces

Activity: Ask participants to raise their hand if they have used Alexa, Google Home, or Siri before. This will give you a good idea about how familiar your audience is with Voice Interfaces. Ask participants to keep their hands up if they have an Amazon Echo device. This will help you narrow in on how many have experience with Alexa.

What is Alexa?

Alexa and similar Voice User Interfaces (VUIs) like Google Home or Siri enable you to *speak* commands rather than type them on a keyboard or swipe a touchscreen. It's a hands-free way to interact with computers and control IoT enabled devices.

Alexa is Amazon's voice service that listens to spoken input and uses it to execute computing tasks or skills in the cloud, and then returns output -- just like a JavaScript function. When you talk to Alexa, you are communicating with a cloud-based service. Although it might seem like a conversation, you are actually issuing commands to a web service and receiving a response from a database. There is simply another step on both ends: your voice needs to be translated to a command, and the information retrieved from the database needs to be converted back to voice.

The name “Alexa” was chosen for the service both to harken back to the Library of Alexandria that attempted to collect all the world’s knowledge. As well, the “x” in the name “Alexa” is an uncommon sound, which decreases the likelihood that Alexa will be woken up by accident.

Why do VUIs matter?

VUIs (Voice User Interfaces) signify a big shift in how we interact with technology. Instead of using our fingers to control a computer or a machine, we can physically separate ourselves from devices and speak naturally - the same way we would communicate with another person. This affords the ability for people to interact with technology while their hands are unavailable, such as when cooking, or fixing a tire. Where else can participants imagine VUIs being integrated into the world around them?

Why Build With Alexa

Alexa is programmable, so you can connect it to webapps, other APIs, physical devices and more. You could build smart remotes for your TV, check your bank balance, or start an order at Starbucks. If any participants already use Alexa, ask them specifically what skills they already use.

There are some big benefits for developers to build with Alexa. By putting your code in the cloud and connecting it to Alexa, there are *no servers to maintain or provision*, so you can build without worrying about infrastructure or scaling. This adds greater efficiency and reliability to a business. When developers build with Alexa, they have a *wide reach* to millions of customers who are already familiar with and primed to use VUIs, and in turn a large potential market. As well, with each command Alexa has access to more *training data*, making its ability to recognize new tones and voices ever better.

Alexa skills are made of two parts: a front-end and a back-end. The **front-end**, called the Alexa Voice UI, handles text to speech and converts the audio into text commands that the back-end can analyze. The **back-end**, called AWS Lambda, is the logic code that powers the app. It takes the parsed speech and retrieves the information or service, then this information is passed back to the front end to be said vocally to the user. AWS Lambda is an event-driven computing platform. Lambda runs when triggered by an event and executes code that has been loaded into the system.

Activity: Ask participants whether they have used Alexa's Skills before. Have they found any obscure skills that Alexa offers, or do they have ideas of what kinds of skills they would want built? Give participants time to talk, and to try out Alexa (or echo.sim if you do not have access to an Alexa device) before they build their skill.

Speech Recognition Technology

Speech recognition technology is difficult to get right. When one person says "*fo□r tē tīmz.* " it could mean "forty times" or "for tea time". Context, and human comprehension, is usually needed to understand what the speaker meant, and in turn what the response should be. **Training data** is used to give Alexa several potential phrases that a person might use to trigger one **Intent**. Intent in this context means a vocal command that is translated into an action for Alexa to perform.

For example, if a person's Alexa is linked to their Spotify account, and they say "Play a Childish Gambino song" Alexa first has to process the words mean "play a specific song" and then connect to the Spotify app, look up the specific song, and then play it. To do this,

Alexa needs to first convert the audio to text, then understand from the text what command to run.

All together, the progression moves from a user issuing an Intent verbally, Alexa translates these words into code, Alexa retrieves the data or service referenced in the command, and then Alexa translates that data into an audio output to give back to the user.

Building an Alexa Skill

Objective: Demonstrate & Build an Alexa Skill

Demo: About You Skill

During this workshop, participants will build an "About You" skill that will make it so they can ask Alexa to repeat random facts about themselves.

By building this skill participants will learn how to create a VUI, how Alexa processes voice commands, how to deploy code to AWS Lambda, and how to publish their skills so other people can use it.

Use this opportunity to demonstrate the skill (using the one you created about yourself) to the participants. You can do this from the simulator (https://echosim.io) or from your own Echo device.

Do any of the participants know some sci-fi movies where voice technology is used? Getting participants to think about some of the potential uses of voice tech can help get them excited as they build their own skill.

Tools

The only requirements for this workshop are an Amazon Developer Account and an AWS account. However, some students may prefer to do their development offline, using their own text editor. In this case, they would need to install Node.js and the Alexa Skills kit.

Building About You

There are 5 steps to build and publish the About You skill. During the first three steps, participants will setup the main components and connect them together. During step 4, they will test that their skill works, and in Step 5 they will publish their skill.

For this workshop, participants will modify an existing skill template and copy/paste some json & text data into form fields. Participants will not write a lot of code, but focus on understanding the steps involved in creating a skill.

