

TECHSTERS

May 12th, 2017

*Controlling the world with our voice.
Innovating with the Amazon Alexa.*



Welcome to the Future!

It wasn't long ago that the idea of talking to computers and using them to learn more about our world and control the things around us felt like science fiction. Today, new innovations like Amazon's Alexa, Google Home, and others are making our ability to interact with the world almost limitless.

Around the world, software engineers, innovators, and technology enthusiasts are making Alexa smarter by creating new Skills. People just like us are making Alexa better and more powerful every day.

Today we'll be learning more about how to build new Alexa Skills and how we can use technology in our physical world to bring the entire internet within reach our voice.



Today's Agenda

- 3:00 – 3:30pm **Arrival / Snack**
- 3:30 – 3:45pm **Kick Off**
- 3:45 – 3:55pm **Be a Coding Master Chef**
- 3:55 – 4:05pm **Logging in to our Amazon Developer Accounts**
- 4:05 – 4:40pm **Exercise 1: “Alexa, ask my cookbook what is your favorite ice cream?”**
- 4:40 – 5:00pm **Exercise 2: “Alexa, ask my cookbook for cooking trivia!”**
- 5:00 – 6:00pm **Exercise 3: “Alexa, ask my cookbook for a recipe containing chocolate.”**
- 6:00 – 6:45pm **Dinner**
- 6:45 – 7:15pm **Exercise 4: “Alexa, ask my Kitchen for a recipe containing _____. ”**
- 7:15 – 7:30pm **Closing**

Interacting with Alexa

Before we begin building with Alexa, let's start by turning on the Dot and seeing how it works.

Enable Skills:

Choose a Skill to enable on your device-

Education/Reference Skills:

- [Sushi Facts](#)
- [Fruit Facts](#)
- [Bacon Facts](#)
- [Taco Facts](#)

Food and Drink Skills:

- [Food Network](#)
- [Restaurant Explorer](#)

Building your first Lambda

“Alexa, ask my cookbook what is your favorite ice cream?”

The Amazon Web Services (AWS) console provides access to all of the great developer tools and features that Amazon makes available to developers, inventors, and creators. These tools include the ability to process large (or small) amounts of data, use machine learning to study patterns, and much more.

Today we'll be using what Amazon calls “Lambdas” which are tiny bits of computer instructions that we run when specific events take place. The great thing about Lambdas is that we don't have to care anything about the computers where these instructions run - they make it easy for developers. We'll use Lambdas to run instructions that react to things we speak to our Amazon Echo. We will write the instructions that will use what we say to our Echo to figure out how to respond.



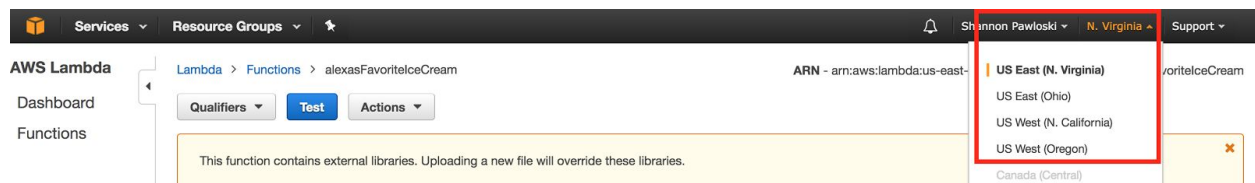
Each Lambda we write is “**event-driven**” which means it runs when something else happens to trigger it. There are lots of things that we use every day that are “event-driven”.

- *Microwave cooking* - the microwave turns on when you hit start, then turns off when the timer runs out. The microwave reacts to these two events.
- *Refrigerator ice maker* - the ice maker on the front of the refrigerator waits for the button to be pressed before putting ice in your cup. Pressing and releasing the button triggers events that the refrigerator expects and reacts to.

