

Workshop

# Hacking with Amazon Alexa

*Welcome to  
MLH Localhost: Hacking with Alexa!*



Wifi Network:  
**[eduroam]**  
**[UB\_Secure]**



Event Hashtag:  
**#MLHLocalhost**

Twitter Handle:  
**@MLHacks**

# Event Sign In Form

<https://tinyurl.com/yd5nqhmt>

1

*Using your Web Browser,  
Open this URL & Fill out the Form:*

**<http://mlhlocal.host/checkin>**

---

2

*Afterwards, Check your Email to Find:*

- Setup Instructions
- An Invite to the MLH Slack
- The Code Samples
- A Workshop FAQ
- These Workshop Slides
- More Learning Resources

# *Welcome! My name is Aniruddha Nandi.*

- 1** I'm will be leading this session to help you learn something new today
- 2** I'm a Junior in CS & Lead of Project AI in the Robotics Club
- 3** Things I love to do include globetrotting, aerial photography and hacking



***Our Mission** is to Empower Hackers.*

**65,000+**  
HACKERS

**12,000+**  
PROJECTS CREATED


**3,000+**  
SCHOOLS

***We hope you learn something awesome today!***  
*Find more resources: <http://mlh.io/>*

# What will you **learn** today?

- 1 Understand Voice User Interfaces & what you can build using them.
- 2 Meet Alexa, an intelligent personal assistant developed by Amazon.
- 3 Create your first voice powered app with Amazon Alexa.

# Table of Contents

- 0. Welcome to MLH Localhost
-  1. Introduction to Alexa & Voice UIs
- 2. Developing for Alexa
- 3. Build Your First Alexa Skill
- 4. Review & Quiz
- 5. Next Steps



# What is Alexa?

Alexa is a Voice User Interface (VUI), that lets you ***speak*** commands, instead of clicking buttons or typing on your keyboard.



Alexa listens to spoken input, uses it to execute tasks or skills in the cloud, and then returns output -- just like a JavaScript function.

# Why do Voice UIs Matter?



Instead of typing, clicking, or tapping - we can physically separate ourselves from our devices and speak commands naturally.

Voice UIs can run code in the cloud and communicate with IoT devices, making them ideal for homes, cars, & more.



# What can you build with Alexa?



*Alexa, ask Lyft for a Lyft Line to work.*

# What can you build with Alexa?

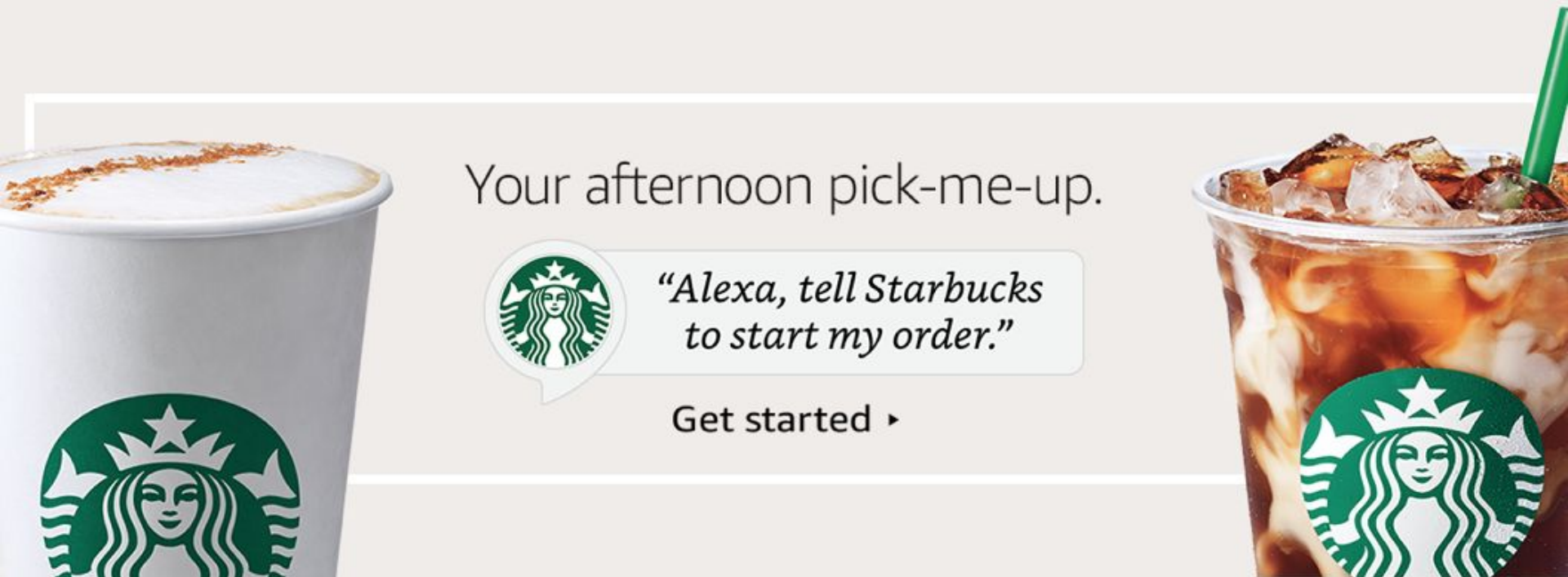


“Alexa, ask Capital One<sup>®</sup>,  
what did I spend?”



*Alexa, ask Capital One, what did I spend?*

# What can you build with Alexa?

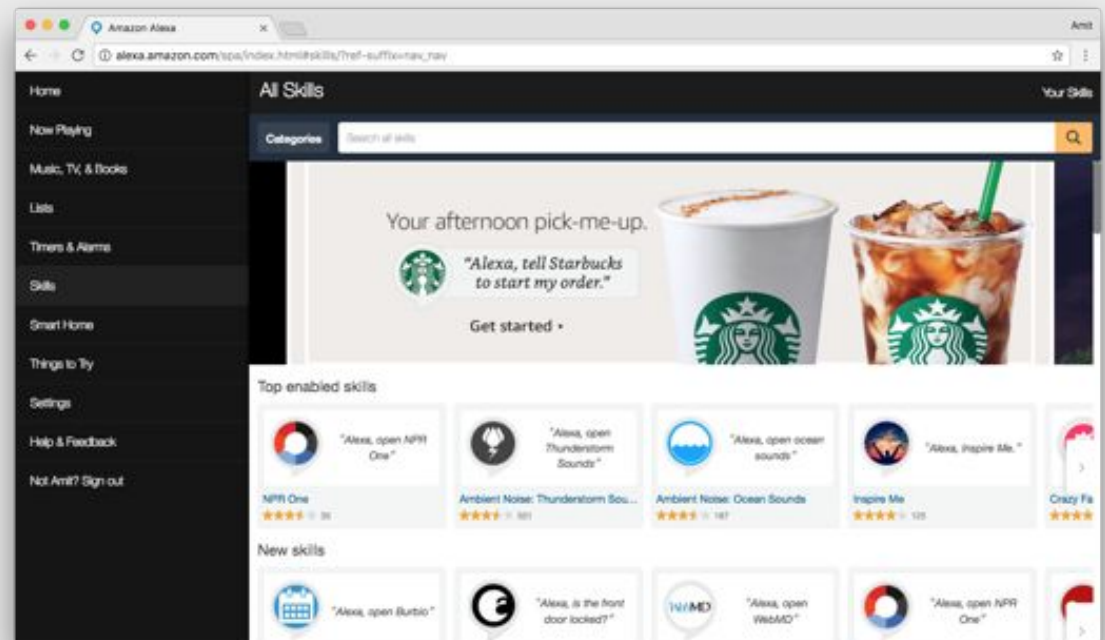


*Alexa, tell Starbucks start my order.*

# Alexa has Skills. Lots of them.

Companion app for device setup, skills, remote control, and more.

[alexa.amazon.com](https://alexa.amazon.com)





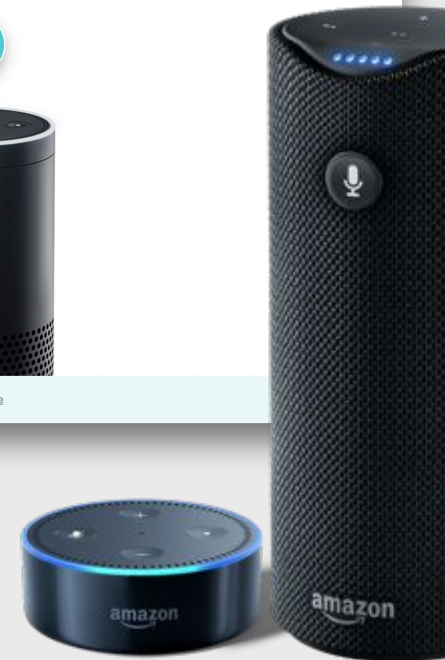
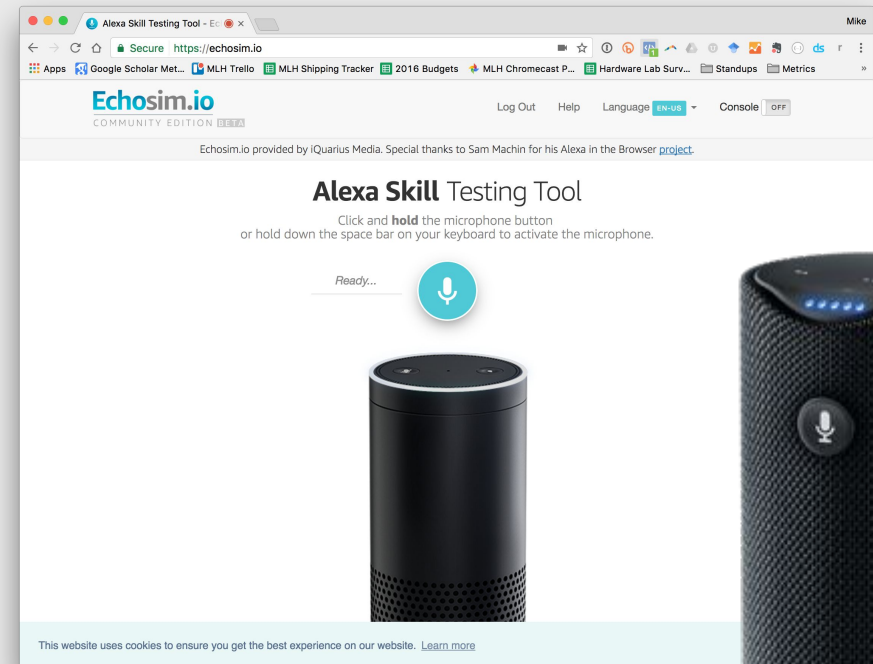
# It's LIVE Demo Time!