The steps include:

1. Create an Alexa Skill in the Developer Portal
2. Write your skill code as an AWS Lambda function
3. Connect Your Lambda Function to Your Skill
4. Test your Skill
5. Publish Your Skill

To begin to create a skill, participants need to sign in to their AWS console. Navigate to: mlhlocal.host/alexa-aws. If participants have not signed up for an AWS account, now is a good time to do that. Make sure participants as they create their account **set their location to U.S. East (N. Virginia)**. This will be important down the road as this location allows participants access to the Alexa Skills Toolkits.

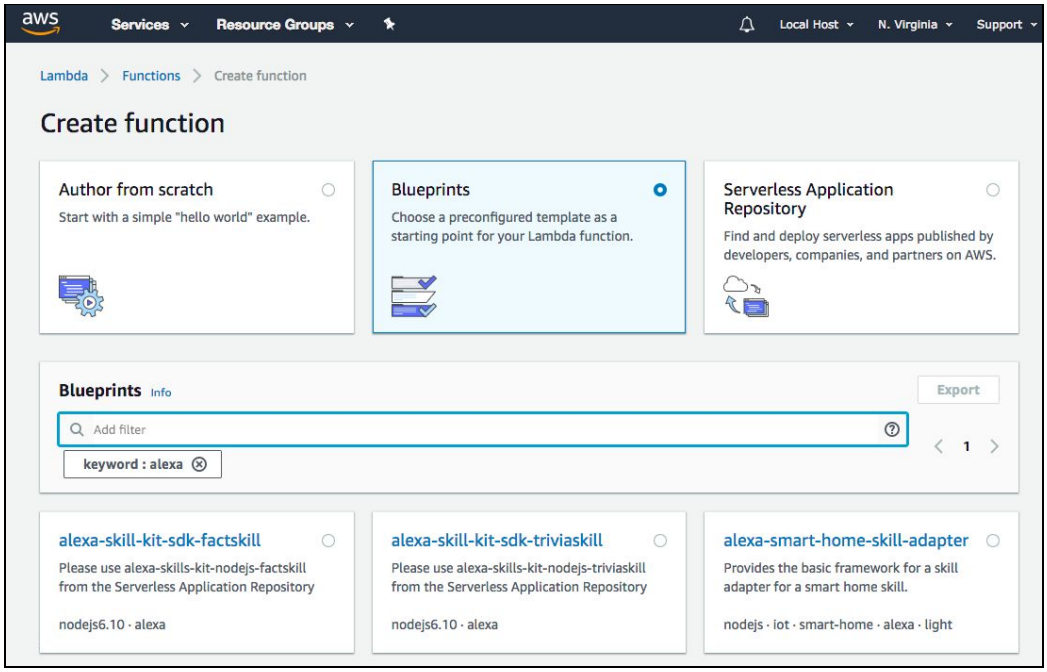
Alexa Skill vs. Lambda Function

Alexa Skills define how the user will interact with the Voice User Interface(VUI), while Lambda functions contain the input/output logic. Participants will first build a Skill, build

the Voice User Interface, and then create the Lambda function. Both Voice Interfaces and Lambda function are hosted in the cloud so there are no development operation issues.

Create A Lambda Function

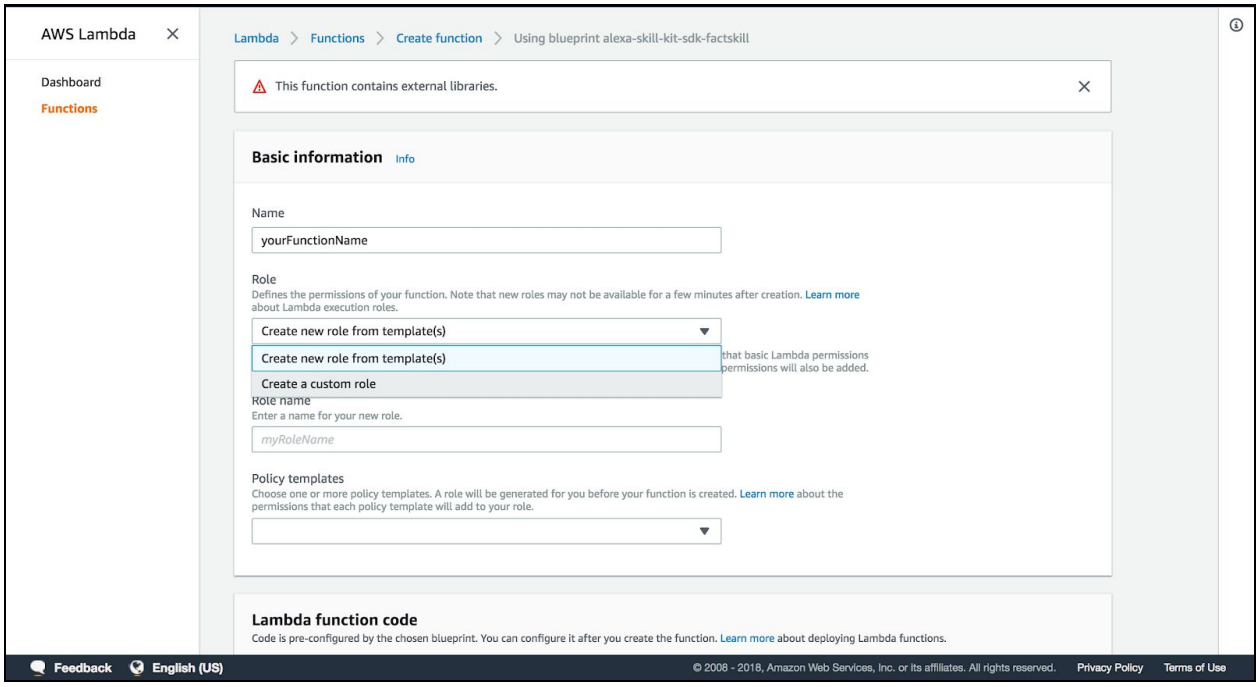
Participants can find the Lambda function either through the Search bar or through the list of available services. Once on the AWS Lambda function page, Select **Create a Function**. On this page, participants should make sure to select “Blueprints” as the type of Skill that they will create, and “alexa-skill-kit-sdk-factskill” as the type of library that participants will pull commands from.