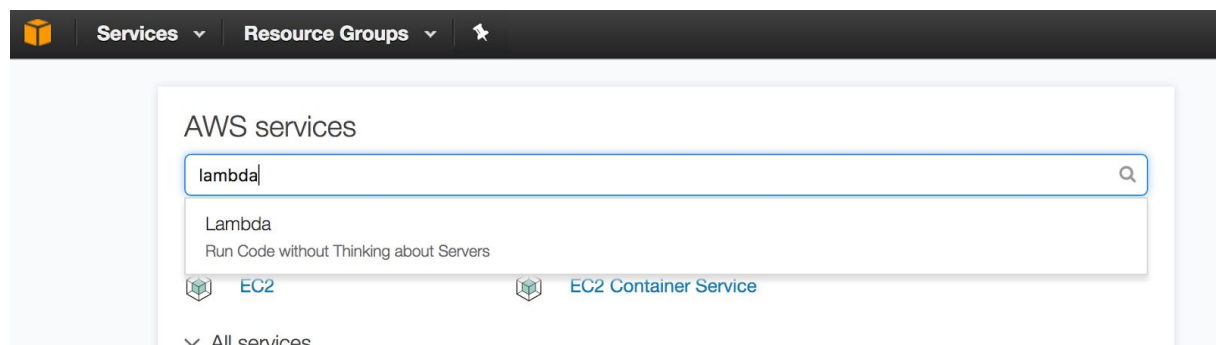
- *Amazon Alexa* - Alexa waits for keywords before it starts listening to what you say. It then processes what it hears, reacting to both of these two events. Today, we'll teach it how to react to even more voice events!

Each Lambda we write is “**serverless**”, which means we don’t have to know which computer (or “server”) is running the instructions, or even where that computer is. All we have to do is create the instructions, and AWS takes care of the rest.

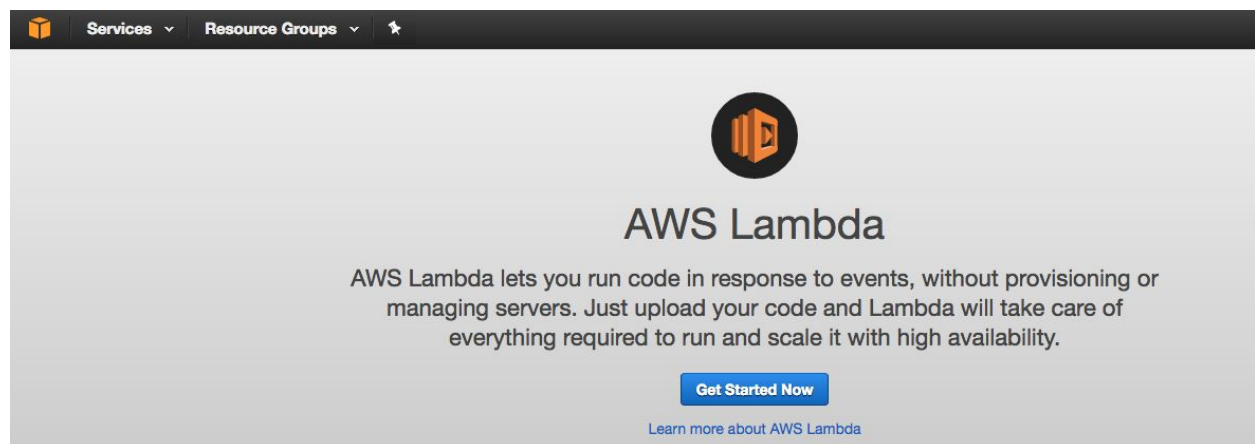
0) All of the work we will do today requires us to be in the US East (N. Virginia) region. To make sure you’re in the correct region, look on the upper right side of the AWS console near your account name. Click the dropdown, and make sure US East (N. Virginia) is selected.



1) From the AWS console, **search for ‘Lambda’**. This will take us to the console for creating, editing, and testing our own Lambda instructions.



2) Click the ‘Get Started Now’ button to reach the Alexa Skill Dashboard.