Try these commands:


*"Alexa, tell me a joke!"*

*"Alexa, what is  
Major League Hacking?"*

Don't have an Alexa device?  
Head to: [echosim.io](https://echosim.io)



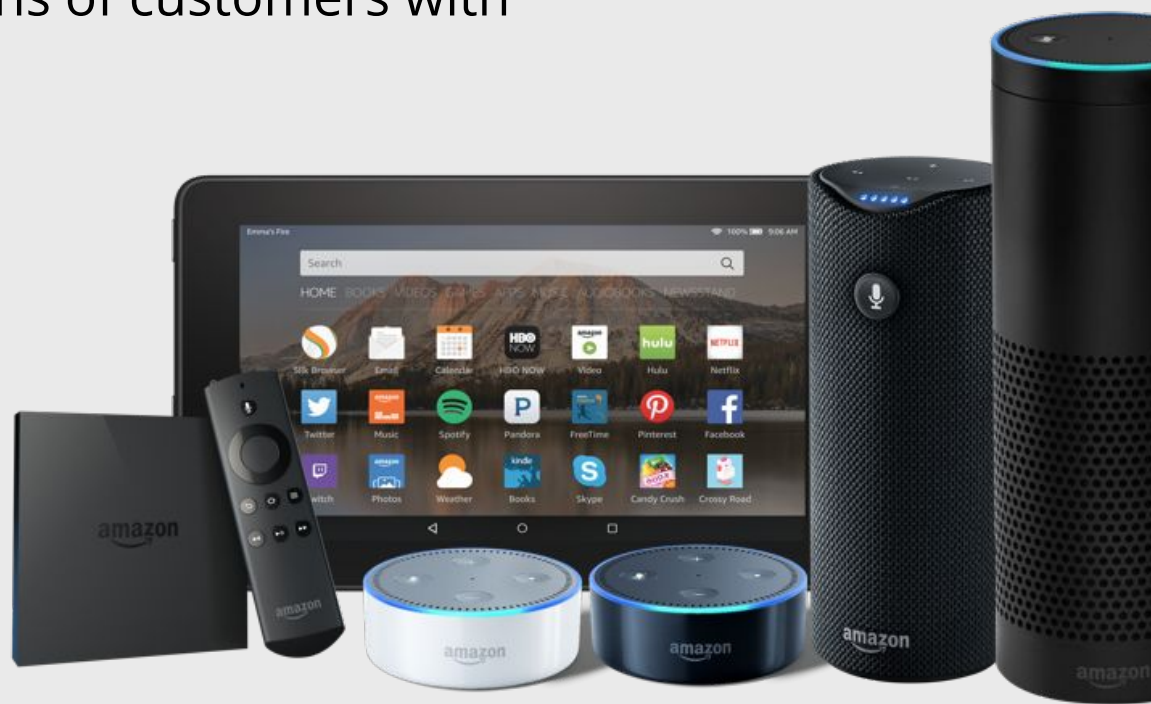
# Table of Contents

- 0. Welcome to MLH Localhost
- 1. Introduction to Alexa & Voice UIs
-  2. Developing for Alexa
- 3. Build Your First Alexa Skill
- 4. Review & Quiz
- 5. Next Steps



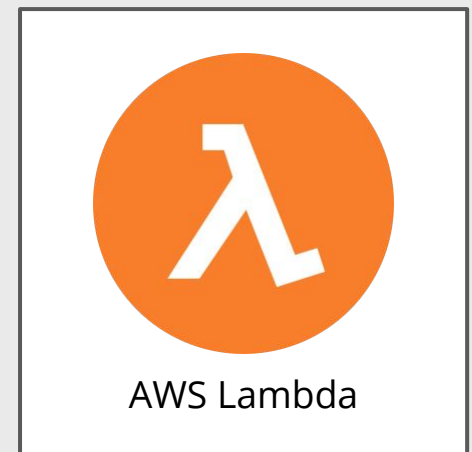
# Why build an Alexa Skill?

1. Alexa, Amazon's voice service, is already integrated into Echo devices and can perform hundreds of skills.
2. Your skill can reach millions of customers with Alexa enabled devices.
3. Building with Alexa is free, easy, & fun!



# Alexa Skills are made of 2 parts:

1. **Front End** - The Alexa Voice UI handles text to speech, converting the audio into something our app can use, etc.
2. **Back End** - The logic code that actually powers our app. Usually this is written on a service called AWS Lambda.



# Speech Recognition is Hard.

You said: *for tē tīmz*. But,  
what did you mean?

1. Forty Times?
2. For Tea Times?
3. For Tee Times?
4. Four Tee Times?



# Parts of Speech Recognition

## Automatic Speech Recognition (ASR)

Enables the recognition and translation of spoken language into text by computers.

1. Traditional phonetics-based using HMM's (Hidden Markov Models)
2. Deep feedforward neural network or RNN (recurrent neural nets) based

# Parts of Speech Recognition

## Natural Language Understanding (NLU)

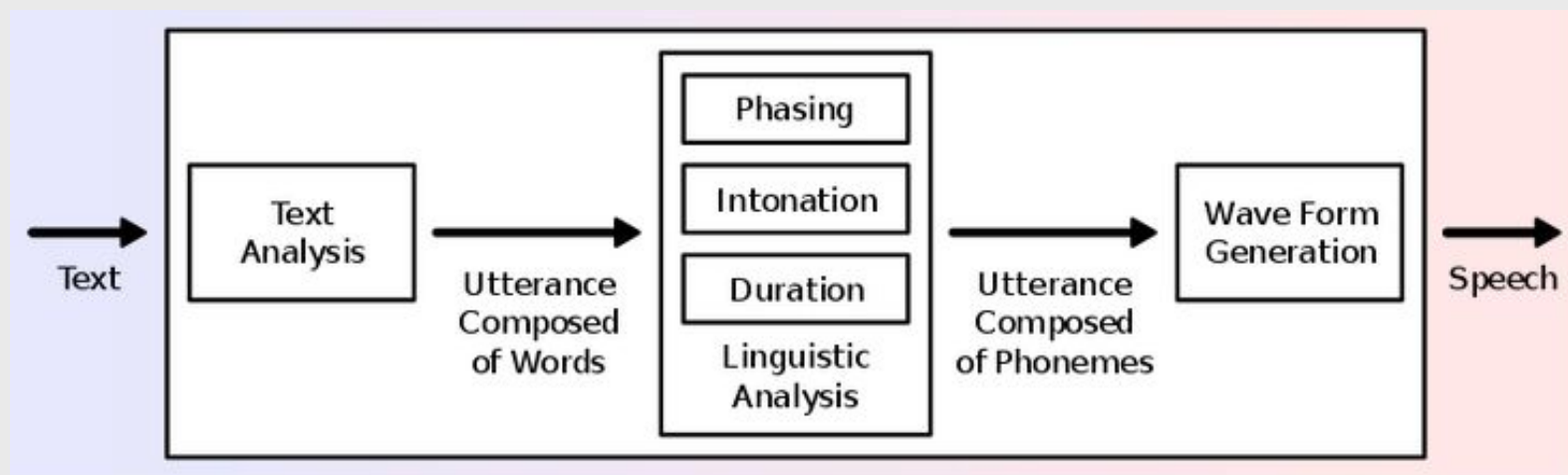
Deals with how to best handle unstructured inputs that are governed by poorly defined, flexible rules and convert them into a structured form that a machine can understand. It is an AI-hard problem.

"I need a flight and hotel in Miami from October 4 to 10"  
need:flight {intent} / need:hotel {intent} / Miami {city} /  
Oct 4 {date} / Oct 10 {date} / sentiment: 0.5723 (neutral)

# Parts of Speech Recognition

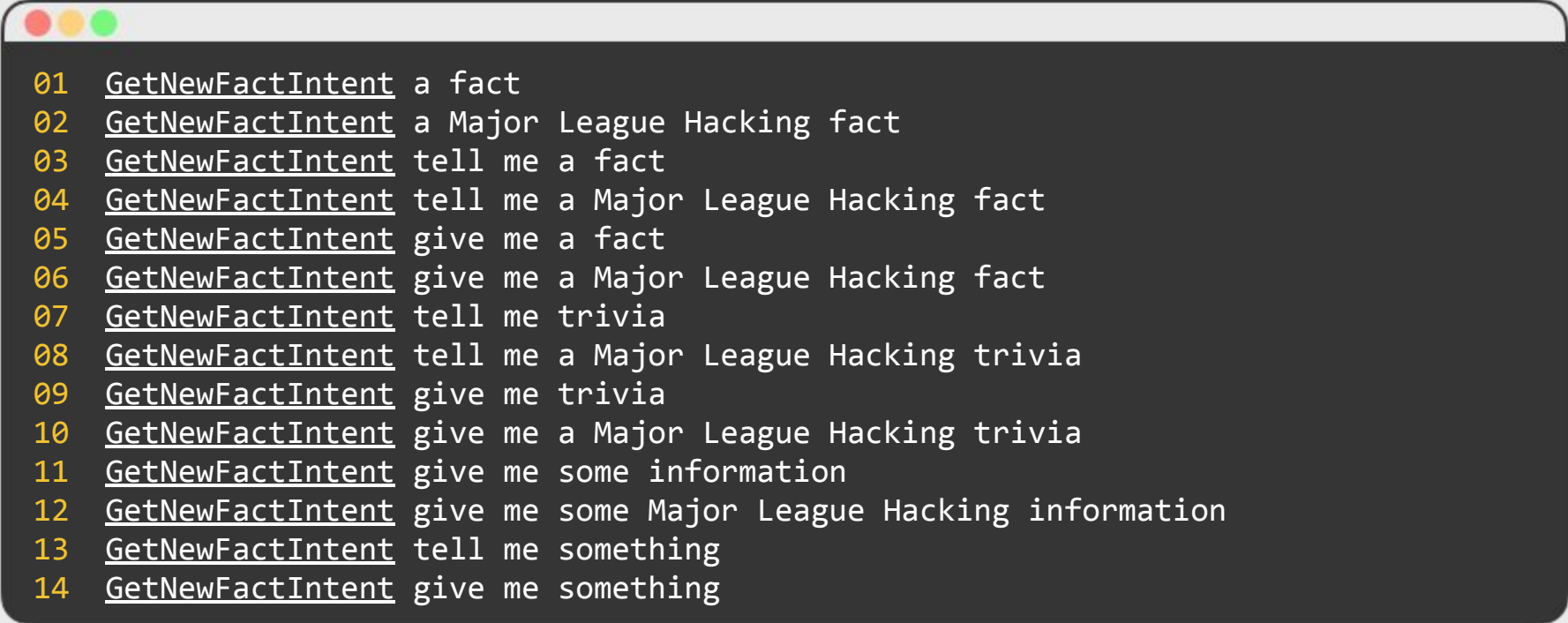
## Text to Speech (TTS)

Converts text back to human-understandable speech



# Alexa uses Sample Utterances for Training.

In order to **map user input** to a behavior, we provide **training data**, for each intent.



```
01 GetNewFactIntent a fact
02 GetNewFactIntent a Major League Hacking fact
03 GetNewFactIntent tell me a fact
04 GetNewFactIntent tell me a Major League Hacking fact
05 GetNewFactIntent give me a fact
06 GetNewFactIntent give me a Major League Hacking fact
07 GetNewFactIntent tell me trivia
08 GetNewFactIntent tell me a Major League Hacking trivia
09 GetNewFactIntent give me trivia
10 GetNewFactIntent give me a Major League Hacking trivia
11 GetNewFactIntent give me some information
12 GetNewFactIntent give me some Major League Hacking information
13 GetNewFactIntent tell me something
14 GetNewFactIntent give me something
```