On the next page, participants should select the option of the “alexa-skill-kit-sdk-factskill” then press **Configure.**

Create Your IAM Role

On the next page participants should add a Name for their function and select “Create a Custom Role.”



When you select “Create a Custom Role” a new tab should open. Select “Create New IAM Role” from the dropdown menu and Select **Allow** on the Bottom Right corner. “Role Name” and “Policy” will automatically populate. IAM Roles tells Amazon which services can access which information. This new role enables our Lambda function to execute.

Participants should be automatically be redirected back to “Lambda_basic_execution” will be in the Existing Role field.

Take time to make sure that participants have all gotten to this point.

Create Function

Click **Create Function** at the bottom of your page. Then head over to

mlhlocal.host/alexa-code to download the sample code. Unzip the file and run `src/index.js` in your favorite text editor. Walk around and make sure everyone can open the file.

Within the file, find the `FACTS` array. This array will have MLH facts in it. Change the MLH facts to facts about yourself. Give participants 5-10 minutes to do this. Remind participants

that they need to keep their facts in quotation marks, and every line has to end with a comma except for the last one.

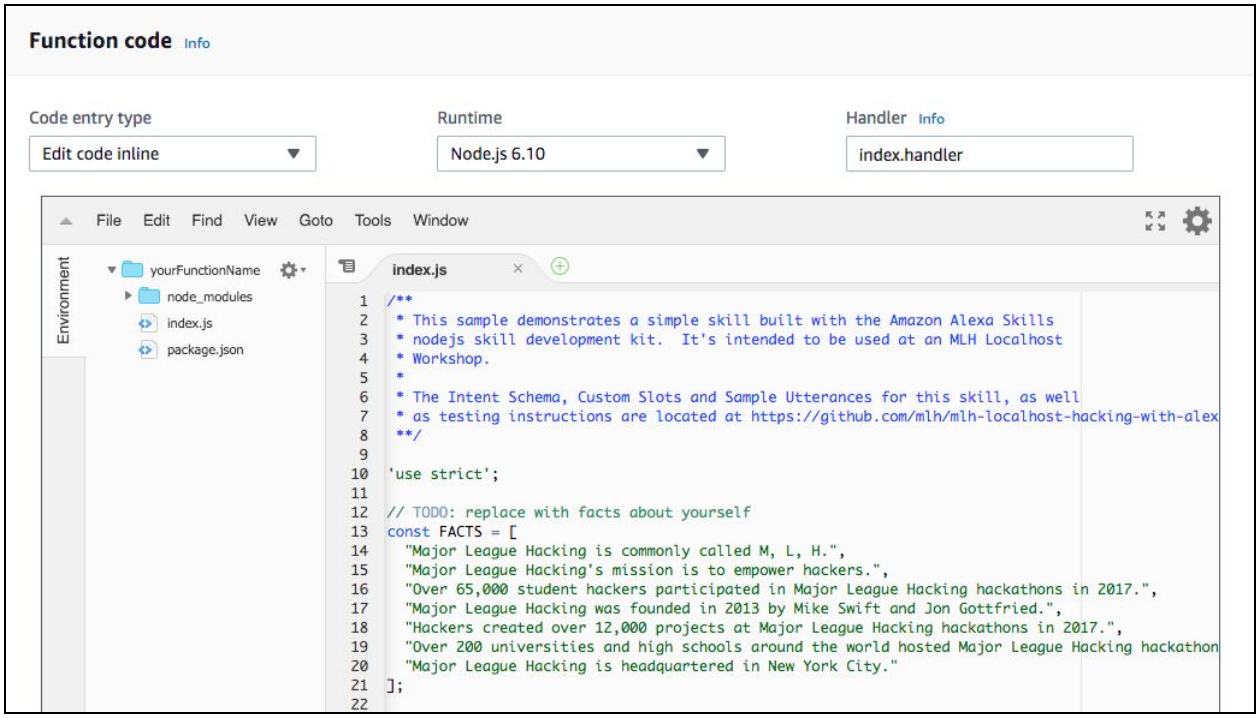
The Handlers Object

The **handlers** object tells Alexa how to handle various actions. In this case, `handlers = { }` tells Alexa that if it hears any of the intents, `'LaunchRequest'`, `'GetNewFactIntent'`, to trigger the `'GetFact'` function. The `'GetFact'` function first Selects a random element from the Facts list. Alexa then prepends 'Here's your fact' to the beginning of the fact, then says the fact out loud by emitting a `":tellWithCard"` event.