3) There are so many things that Lambdas can do, and using them with the Alexa is only the beginning! For today's activity, we will start with a sample project referred to as a "blueprint." **Select the "alexa-skill-kit-sdk-factskill" blueprint to proceed.**

Select blueprint



Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

Select runtime ▼

Filter

« < Viewing 1-9 of 89 > »

Blank Function Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard. custom	kinesis-firehose-syslog-to-json An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON. nodejs · kinesis-firehose	alexa-skill-kit-sdk-factskill Demonstrate a basic fact skill built with the ASK NodeJS SDK nodejs · alexa
batch-get-job-python27 Returns the current status of an AWS	kinesis-firehose-apachelog-to-... An Amazon Kinesis Firehose stream	cloudfront-modify-response-h... Blueprint for modifying CloudFront

Notice that if you type 'alexa' in the filter box, you will see many *other* Alexa templates. These are a great way to start out your next Alexa Skill!

4) There are lots of ways we can trigger a lambda. Since we are using Alexa, we'll want to **click on the box with the gray dashed line around it, then select 'Alexa Skills Kit'**. Then press 'Next.'

Configure triggers

You can choose to add a trigger that will invoke your function.

Lambda

Remove

Filter integrations

API Gateway

AWS IoT

Alexa Skills Kit

Alexa Smart Home

CloudFront

Cancel

Previous

Next

5) Let's give our new Lambda a name and description! Our new Alexa Skills will be all about food, pick a name that you like - we'll choose 'MyAlexaCookbook'. There are lots of different

languages that can be used to write the instructions for a Lambda. When the Lambda runs, it will need to know which language we chose when writing our instructions. Select 'Node.js 4.3' as the Runtime.

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name*	<input type="text" value="MyAlexaCookbook"/>
Description	<input type="text" value="Demonstrate a basic fact skill built with the /"/>
Runtime*	<input type="text" value="Node.js 4.3"/>

6) Now it's time to start writing some code, and create the instructions that Alexa will follow when it hears us talk! **Replace the code that is in the text box** with the code from <http://bit.ly/techstersLambda1> or retype it from below.

Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. [Learn more](#) about deploying Lambda functions.

Code entry type	<input type="text" value="Edit code inline"/>
-----------------	---

```
1  const alexa = require('alexa-sdk');
2
3  const handlers = {
4    'LaunchRequest': function () {
5      this.emit(':ask', "Welcome to your cookbook, ask me some questions and I'll try to answer!");
6    },
7    'FavoriteIceCream': function () {
8      this.emit(':ask', "I love all types of ice cream, but rocky road is my favorite!");
9    }
10 };
11
12 exports.handler = (event, context) => {
13   const handler = alexa.handler(event, context);
14   handler.registerHandlers(handlers);
15   handler.execute();
16 };
```

Once your code is entered, there are a few configuration options to select, then you'll be ready to create your new Lambda function and test! Note that the name of the role does not matter; we're just using 'myCookbook' so that we know what it's for.

Lambda function handler and role

Handler* ⓘ

Role* ⓘ

Lambda will automatically create a role with permissions from the selected policy templates. Note that basic Lambda permissions (logging to CloudWatch) will automatically be added. If your function accesses a VPC, the required permissions will also be added.

Role name* ⓘ

Policy templates ⓘ

8) **Press ‘Next’** to review your function, then **press ‘Create function.’** You’ve just created your first Lambda function!

9) You’ll notice in the top right-hand corner, a long row of text that starts with “arn” This is the address of your new Lambda. We’ll use this later when we connect Alexa to the new behavior we’ve created in our Lambda. No need to write it down or try to remember it! We’ll come back for it later.

ARN - `arn:aws:lambda:us-east-1:095207682014:function:MyAlexaCookbook`

10) Now that we have our Lambda finished, let’s give it a quick test run from within the console before we start talking to it from our Alexa. **Click the blue or ‘Test’ button at the top of the page.**

[Lambda](#) > [Functions](#) > MyAlexaCookbook



We’ll be prompted with the type of event trigger we want to send to our Lambda. We’ll want to scroll down and **pick the ‘Alexa Start Session’ choice.**

Qualifiers ▾ **Test** Actions ▾

Input test event ⓘ

Use the editor below to enter an event to test your function with. You can edit the event again by choosing **Configure test event** in the Actions list. Note that changes to the event will only be saved locally.