# Alexa Maps Speech Input to Intents.


Once Alexa figures out what Intent you wanted, you can easily map that back to code.

```
01 {  
02   "intents": [  
03     { "intent": "GetNewFactIntent" },  
04     { "intent": "AMAZON.HelpIntent" },  
05     { "intent": "AMAZON.StopIntent" },  
06     { "intent": "AMAZON.CancelIntent" }  
07   ]  
08 }
```

*"Tell me a Fact about MLH"* → GetNewFactIntent



# Table of Contents

- 0. Welcome to MLH Localhost
- 1. Introduction to Alexa & Voice UIs
- 2. Developing for Alexa
-  3. Build Your First Alexa Skill
- 4. Review & Quiz
- 5. Next Steps

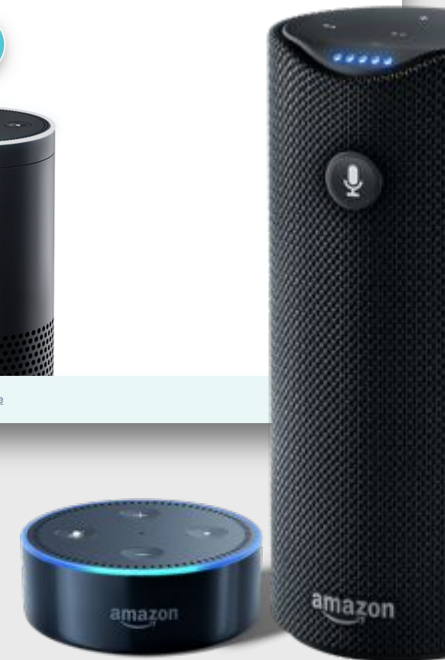
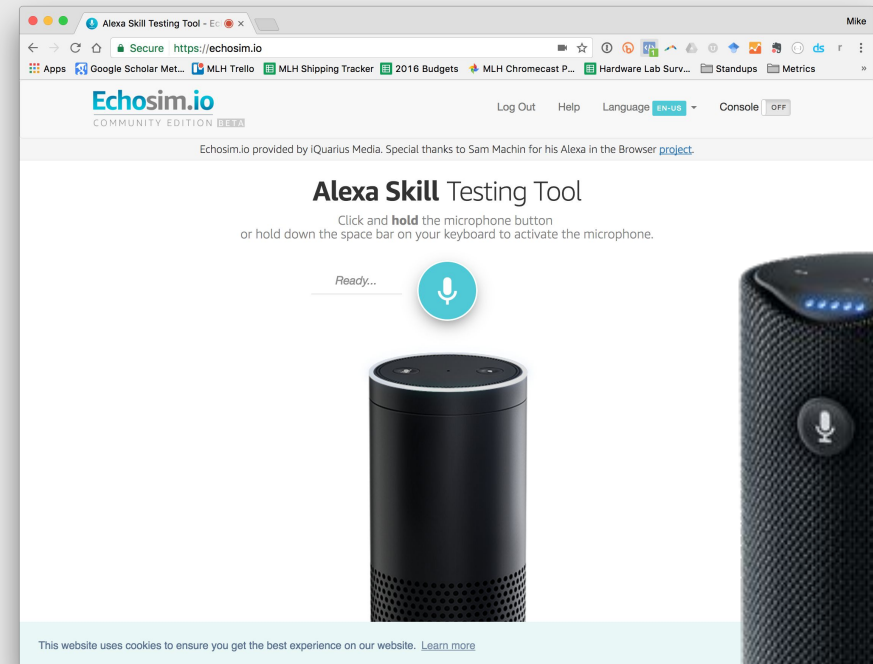
# Demo: About MLH Skill

Ask your Alexa:

*"Alexa, open MLH Facts."*

*"Alexa, ask MLH Facts for a fact."*

Don't have an Alexa device?  
Head to: [echosim.io](https://echosim.io)



# Steps to Build Your Skill:

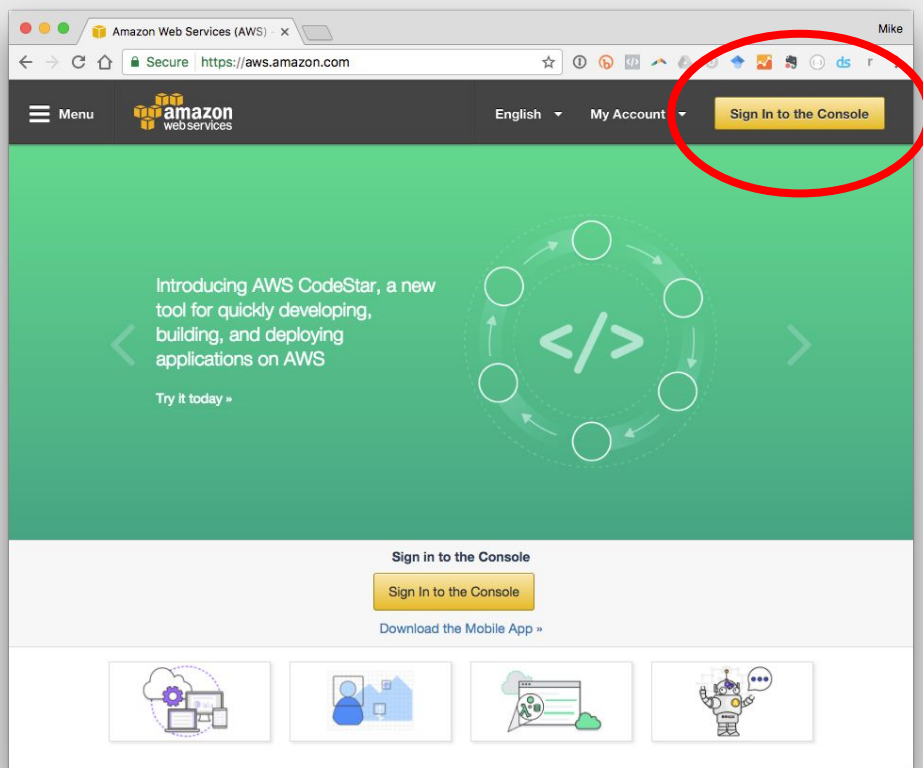
To build your first skill, we'll complete the following steps:

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

# Sign into the AWS Console.

Navigate to:

[mlhlocal.host/alexa-aws](https://mlhlocal.host/alexa-aws)

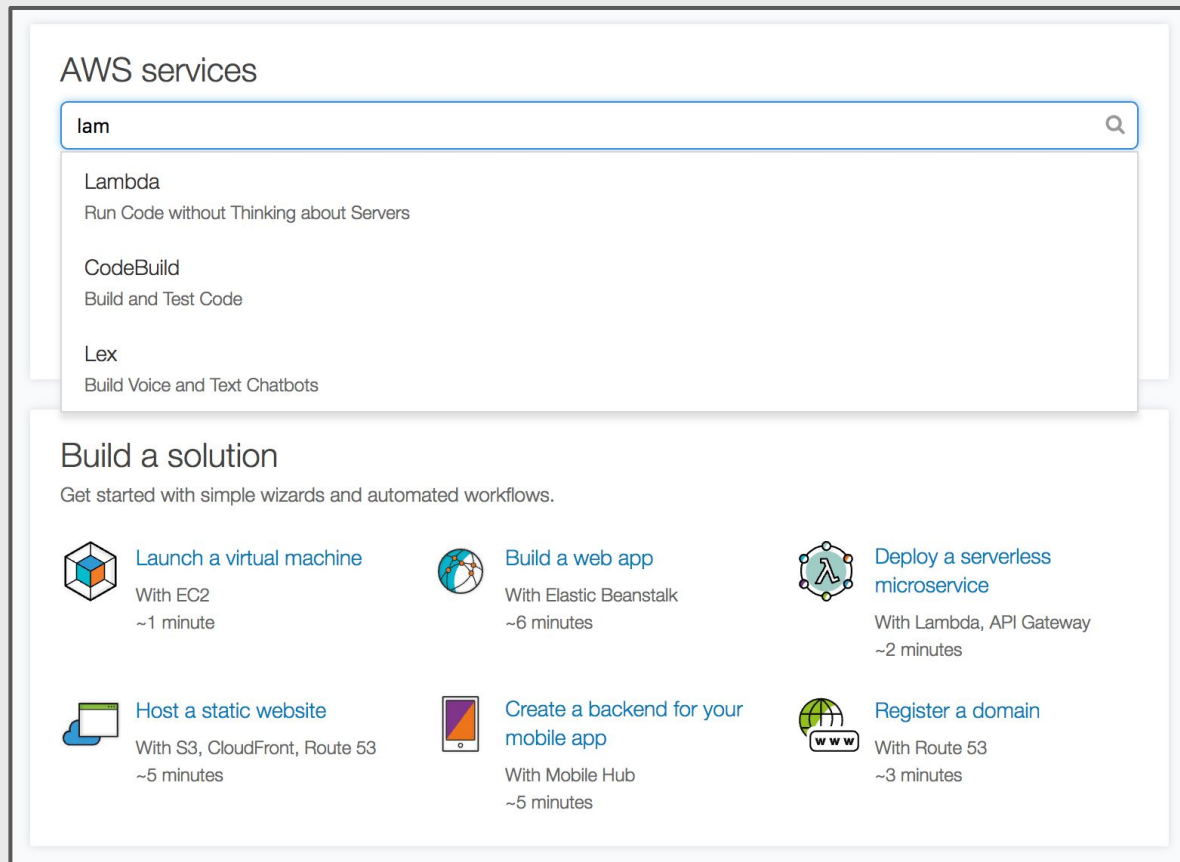


## Instructions

Click on “Sign in to Console” to get started.

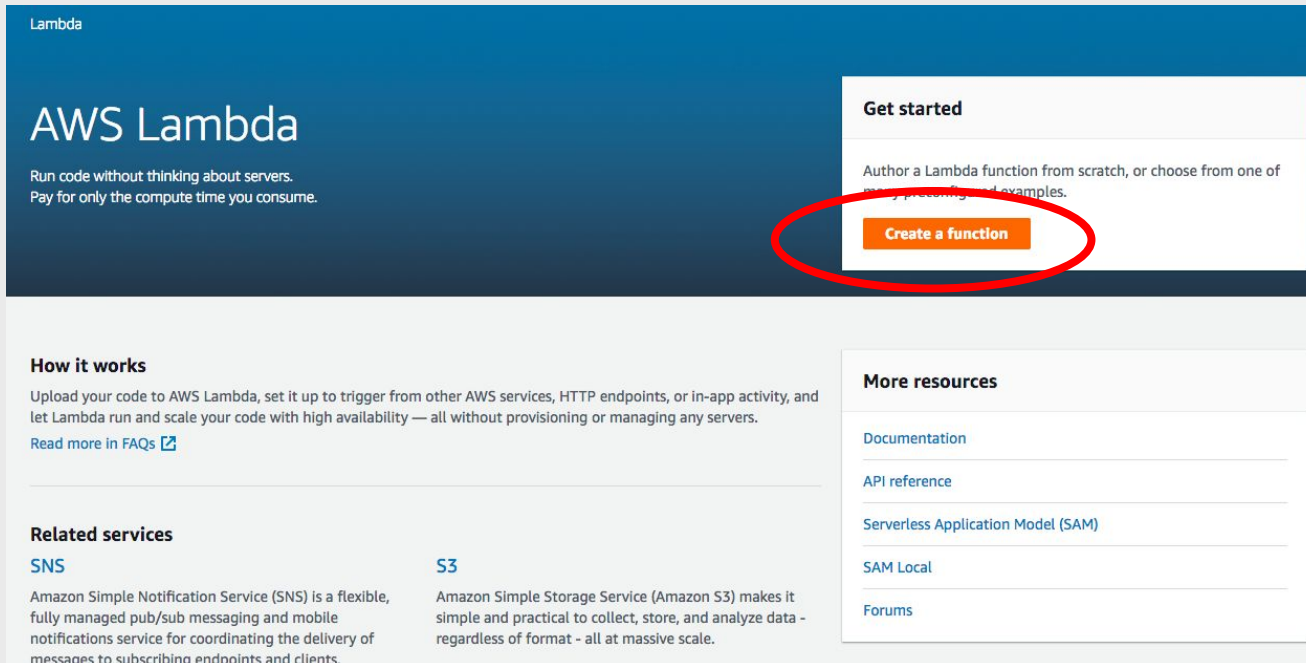
# Navigate to the Lambda Manager.