The **handler function** tells Alexa to route voice commands by passing a copy of the handlers object. It passes our specific code to the Alexa SDK, where a new Alexa object is created and passed to the parsed speech the users gave to Alexa. We created the list of replies Alexa can pull from (our list of facts) that Alexa can pull from.

Lambda Editor

If there is already text in AWS's code editor, delete it. Copy and paste the entire contents of `index.js` into the inline editor of the AWS Console. At the top of the code editor, keep **Code entry type** as "Edit code inline," **Runtime** as "Node.js 6.10", and **Handler** as "index.handler." Double check to make sure there is not any placeholder code left, and that the entirety of `index.js` is pasted into the code editor.



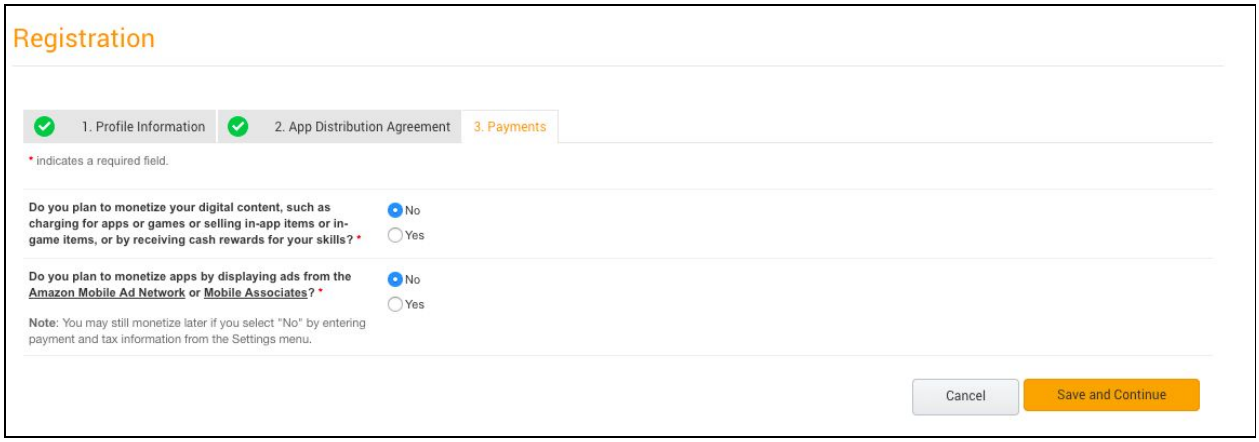
When finished, press **Save**. Do not press **Test**.

Create Amazon Developer Account

This next section will focus on connecting the code that participants have with the Alexa platform that can run the skill. In another tab, navigate to: mlhlocal.host/alexa-portal. Participants will now need to create a developer account. In emails introducing the workshop, encourage participants to set up developer accounts before the event. If they have, skip this section. If not, we will walk through the account creation process.

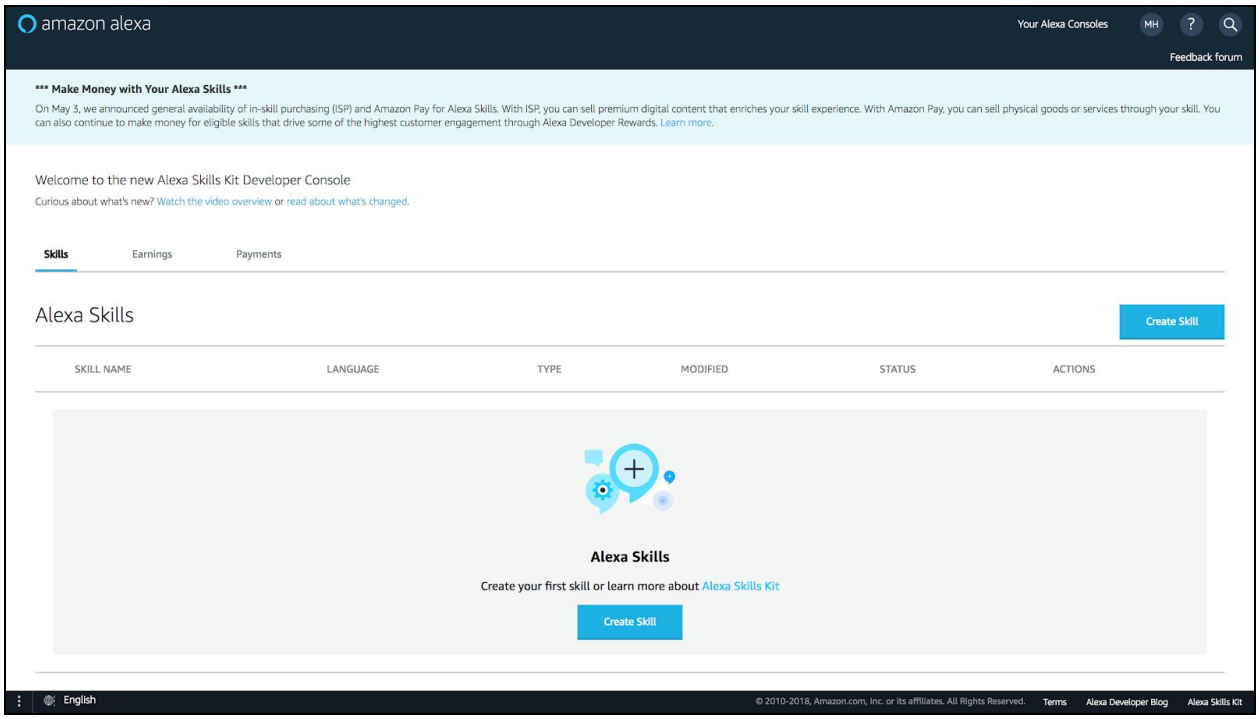
Participants should fill in the account creation form with their name, email, and a password. After the initial account is created,participants will need to create their developer profile and agree to the “Terms of Use.”