Sample event template ▾

```

1 {
2   "session": {
3     "new": true
  }
}
```


At the bottom of the window, **click 'Save and Test'**. We'll know it works when we see the "Execution result: succeeded" message, and can find the message that we'll expect Alexa to speak. Great job!

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



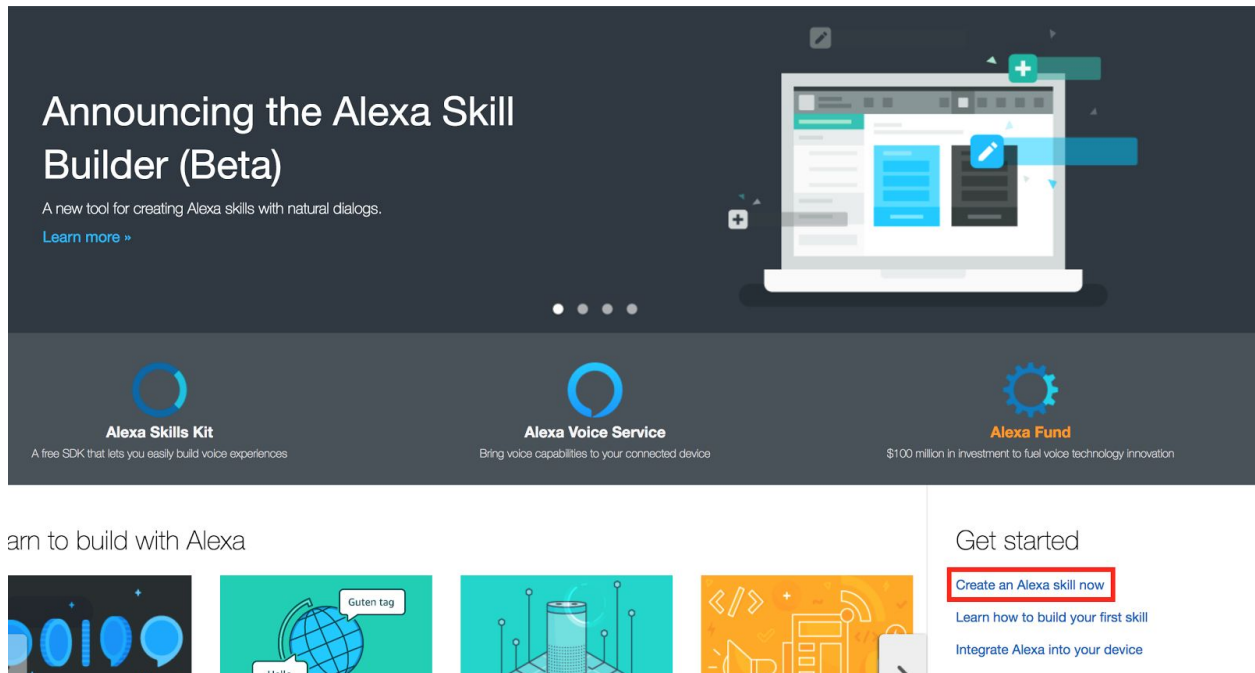
The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "type": "SSML",
      "ssml": "<say> Welcome to your cookbook, ask me some questions and I'll try to answer! </say>"
    }
  }
}
```


Creating an Alexa Skill

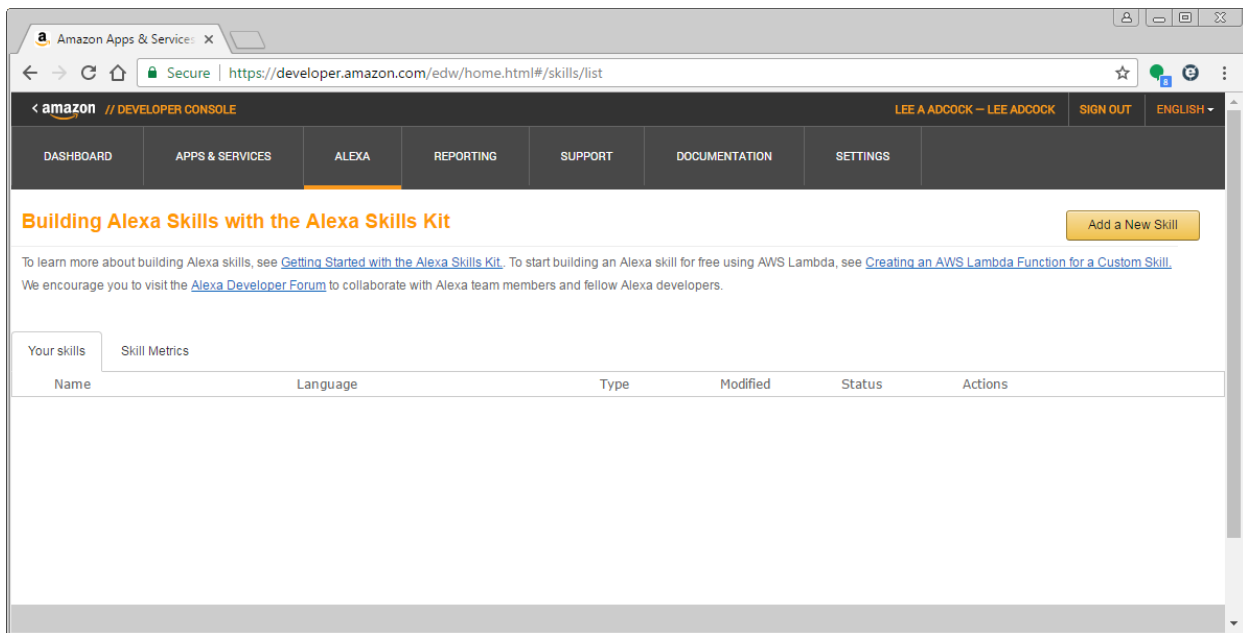
1) Keeping the current browser tab open, in a new tab head back to the Developer website to create the Alexa Skill: <https://developer.amazon.com/alexa>

2) Choose **'Create an Alexa Skill now'** from the list on the middle right-side of the page.



3) You may be presented with two options, adding new Skills to Amazon Alexa, or adding the ability to use Alexa with new devices. For our project, we are adding new Skills, so **click the Alexa Skills Kit 'Get Started' button** if this option is available.

A page will open with the list of Skills you've created - it'll be empty now but not for long! **Click the 'Add a New Skill' button** in the top right corner.



4) Fill in the form with the information below:

Skill Type: 'Custom Interaction Model'

Language: 'English'

Name: 'MyCookbook'

Invocation Name: 'MyCookbook'

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

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.

MyCookbook

Invocation Name
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".
[Invocation Name Guidelines](#)

MyCookbook

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

Audio Player
Does this skill use the audio player directives? ☐ Yes ☒ No
[Learn more](#)

5) Click 'Save' and then 'Next' to go to the next step.

We'll need to tell Amazon the different types of behaviors that we'll be expecting our Alexa to understand. Amazon calls these different behaviors "intents". For our first try, the Lambda we created only knows how to respond to the "*FavoriteIceCream*" intent. You can **retype from the example below**, or copy and paste from <http://bit.ly/techstersIntent1>.

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": "FavoriteIceCream"  
5     }  
6   ]  
7 }  
8
```