You can search for Lambda in the search box or find it on the list of available services.



# Create your First Lambda Function.

Each AWS Lambda Function is responsible for one thing (like returning facts about you!).



The screenshot shows the AWS Lambda console's 'Get started' page. The page has a dark blue header with the 'AWS Lambda' logo and a sub-header 'Run code without thinking about servers. Pay for only the compute time you consume.' Below this, there's a 'Get started' section with the text 'Author a Lambda function from scratch, or choose from one of many prebuilt examples.' and a prominent orange 'Create a function' button, which is circled in red. To the left of the button, there's a section titled 'How it works' with a brief description and a link to 'Read more in FAQs'. Below that is a 'Related services' section with links to 'SNS' and 'S3', each followed by a short description. On the right side of the page, there's a 'More resources' section with links to 'Documentation', 'API reference', 'Serverless Application Model (SAM)', 'SAM Local', and 'Forums'.

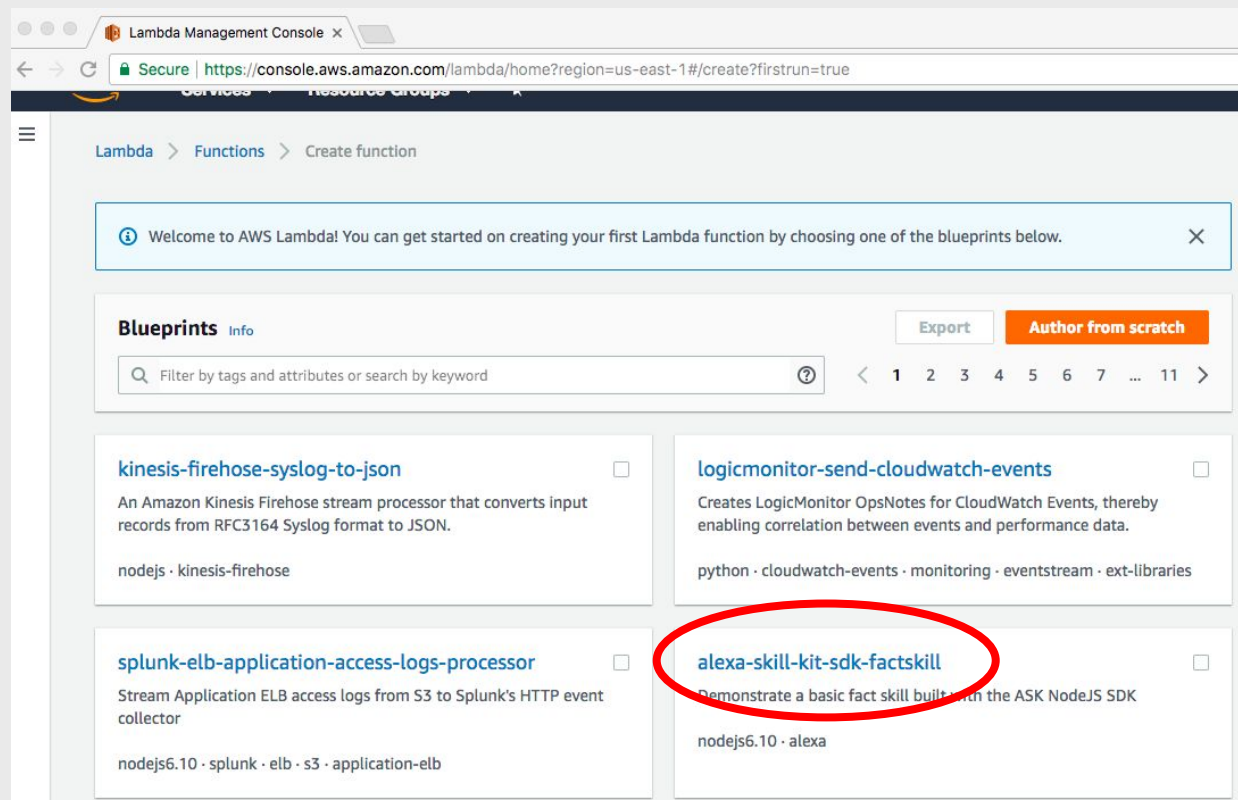
## Instructions

Click “Create a Function” to create your first function!

# Choose a Blueprint.

## Instructions

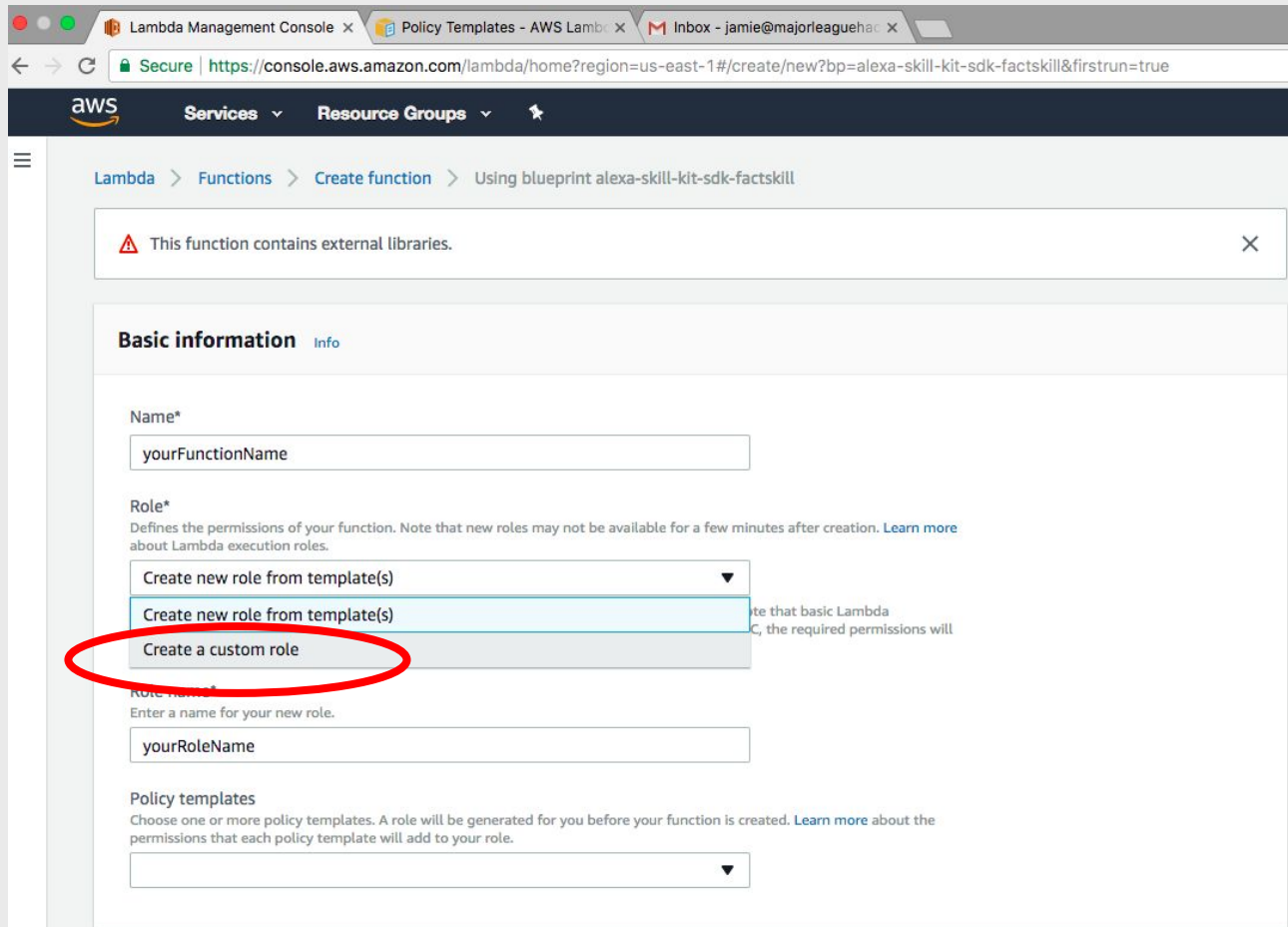
Select “alexa-skill-kit-sdk-factskill” from the options.



# Set Basic Information.

## Instructions

Enter a name, select "Create a custom role," and name your role.



The screenshot shows the AWS Lambda Management Console interface. The browser tabs include 'Lambda Management Console', 'Policy Templates - AWS Lambda', and 'Inbox - jamie@majorleaguehacking.com'. The URL is <https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/new?bp=alexa-skill-kit-sdk-factskill&firstrun=true>. The navigation bar shows 'aws', 'Services', and 'Resource Groups'. The breadcrumb trail is 'Lambda > Functions > Create function > Using blueprint alexa-skill-kit-sdk-factskill'. A warning message states: 'This function contains external libraries.' The 'Basic information' section is active, showing the 'Name\*' field with the value 'yourFunctionName'. The 'Role\*' dropdown menu is open, displaying three options: 'Create new role from template(s)', 'Create new role from template(s)', and 'Create a custom role'. The 'Create a custom role' option is highlighted with a red circle. Below the dropdown, the 'Role name\*' field is empty, with the instruction 'Enter a name for your new role.' and the placeholder text 'yourRoleName'. The 'Policy templates' section is also visible, with the instruction 'Choose one or more policy templates. A role will be generated for you before your function is created. Learn more about the permissions that each policy template will add to your role.'



# Create your IAM Role.

## Instructions

Select "Create a new IAM Role" from dropdown menu.  
Role Name & policy will automatically populate.

**AWS Lambda requires access to your resources**

AWS Lambda uses an IAM role that grants your custom code permissions to access AWS resources it needs.

▼ Hide Details

**Role Summary** ⓘ

<b>Role Description</b>	Lambda execution role permissions
<b>IAM Role</b>	Create a new IAM Role ▼
<b>Role Name</b>	lambda_basic_execution

▶ View Policy Document

Select "Allow" in the lower right corner and you will be returned to your Lambda function.

# Create your IAM Role.

Notice that “lambda\_basic\_execution” is now in the “Existing Role\*” field.