Participants will be asked on the next screen whether they will monetize their app or display ads. For both questions participants should Select **No**.



Create New Alexa Skill

Now that participants' developer accounts are set up, they should navigate to the **Alexa** tab. Find the **Alexa Skills Kit** tab. Then Select **Get Started**. They will be brought to a page that lists their existing Alexa skills. Select the button **Create Skill**.



Do any participants have screens that look different? Amazon recently released a new version of the Skills interface. To better follow along with the rest of the steps, we encourage participants to opt out of the newer interface and use the original interface.

Participants will be asked to name their skill. They should title their skill with their Name. On the following page they should Select that they are building a **Custom skill.**

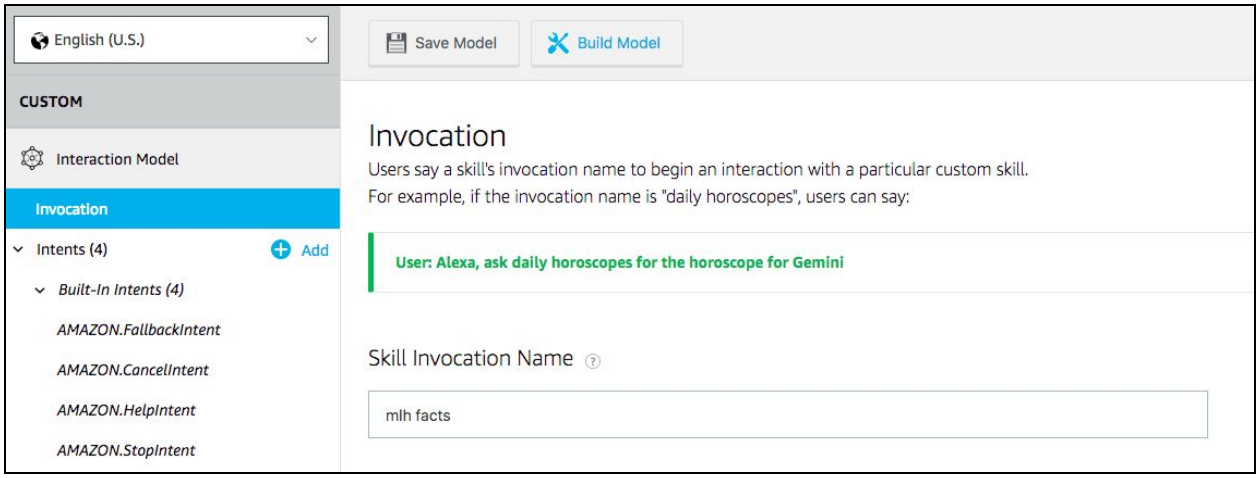
Invocation Names

An **Invocation Name** is a one to three word phrase that when said out loud, launches a skill on Alexa. An Invocation Name has three parts: a Wake word, a Launch word, and a Skill Invocation Name.

- The **Wake word** signals to Alexa to start listening.
- The **Launch word** cues Alexa to launch a program
- The **Skill Invocation** name says the specific program for Alexa to launch

In our example we tell Alexa to launch the **MLH Facts program**. Out loud, it would sound like "**Alexa**, **launch** **MLH facts**."

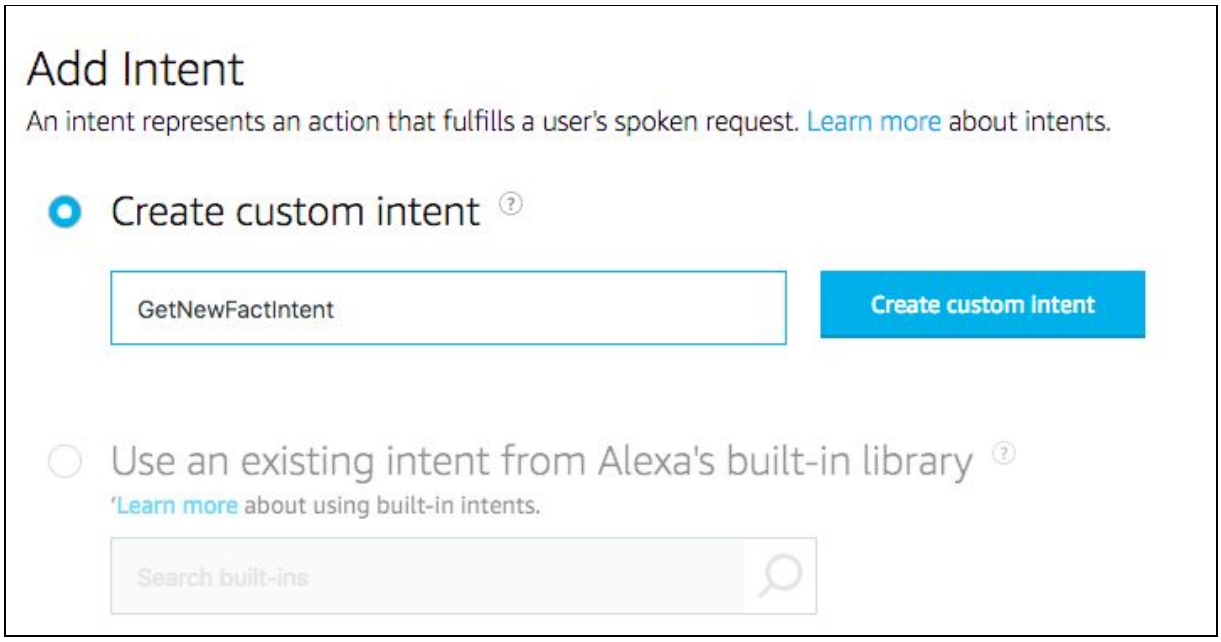
In the Skill Invocation Name section, put something recognizable as yours on the Alexa App. A good choice is your full name and the word "facts" after it.



Make sure that "Customer Interaction Model" is checked for Skill Type.

Add An Intent

After participants choose a name for their skill, they need to configure it. Make sure "Customer Interaction Model" is chosen for Skill Type. In the field "Create Custom Intent" type in "GetNewFactIntent" and then press **Create Custom Intent**.



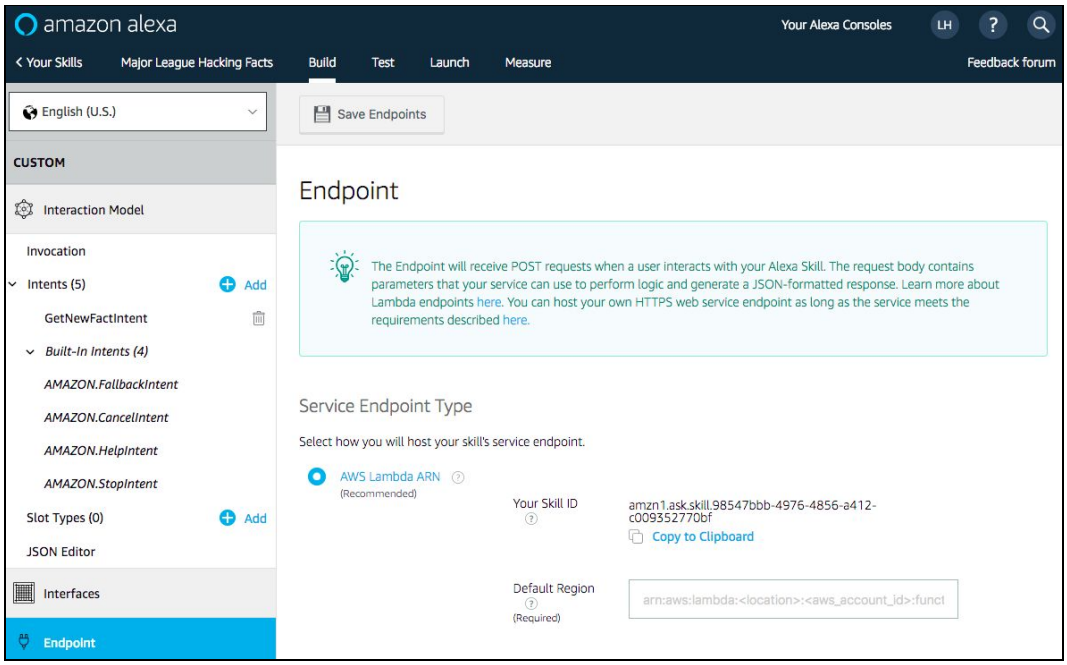
Customize Your Utterances

In the next step, participants will need to create Sample Utterances that provide different cues to signal Alexa to start and retrieve facts about themselves. Instead of starting from scratch, MLH has provided a text file that participants can use as a template. Open `SpeechAssets/SampleUtterances.txt` (it's in the `SpeechAssets.zip` folder participants downloaded earlier) and change everywhere participants see "Major League Hacking" to their Invocation name.