Basic information

Info

Name\*

yourFunctionName

Role\*

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Existing role\*

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

lambda\_basic\_execution

# Step 1: Download the Sample Code

To get the sample code, head to this URL:

[mlhlocal.host/alexa-code](http://mlhlocal.host/alexa-code)

---

## Step 2: Open Index.js

Unzip the directory and open [src/index.js](#) in your favorite code editor.



# Customize your Facts

## Instructions

Find the **FACTS** array inside `index.js`.

Replace the facts about MLH with facts of your choice!

```
12 // TODO: replace with facts about yourself
13 const FACTS = [
14   "Major League Hacking is commonly called MLH.",
15   "Major League Hacking's mission is to empower hackers.",
16   "Over 65,000 student hackers participated in Major League Hacking...",
17   "Major League Hacking was founded in 2013 by Swift and Jon.",
18   "Hackers created over 12,000 projects at MLH hackathons in 2016.",
19   "Over 200 schools around the world hosted MLH hackathons in 2016.",
20   "Major League Hacking is headquartered in New York City."
21 ];
```

# Code Review: The Handlers Object

The `handlers` object tells Alexa how to handle various actions. `GetFact` is the main logic of our application.

```
23 var handlers = {
24   'LaunchRequest': function() { this.emit('GetFact'); },
25   'GetNewFactIntent': function() { this.emit('GetFact'); },
26   'GetFact': function() {
27     // Randomly select a fact from the array
28     const factIndex = Math.floor(Math.random() * FACTS.length);
29     const randomFact = FACTS[factIndex];
30
31     // Create speech output
32     const speechOutput = "Here's your fact: " + randomFact;
33     this.emit(':tellWithCard', speechOutput, "MLH Facts", randomFact);
34   }
35 };
```

# Code Review: The Handler Function

The `handler` function tells Alexa how to route voice commands by passing a copy of the `handlers` object.

```
37 // AWS Lambda calls this function every time Alexa uses our skill.
38 exports.handler = function(event, context, callback) {
39   // Include the AWS Alexa Library.
40   const Alexa = require("alexajs");
41
42   // Create an instance of the Alexa library & pass it the requested command.
43   var alexa = Alexa.handler(event, context);
44
45   // Give our Alexa instance handling instructions & execute the request.
46   alexa.registerHandlers(handlers);
47   alexa.execute();
48 };
```

# Create your Function.

## Instructions

Click “Create Function” at the bottom of the page

### Lambda function code

Code is pre-configured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

#### Runtime

Node.js 6.10

```
2  /* eslint quote-props: ["error", "consistent"] */
3  /**
4   * This sample demonstrates a simple skill built with the Amazon Alexa Skills
5   * nodejs skill development kit.
6   * This sample supports multiple languages. (en-US, en-GB, de-DE).
7   * The Intent Schema, Custom Slots and Sample Utterances for this skill, as well
8   * as testing instructions are located at https://github.com/alexa/skill-sample-nodejs-fact
9   */
10
11  'use strict';
12
13  const Alexa = require('alexa-sdk');
14
15  const APP_ID = undefined; // TODO replace with your app ID (OPTIONAL).
16
17  const languageStrings = {
18    'en': {
19      translation: {
20        FACTS: [
21          'A year on Mercury is just 88 days long.',
22          'Despite being farther from the Sun, Venus experiences higher temperatures than Mercury.',
23          'Venus rotates anti-clockwise, possibly because of a collision in the past with an asteroid.',
24          'On Mars, the Sun appears about half the size as it does on Earth.',
25          'Earth is the only planet not named after a god.',
26          'Jupiter has the shortest day of all the planets.',
27          'The Milky Way galaxy will collide with the Andromeda Galaxy in about 5 billion years.'
```

\* These fields are required.

Cancel

Previous

Create function

# Copy & Paste your Code into the Lambda Editor

## Instructions

Copy the entire contents of index.js and paste it over the code in the inline editor on the AWS Console.

**Configuration** Triggers Monitoring

▼ **Function code**

⚠ This function contains external libraries. Uploading a new file will override these libraries. ✕

Code entry type: Edit code inline ▼ Runtime: Node.js 6.10 ▼ Handler: index.handler [Info](#)

```

index.js
1  /* eslint-disable func-names */
2  /* eslint quote-props: ["error", "consistent"]*/
3  /**
4   * This sample demonstrates a simple skill built with the Amazon Alexa Skills
5   * nodejs skill development kit.
6   * This sample supports multiple languages. (en-US, en-GB, de-DE).
7   * The Intent Schema, Custom Slots and Sample Utterances for this skill, as well
8   * as testing instructions are located at https://github.com/alexa/skill-sample-nodejs-fact
9   */
10
11 'use strict';
12
13 const Alexa = require('alexa-sdk');
14
15 ⚠ const APP_ID = undefined; // TODO replace with your app ID (OPTIONAL).
16
17 const languageStrings = {
18   'en': {
19     translation: {

```



# Save your Code.

## Instructions

Click the white "Save" button (not the orange one).

Lambda > Functions > yourFunctionName ARN - arn:aws:lambda:us-east-1:448703130969:function:yourFunctionName

yourFunctionName Qualifiers ▼ Actions ▼ Save Select a test event.. ▼ Save and test

✔ Congratulations! Your Lambda function "yourFunctionName" has been successfully created. You can now change its code and configuration. Click on the "Test" button to input a test event when you are ready to test your function. ✕

**Configuration** Triggers Monitoring

▼ **Function code**

⚠ This function contains external libraries. Uploading a new file will override these libraries. ✕

Code entry type Runtime Handler [Info](#)

Edit code inline ▼ Node.js 6.10 ▼ index.handler

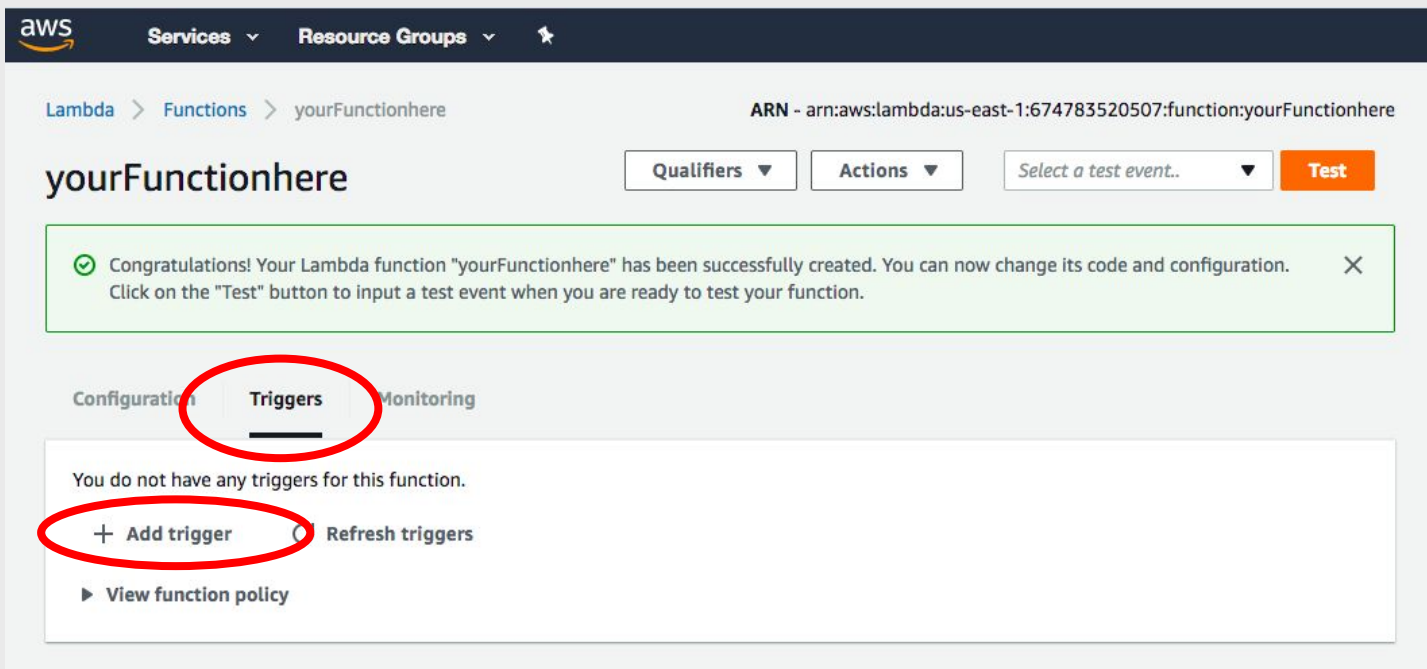
# Configure your Trigger.

We need to setup a trigger to call our new Lambda Function (*in our case Alexa Skills Kit*).

## Instructions

Click "Triggers"

Click "+ Add Trigger"

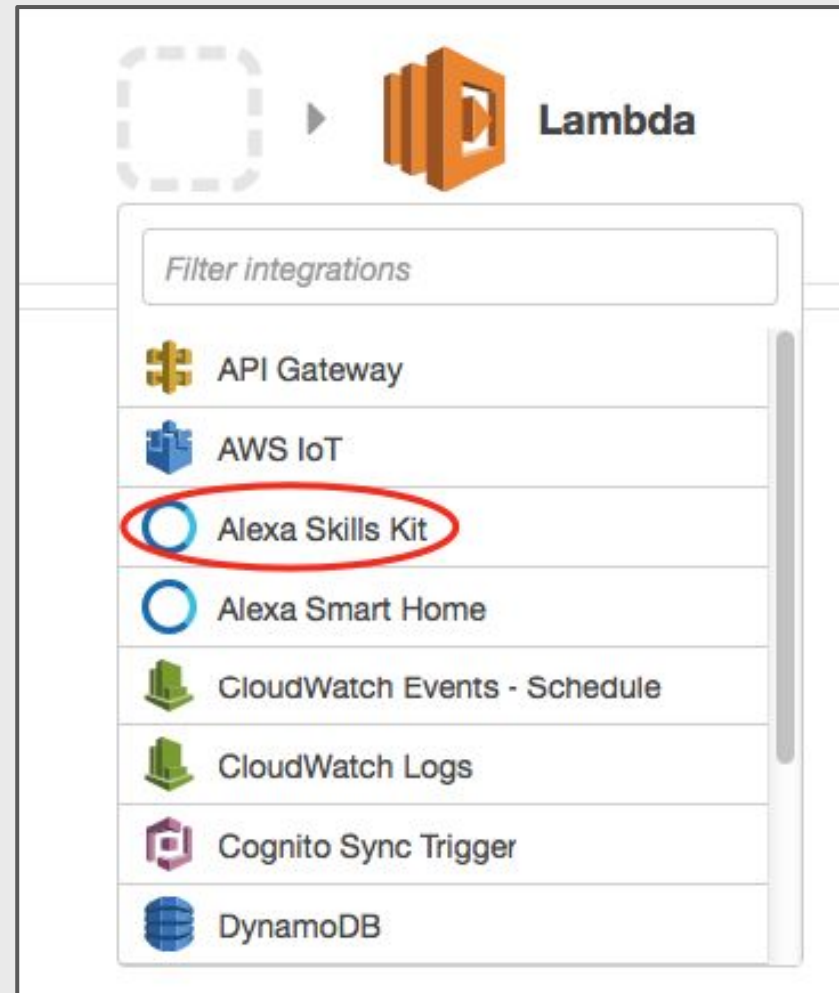


The screenshot shows the AWS Lambda console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a star icon. Below this, the breadcrumb trail reads 'Lambda > Functions > yourFunctionhere'. The function's ARN is displayed as 'arn:aws:lambda:us-east-1:674783520507:function:yourFunctionhere'. The main heading is 'yourFunctionhere', followed by buttons for 'Qualifiers', 'Actions', a dropdown for 'Select a test event..', and a 'Test' button. A green notification box contains a success message: 'Congratulations! Your Lambda function "yourFunctionhere" has been successfully created. You can now change its code and configuration. Click on the "Test" button to input a test event when you are ready to test your function.' Below the notification, there are three tabs: 'Configuration', 'Triggers' (which is selected and circled in red), and 'Monitoring'. Under the 'Triggers' tab, the text says 'You do not have any triggers for this function.' Below this text, there are two buttons: '+ Add trigger' (circled in red) and 'Refresh triggers'. At the bottom, there is a link to 'View function policy'.