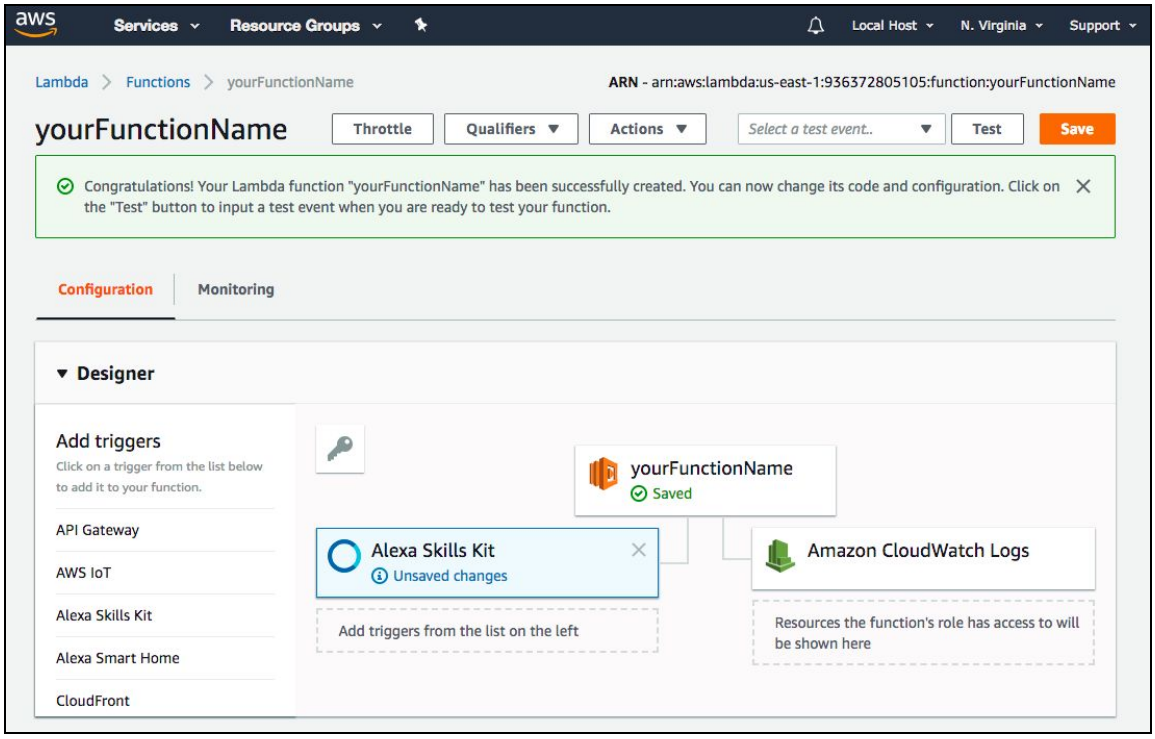
Once participants have changed the wording in their file to reflect their Invocation Name, they should paste each line one by one into the **Sample Utterances** fields. When participants have finished, they should press **Save Model**. Make sure participants do not put any text into the "Custom Slot Types" area.

Add an Endpoint

Now comes the time when participants will connect their skill to AWS. Click **Endpoint** on the left side of the screen and Select **AWS Lambda ARN**. One this page **Intent Schema** and **Sample Utterances** should be filled out, then Select **Next**.

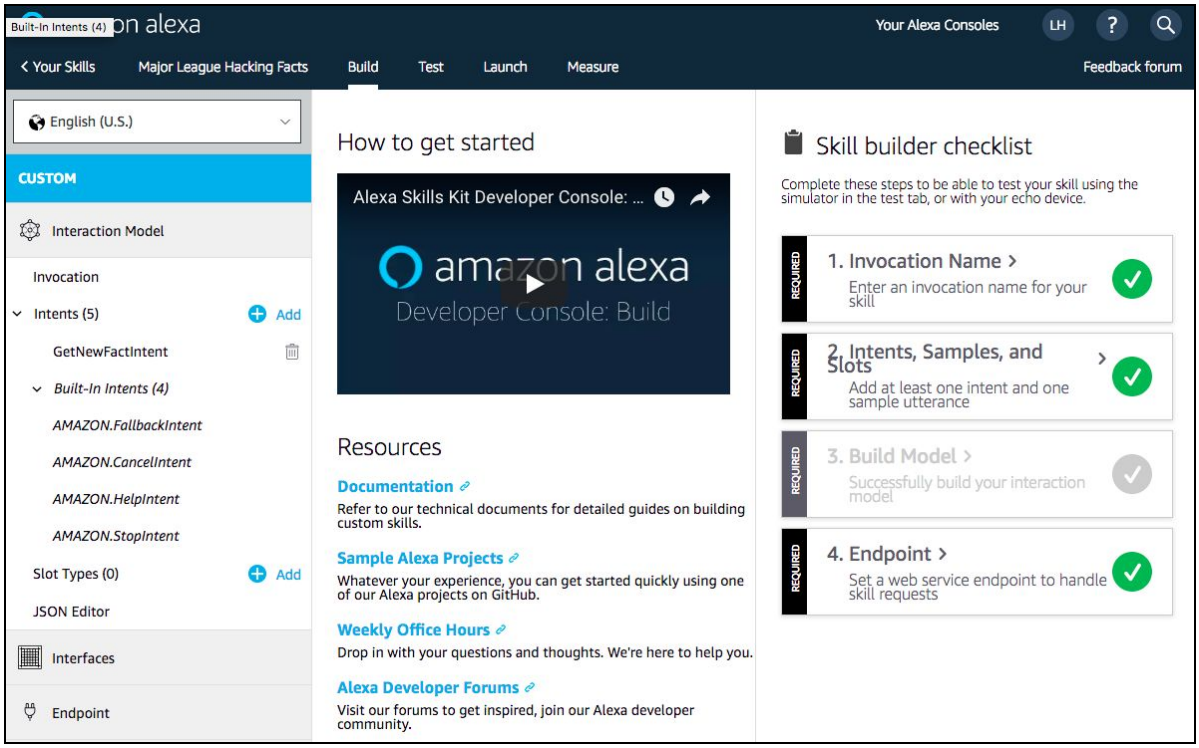


Next, participants should copy their **Skill ID** then return to the Lambda Management Console. In this window, participants should Select **Add Triggers** then **Select Alexa Skills Kit**. Scroll down and paste the **Skill ID** copied from the other page into the **Skill ID** field and press **Add**.



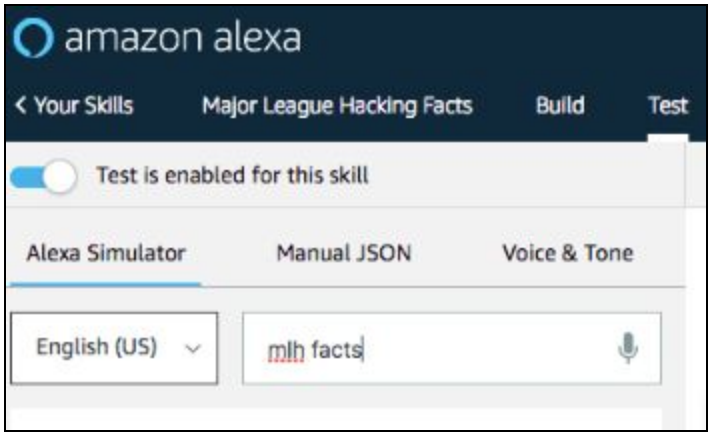
Click **Save** at the Top of the screen, and copy the **ARN code** at the Top Right. Then return to the Alexa Skills Console Tab and paste the **ARN code** into the **Default Region**, then Click **Save Endpoints**. After this, Select **Build**.

This will bring participants to the “Skill Builder Checklist” page. Click **Build Model**. Participants should Select **AWS Lambda ARN** (Amazon Resource Name) as the Endpoint and **North America** as the region (even participants in the EU). A box will pop up that participants should paste their **ARN code** into. “Account linking” should be off and everything else should be unchecked.



After the Build is completed, Click **Test** at the Top of the screen. A new page will open with text that says “Test is Disabled for this skill.” Slide the Slider until it says “Test is enabled.”

When Tests are enabled, participants will be able to type in their Invocation Names.



Test Your Skill

Participants can now test their skills on an Alexa device or head to EchoSim.io to test their skills. If participants want to use an Alexa device, they need to make sure that their Alexa skill was created on the same device that their Alexa is paired with, or else they won't have access to this skill.

If participants' skills are not working, there can be a few reasons.

1. The **ARN code** they copied might be incorrect or incomplete. Double check that they have the right **ARN code**.
2. Invocation names are highly specific. Make sure participants are using the right one.
3. Are participants saying "launch," "start," or "open" before their Invocation Name?
4. Are participants sure that they have no other skills in their accounts with the same invocation name?

Participants need to be very exact when it comes to the phrasing they use with Alexa.

Recap

In this workshop participants have learned how Voice User Interfaces afford new interactions between people and technology. Amazon's Alexa service makes it easier for individuals to create ways that people can interact with their voice by doing much of the heavy lifting of translating a human voice to computer commands.

To make sure participants have understood the theory behind the workshop, they can visit http://mlhlocal.host/quiz for a quiz. To encourage people to make skills for Alexa, Amazon also has promotions, with the ability to win everything from t-shirts to a free Alexa device. Visit http://mlhlocal.host/alexa-promo to get credit for making this skill.

It's far more fun to make a skill that you can share with your friends! To do so, you need to complete the "Publishing Information" and "Privacy and Compliance" sections in the developer portal. Then, participants can publish their skills for the world to see.

Next Steps

When participants have completed the workshop and built their skills, encourage them to think about what they could build in the future. One such possibility is to use Slots to customize their skills further. Slots enable you to pass additional data to your Lambda function, and in turn categorize facts into a handful of categories. You could use Slots to make it so Alexa can give you specifically facts about your favorite movies, or facts about your favorite foods.

Encourage participants to also join MLH's **#amazon-alexa** channel on Slack to give feedback and stay in touch, as well as to check their emails for practice problems to keep learning about VUIs.

End the workshop with an open-ended discussion. Where could voice technology go in the future? What kind of fields can it impact? What new activities can we do with this technology? As participants finish up publishing their skill, use any extra time to connect understanding this one skill to the larger scope of how voice tech can impact the world.