# Configure your Trigger.

## Instructions

Click on the gray dash-lined box  
Select "Alexa Skills Kit" from the  
dropdown menu



# Configure your Trigger.


## Instructions


Click "Submit."


Add trigger

Configure your Lambda function **yourFunctionhere** to respond to events from the selected trigger. Click on the box below to select your trigger type.

Alexa Skills Kit







Lambda

Lambda will add the necessary permissions for Amazon Alexa to invoke your Lambda function from this trigger.  
[Learn more](#) about the Lambda permissions model.

Cancel

Submit

# Test your Trigger.

## Instructions

Click "Save and Test."

yourFunctionName

Qualifiers ▾

Actions ▾

Save

Select a test event.. ▾

Save and test

✔

Congratulations! Your Lambda function "yourFunctionName" has been successfully created. You can now change its code and configuration. Click on the "Test" button to input a test event when you are ready to test your function.


Successfully added the trigger to function yourFunctionName. The function is now receiving events from the trigger. To configure the Alexa service to work with your Lambda function, go to the [Alexa Developer](#) portal.

✕

Configuration

Triggers

Monitoring



Alexa Skills Kit

alex-appkit.amazon.com

ⓘ

To configure your Alexa skill, go to the [Alexa Developer Portal](#).

+

Add trigger

↻

Refresh triggers

▶

View function policy

Delete

# Test your Trigger.

## Instructions

# Click "Hello World."

Scroll down and select "Alexa Intent - GetNewFact."

## Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event

☐ Edit saved test events

Event template

Hello World ▼

🔍 |

CloudFront Response Generation

Kinesis Firehose Apache Log

S3 Delete

Alexa

Alexa Smart Home - Discovery

Alexa Intent - GetNewFact

Alexa Start Session

Alexa Intent - Recipe

Alexa Smart Home - Turn Off

Alexa Intent - Answer

# Test your Trigger.

We need to setup a trigger to call our new Lambda

## Instructions

Name your test event.

Click "Create."

**Configure test event**

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event  
☐ Edit saved test events

Event template  
Alexa Intent - GetNewFact

Event name  
GetNewMLHFact

```
1 {
2   "session": {
3     "new": false,
4     "sessionId": "amzn1.echo-api.session.[unique-value-here]",
5     "attributes": {},
6     "user": {
7       "userId": "amzn1.ask.account.[unique-value-here]"
8     },
9     "application": {
10      "applicationId": "amzn1.ask.skill.[unique-value-here]"
11    }
12  },
13  "version": "1.0",
14  "request": {
15    "locale": "en-US",
16    "timestamp": "2016-10-27T21:06:28Z",
17    "type": "IntentRequest",
18    "requestId": "amzn1.echo-api.request.[unique-value-here]",
19    "intent": {
20      "slots": {},
21      "name": "GetNewFactIntent"
22    }
23  },
24  "context": {
25    "AudioPlayer": {
26      "playerActivity": "IDLE"
27    }
28  }
29 }
```

Cancel Create

# Test your Trigger.

## Instructions

Click "Test."

Lambda > Functions > yourFunctionHere

ARN - arn:aws:lambda:us-east-1:674783520507:function:yourFunctionHere

yourFunctionHere

Qualifiers ▼

Actions ▼

GetNewMLHFact ▼

Test

✓ Execution result: succeeded (logs)

► Details



# Test your AWS Lambda Function

If it's working, you can click "Details" to see output similar to the following:

yourFunctionName

Qualifiers ▼

Actions ▼

GetNewMLHFact ▼

Test

✓ Execution result: succeeded ([logs](#))

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "version": "1.0",
  "response": {
    "shouldEndSession": true,
    "outputSpeech": {
      "type": "SSML",
      "ssml": "<say> Here's your fact: Major League Hacking is headquartered in New York City. </say>"
    },
    "card": {
      "type": "Simple",
      "title": "Major League Hacking (MLH) Facts"
    }
  }
}
```

## Summary

Code SHA-256  
pQzEC+puZNYW0djE  
m2I1X3Manu17/WPn7

## Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

# Note Down your Function's ARN.

You'll need it in a few steps, so either keep this tab open or save it in a file somewhere now.

Lambda > Functions > yourFunctionHere

yourFunctionHere

ARN - arn:aws:lambda:us-east-1:674783520507:function:yourFunctionHere

Qualifiers ▼

Actions ▼

GetNewMLHFact ▼

Test

✓ Execution result: succeeded ([logs](#))

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

# Sign into the Amazon Developer Portal.

Head over to:  
[mlhlocal.host/alexa-portal](https://mlhlocal.host/alexa-portal)

The screenshot shows the Amazon Developer Portal for Alexa. The browser address bar displays <https://developer.amazon.com/alexa>. The navigation bar includes links for Apps & Games, Alexa, AWS, Services & APIs, Devices, Blog, Docs, and Support. The main header features the text "Build Engaging Alexa Skills and Gain Deeper Insights" and a "Learn more" link. Below this, three key services are highlighted: Alexa Skills Kit, Alexa Voice Service, and Alexa Fund. The "Learn to build with Alexa" section contains four cards: AWS Promotional Credits for Alexa, AVS for the UK and Germany Now Available, Alexa SDK for Node.js, and Flash Briefing Skill API. The "Get started" section includes links to create an Alexa skill, learn how to build a first skill, and integrate Alexa into a device. A "Subscribe to Alexa Announcements" form is also present.

**Build Engaging Alexa Skills and Gain Deeper Insights**  
Use the new Device Address API and a new metrics dashboard to build more engaging skills.  
[Learn more >](#)

**Alexa Skills Kit**  
A free SDK that lets you easily build voice experiences

**Alexa Voice Service**  
Bring voice capabilities to your connected device

**Alexa Fund**  
\$100 million in investment to fuel voice technology innovation

**Learn to build with Alexa**

- AWS Promotional Credits for Alexa**  
Amazon makes it free for developers to build and host most Alexa skills
- AVS for the UK and Germany Now Available**  
Language and region-specific voice services to expand your audience.
- Alexa SDK for Node.js**  
Build skills faster and with less complexity—get started on Github.
- Flash Briefing Skill API**  
Learn more about the Flash Briefing Skill API and how to build a Flash Briefing skill.

**Get started**

- ▶ [Create an Alexa skill now](#)
- ▶ [Learn how to build your first skill](#)
- ▶ [Integrate Alexa into your device](#)

**Subscribe to Alexa Announcements**


Email

[Subscribe](#)

# Create your account.

## Instructions

Fill in this form.



### Create account

**Your name**

**Email**

**Password**

**Re-enter password**

Create your Amazon Developer account

By creating an account, you agree to Amazon's [Conditions of Use](#) and [Privacy Notice](#).

Already have an account? [Sign in](#) »

# Create your Developer Profile.

## Instructions

Fill out the registration form & agree to the terms of use.

Registration

1. Profile Information

2. App Distribution Agreement

3. Payments

\* indicates a required field.

Country/Region \*

United States

First name \*

Local

Last name \*

Host

Email address \*

localhost@mlh.io

Phone number \*

212-555-1212

e.g. 212-555-1212, +44 0161 715 3369

Fax number

Developer name or company name \*

MLH Localhost

Displayed on your apps at Amazon.com

Developer description

Maximum characters: 4000. Remaining: 3611

Major League Hacking (MLH) is the official student hackathon league. Each year, we power over 200 weekend-long Invention competitions that inspire innovation, cultivate communities, and teach compute science skills to more than 65,000 students around the world. MLH is an engaged and passionate maker community, consisting of the next generation of technology leaders and entrepreneurs.

Address 1 \*

149 East 23rd St

Address 2

#438

City \*

New York

State \*

New York

Zip code/Postal code \*

10159

Customer support email address

Customer support phone

Customer support website

Cancel

Save and Continue

# Create your Developer Profile.

## Instructions

Fill out the registration form & agree to the terms of use.

### Registration



1. Profile Information



2. App Distribution Agreement

3. Payments

\* indicates a required field.

Do you plan to monetize your digital content, such as charging for apps or games or selling in-app items or in-game items, or by receiving cash rewards for your skills? \*

☒ No  
☐ Yes

Do you plan to monetize apps by displaying ads from the Amazon Mobile Ad Network or Mobile Associates? \*

☒ No  
☐ Yes

**Note:** You may still monetize later if you select "No" by entering payment and tax information from the Settings menu.

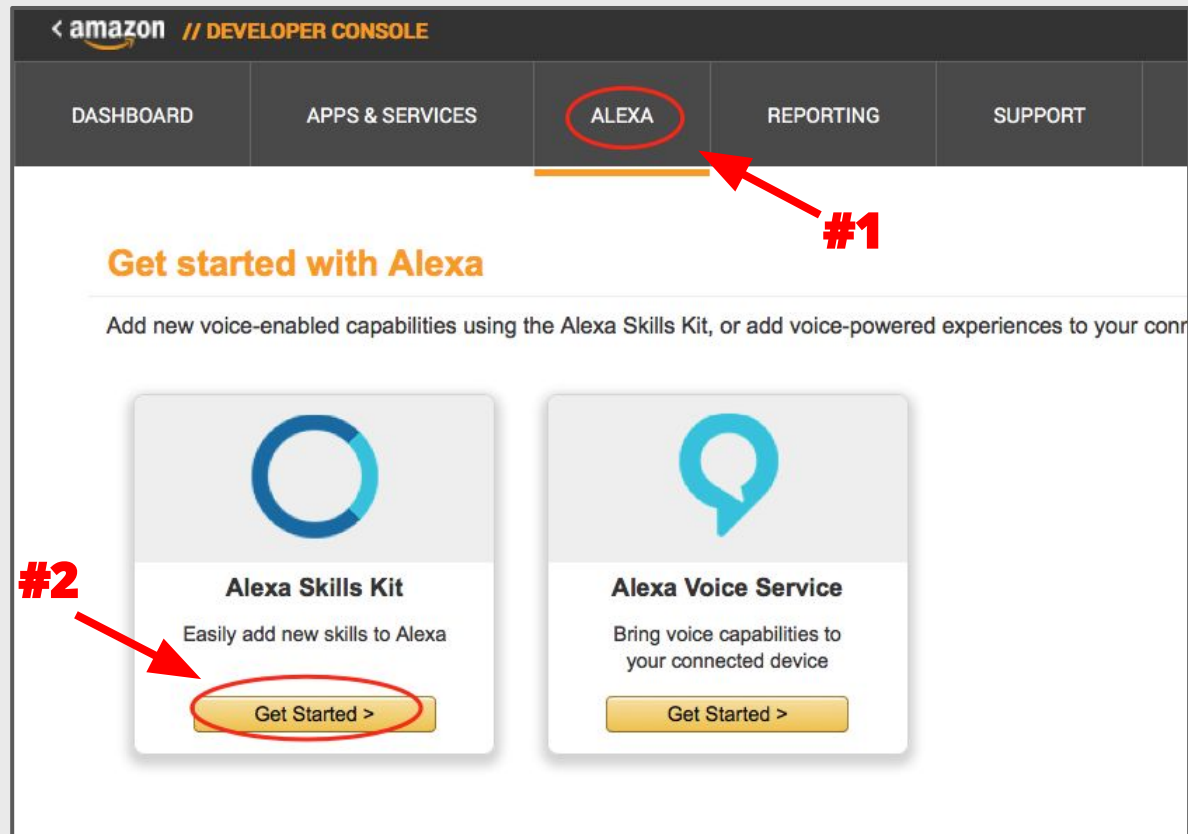
Cancel

Save and Continue

# Create a New Alexa Skill.

## Instructions

Navigate to the Alexa tab.



# Create a New Alexa Skill

## Instructions

Click on "Getting Started" under "Alexa Skills Kit."  
Click "Add a New Skill" in the top right.

### 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.

### Good news! Developers can earn money for the most engaging skills

We're rewarding developers who design Alexa skills that customers love most! Developers can earn money each month for eligible skills that have the highest customer engagement in eligible skill categories. What's your next big idea? [Learn more](#).

Name	Language	Type	Modified	Status	Actions
------	----------	------	----------	--------	---------



# What is the Invocation Name?

This is a **1-3 word phrase** that users will say out loud to launch your skill.

*Alexa, open MLH facts.*



Wake Word



Launch



Skill Invocation  
Name

# Name your Skill

## Instructions

1. Select “*Custom Interaction Model*” for Skill Type
2. Enter a Name for your Skill.
3. Enter an Invocation Name for your Skill.
4. Select the "No" option under Audio Player and click “Next”.

<b>Skill Type</b> Define a custom interaction model or use one of the predefined skill APIs. <a href="#">Learn more</a>	<input checked="" type="radio"/> Custom Interaction Model <input type="radio"/> Smart Home Skill API <input type="radio"/> Flash Briefing Skill API
<b>Language</b> Language of your skill	English (U.S.)
<b>Name</b> Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.	Major League Hacking (MLH) Facts
<b>Invocation Name</b> The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...". <a href="#">Invocation Name Guidelines</a>	MLH Facts

# Grab the Intent Schema JSON

There's a file called `SpeechAssets/IntentSchema.json` inside the code you downloaded earlier with the following JSON:

```
01 {  
02   "intents": [  
03     { "intent": "GetNewFactIntent" },  
04     { "intent": "AMAZON.HelpIntent" },  
05     { "intent": "AMAZON.StopIntent" },  
06     { "intent": "AMAZON.CancelIntent" }  
07   ]  
08 }
```

This code tells Alexa which code to trigger in our Lambda Function and some basic defaults like “STOP” and “HELP”.

# Paste the JSON into the Intent Schema Box

## Instructions

Copy the code you found in `SpeechAssets/IntentSchema.json` into the box on the Alexa Developer dashboard.

### Intent Schema

The schema of user intents in JSON format. For more information, see [Intent Schema](#).

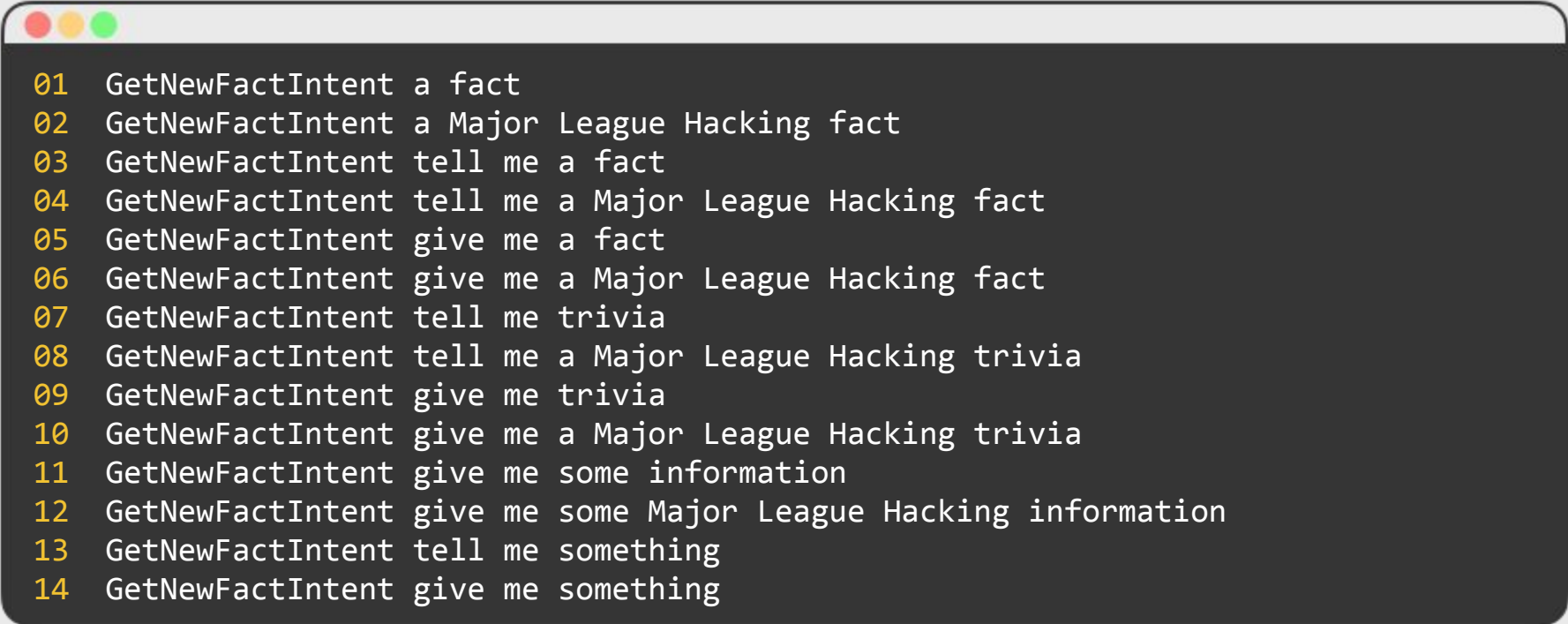
Also see [built-in slots](#) and [built-in intents](#).

```
1 {  
2   "intents": [  
3     { "intent": "GetNewFactIntent" },  
4     { "intent": "AMAZON.HelpIntent" },  
5     { "intent": "AMAZON.StopIntent" },  
6     { "intent": "AMAZON.CancelIntent" }  
7   ]  
8 }
```

# Customize your Utterances

## Instructions

Open up [SpeechAssets/SampleUtterances.txt](#) and replace “Major League Hacking” with your name:



```
01  GetNewFactIntent a fact
02  GetNewFactIntent a Major League Hacking fact
03  GetNewFactIntent tell me a fact
04  GetNewFactIntent tell me a Major League Hacking fact
05  GetNewFactIntent give me a fact
06  GetNewFactIntent give me a Major League Hacking fact
07  GetNewFactIntent tell me trivia
08  GetNewFactIntent tell me a Major League Hacking trivia
09  GetNewFactIntent give me trivia
10  GetNewFactIntent give me a Major League Hacking trivia
11  GetNewFactIntent give me some information
12  GetNewFactIntent give me some Major League Hacking information
13  GetNewFactIntent tell me something
14  GetNewFactIntent give me something
```

# Paste your Custom Utterances into the Sample Utterances Box

## Instructions

Copy your customized `SpeechAssets/SampleUtterances.txt` into the box on the Alexa Developer dashboard.

### Sample Utterances

These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. [Learn more](#)

Up to 3 of these will be used as Example Phrases, which are hints to users.

```

1 GetNewFactIntent a fact
2 GetNewFactIntent a Major League Hacking fact
3 GetNewFactIntent tell me a fact
4 GetNewFactIntent tell me a Major League Hacking fact
5 GetNewFactIntent give me a fact
6 GetNewFactIntent give me a Major League Hacking fact
7 GetNewFactIntent tell me trivia
8 GetNewFactIntent tell me a Major League Hacking trivia
9 GetNewFactIntent give me trivia
10 GetNewFactIntent give me a Major League Hacking trivia
11 GetNewFactIntent give me some information
  
```

# Save your Interaction Model and Continue.

## Instructions

Once you've defined your intent schema & sample utterances, hit "next" in the bottom corner to move on.

< Back to the list of skills [Getting started](#)

**My Fact Skill**  
DEVELOPMENT  
3/24/16 \*Fields required for certification

**Skill Information** ✓  
**Interaction Model** ✓  
**Configuration** ✓  
**Test** ✓  
**Publishing Information** ✓  
**Privacy & Compliance** ✓

**Intent Schema\***  
The schema of user intents in JSON format. For more information, see [Intent Schema](#).  
Also see [built-in slots](#) and [built-in intents](#).

```
1 {  
2   "intents": [  
3     {  
4       "intent": "GetNewFactIntent"  
5     },  
6     {  
7       "intent": "AMAZON.HelpIntent"  
8     },  
9     {  
10      "intent": "AMAZON.StopIntent"  
11    },  
12    {  
13      "intent": "AMAZON.CancelIntent"  
14    }  
15  ]  
16 }
```

**Custom Slot Types**  
Custom slot types to be referenced by the Intent Schema and Sample Utterances  
For general information about custom slots, see [Custom Slot Types](#).  
Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

**Sample Utterances\***  
Phrases end users say to interact with the skill. For better results, provide as many samples as you can. Note that you must select three of these to use as your Example Phrases on the Description tab.  
For more information, see [Sample Utterances](#).

```
1 GetNewFactIntent a fact  
2 GetNewFactIntent a space fact  
3 GetNewFactIntent tell me a fact  
4 GetNewFactIntent tell me a space fact  
5 GetNewFactIntent give me a fact  
6 GetNewFactIntent give me a space fact  
7 GetNewFactIntent tell me trivia  
8 GetNewFactIntent tell me a space trivia  
9 GetNewFactIntent give me trivia  
10 GetNewFactIntent give me a space trivia  
11 GetNewFactIntent give me some information  
12 GetNewFactIntent give me some space information  
13 GetNewFactIntent tell me something  
14 GetNewFactIntent give me something  
15
```

Save Submit for Certification **Next**

# Go get your Lambda Function's ARN

## Instructions

Remember the ARN we noted down earlier?  
Time to put it to use, go grab it now!

Lambda > Functions > yourFunctionHere

ARN - arn:aws:lambda:us-east-1:674783520507:function:yourFunctionHere

yourFunctionHere

Qualifiers ▼

Actions ▼


GetNewMLHFact ▼

Test

Configuration

Triggers

Monitoring



Alexa Skills Kit

alex-appkit.amazon.com

ⓘ

To configure your Alexa skill, go to the [Alexa Developer Portal](#).

Delete

+ Add trigger

↻ Refresh triggers

▶ View function policy



# Connect Your Lambda Function to Your Skill.

## Instructions

Select AWS Lambda ARN and the Region closest to you.  
Paste the ARN from your function in the text box.

**Global Fields**  
These fields apply to all languages supported by the skill.

**Endpoint**

Service Endpoint Type: ☒ **AWS Lambda ARN (Amazon Resource Name)** ⓘ ☐ HTTPS

*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](#)

[How to integrate AWS Lambda with Alexa](#)

**Default**

`arn:aws:lambda:us-east-1:674783520507:function:yourFunctionHere`

**Provide geographical region endpoints?**  
(Optional) ⓘ ☐ Yes ☒ No

**Account Linking**

Do you allow users to create an account or link to an existing account with you? ☐ Yes ☒ No

[More info about Account Linking](#)

[Tips for successful Account Linking](#)

# Test your Skill

## Instructions

Type one of your sample utterances into the service simulator to see how Alexa would respond.

Service Simulator

Use Service Simulator to test your lambda function: `arn:aws:lambda:us-east-1:541236211448:function:mlhFactSkill2`

Note: Service Simulator does not currently support testing audio player directives and customer account linking.

**Text** JSON

Enter Utterance

Give me an M. L. H. fact

Ask Major League Hacking (MLH) Facts Reset

**Lambda Request**

```
1 {
2   "session": {
3     "sessionId": "SessionId.8c3da3e1-6bbd-4fc6-
4     "application": {
5       "applicationId": "amzn1.ask.skill.2f6b3b6
6     },
7     "attributes": {},
8     "user": {
9       "userId": "amzn1.ask.account.AEFWLK23SWYI
10    },
11    "new": true
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "EdwRequestId.c3ba0723-8da3-45
16    "locale": "en-US",
```

**Lambda Response**

```
1 {
2   "version": "1.0",
3   "response": {
4     "outputSpeech": {
5       "type": "SSML",
6       "ssml": "<speak> Here's your fact: Hack
7     },
8     "card": {
9       "content": "Hackers created over 12,000
10      "title": "Major League Hacking (MLH) Fa
11      "type": "Simple"
12    },
13    "shouldEndSession": true
14  },
```

Listen

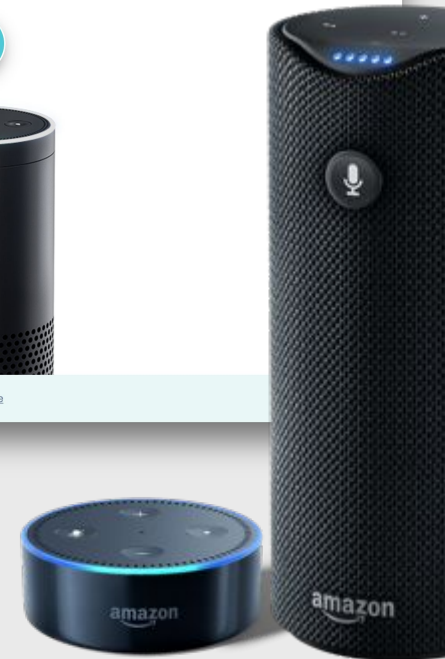
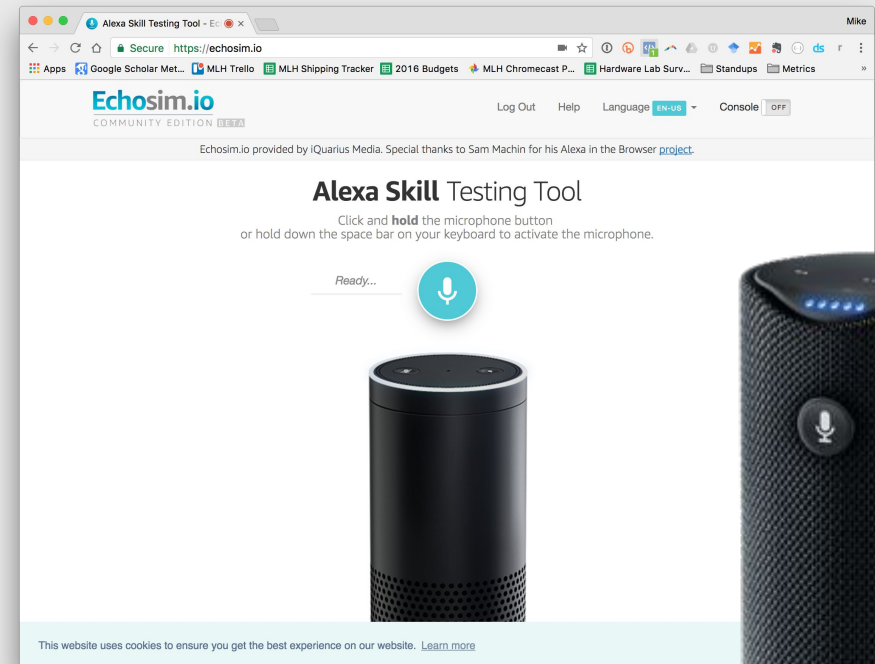
*Hear what Alexa would say  
by clicking "Listen".*

# Try out Your Skill!

Head to [EchoSim.io](https://echosim.io) to try out your fact.

*"Alexa, open [Your Invocation Name]."*

If you want to try your skill on an Alexa Device, pair it with your account first!




# Troubleshooting Your Skill

If you aren't getting a valid response, check the following:

1. Do you have the right ARN copied from your Developer Portal/Skill into your your Lambda function?
2. Are you calling the right invocation name?
3. Are you saying launch, start or open?
4. Are you sure you have no other skills in your accounts with the same invocation name?

# Table of Contents

- 0. Welcome to MLH Localhost
- 1. Introduction to Alexa & Voice UIs
- 2. Developing for Alexa
- 3. Build Your First Alexa Skill
-  4. Review & Quiz
- 5. Next Steps



## *Let's recap quickly...*


- 1 Voice User Interfaces allow us to physically separate ourselves from devices.
- 2 Amazon Alexa makes it easy for you to create apps (skills) that utilize Voice User Interfaces.
- 3 Alexa takes care of speech recognition and context so you can focus on making a great app.

# What did you learn today?

We created a fun quiz to test your knowledge and see what you learned from this workshop.

[\*\*http://mlhlocal.host/quiz\*\*](http://mlhlocal.host/quiz)

# Table of Contents

- 0. Welcome to MLH Localhost
- 1. Introduction to Alexa & Voice UIs
- 2. Developing for Alexa
- 3. Build Your First Alexa Skill
- 4. Review & Quiz
-  5. Next Steps



# Get credit & freebies from Amazon!

<http://mlhlocal.host/alexa-promo>

Our friends at Amazon Web Services run promotions in certain areas to reward hackers for making skills for the Amazon Alexa Platform.

**Fill this out to get credit!**

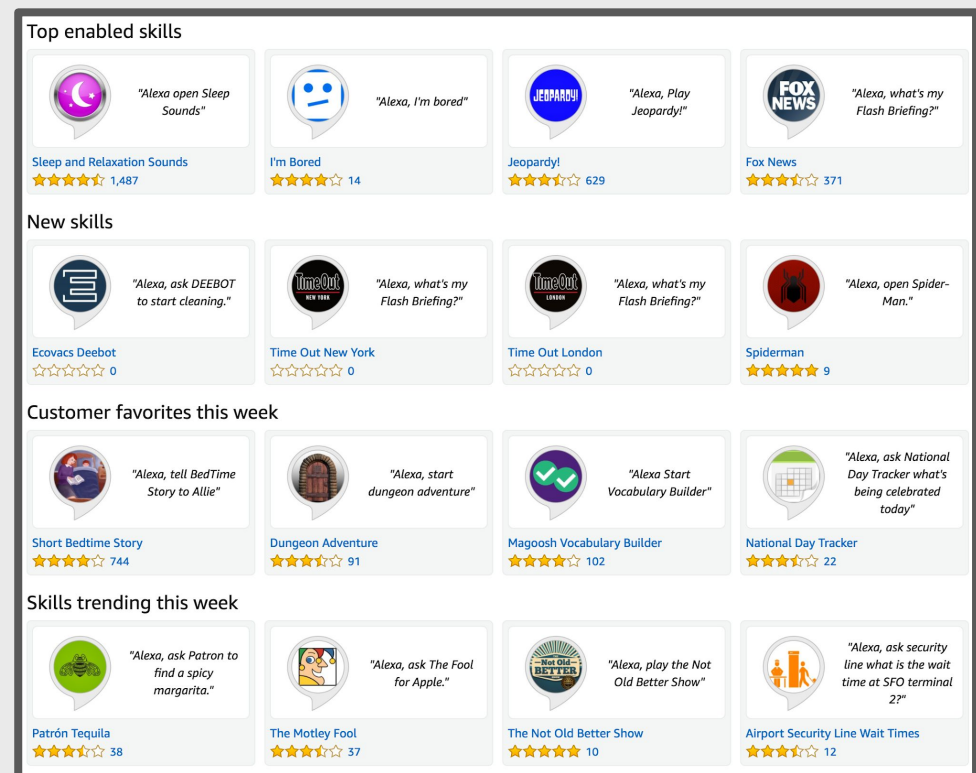


# Next Steps: Publish your Skill!

You have a working skill, the next step is to publish it on the Alexa Skill Marketplace.

## Instructions

1. Complete the “Publishing Information” and “Privacy and Compliance” sections on the developer portal.
2. Ensure that your skill meets the basic requirements.

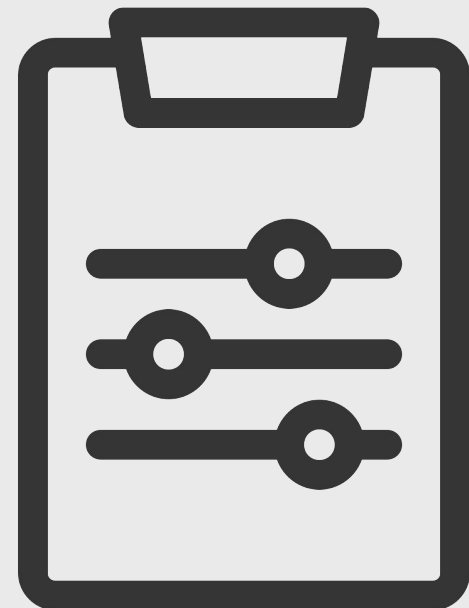


# Next Steps: Use Slots to Customize Further

Slots enable you to pass additional data to your Lambda function. Use slots to return a fact in a specific category (Ex. "facts about my education").

## Instructions

1. *Read up about Slots and how to use them to pass data to your function.*
2. *Categorize your facts into a handful of groups.*
3. *Customize your function to return a fact based on the user specified category*



# Learning shouldn't stop when the workshop ends...



**Check your email for access to:**

- These workshop slides
- Practice problems to keep learning
- Deeper dives into key topics
- Instructions to join the community
- More opportunities from MLH!

Workshop

# Hacking with Amazon Alexa