There are lots of different ways that we might say the same thing when we talk to the Alexa, and we have to list what those different ways might be. For example, we could say "What is your favorite ice cream?" or "What ice cream is your favorite?" and expect the same answer. Amazon calls these different ways of saying something an "Utterance". In the next box, let's enter our Intent followed by each Utterance we might expect someone to say. Be creative here: How many ways can you figure out how to say the same thing?

You can **retype from the example below**, copy and paste from <http://bit.ly/techstersUtterances1> or create your own list of Utterances. Be sure to enter this in the Utterance section and not in the Custom Slot Type section.

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 FavoriteIceCream what is your favorite ice cream  
2 FavoriteIceCream what is Alexa's favorite ice cream  
3 FavoriteIceCream what ice cream is your favorite  
4 FavoriteIceCream do you like ice cream  
5 FavoriteIceCream do you eat ice cream  
6 FavoriteIceCream do you have a favorite ice cream  
7 FavoriteIceCream is ice cream your favorite  
8 FavoriteIceCream is ice cream your favorite dessert  
9 FavoriteIceCream have you had ice cream
```

6) Click 'Next' to enter the Configuration. Click the check boxes for AWS Lambda ARN and North America.

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](#)

Pick a geographical region that is closest to your target customers: ⓘ

☒ North America

☐ Europe

North America

7) Now that we've gotten our Alexa Skill mostly set up, we'll need to link our Skill with the Lambda we created earlier. Remember that long Lambda address we pointed out earlier? Go back to the first browser tab where we created the Lambda, **copy the ARN of your Lambda function and paste it into the ARN endpoint** in the Skill configuration.

Lambda > Functions > alexasFavoritelceCream

ARN - arn:aws:lambda:us-east-1:843658851123:function:alexasFavoritelceCream

Qualifiers ▼

Test

Actions ▼

Congratulations! Your Lambda function "alexasFavoritelceCream" has been successfully created and configured with as a trigger. You can now click on the "Test" button to input a test event and test your function. ✕



#RVATech

Techsters 2017

Page 11 of 26

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](#)

Pick a geographical region that is closest to your target customers: ⓘ

☒ North America

☐ Europe

North America

arn:aws:lambda:us-east-1:843658851123:function:a

Account Linking

Do you allow users to create an account or link to an existing account with you?

☐ Yes ☒ No

[Learn more](#)

8) Click 'Next' and we're done!

English (U.S.) ⓘ Add New Language

Skill Information ✓

Interaction Model ✓

Configuration ✓

Test ✓

Publishing Information ⚙

Privacy & Compliance ⚙

ⓘ Please complete the Interaction Model tab to start testing this skill.

☒ Enabled

This skill is enabled for testing on your account. ⓘ

Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

You will be able to see your skill in the Skills tab in Alexa App and you can enable the skill and start testing.

After completing your testing please submit the skill for certification. If it passes Amazon's testing and certification process, it will become available to Alexa end users.

The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality by asking Alexa, **ask about my future**

Voice Simulator

Hear how Alexa will speak a response entered in plain text or SSML. [Learn more about supported SSML tags.](#)

For example: Here is a word spelled out: <say-as interpret-as="spell-out">hello</say-as>.

Listen ▶

9) Now that our new Lambda and Alexa Skill are ready, we can test both through the Amazon developer console, and on our actual Alexa. From the "test" step, **try entering some of the Utterances you chose** in the earlier step into the "Service Simulator" and **click the 'Ask' button**.

Service Simulator

Use Service Simulator to test your lambda function: `arn:aws:lambda:us-east-1:095207682014:function:MyAlexaCookbook` ▼

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

Text

JSON

Enter Utterance

what is your favorite icecream?

Ask MyCookbook

Reset

Lambda Request

```
1 {
2   "session": {
3     "sessionId": "SessionId.b9ed8b52-c570-46a5
4     "application": {
5       "applicationId": "amzn1.ask.skill.57f6a4
6     },
7     "attributes": {},
8     "user": {
9       "userId": "amzn1.ask.account.AGMPVK23QZV
10    },
11    "new": true
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "FdwRequestId.90514f8a-3594-4i
```

Lambda Response

```
2  "version": "1.0",
3  "response": {
4    "outputSpeech": {
5      "type": "SSML",
6      "ssml": "<say> I love all types of i
7    },
8    "reprompt": {
9      "outputSpeech": {
10        "type": "SSML",
11        "ssml": "<say> undefined </say>"
12      }
13    },
14    "shouldEndSession": false
15  }
```

Listen

What happens? Does Alexa respond as you would expect?

- If it doesn't work, try to troubleshoot. Is the error message telling you anything? Did you skip a step of this guide?
- When it does work, try entering some more utterances!

10) Now try to change Alexa's favorite ice cream flavor. Go back to your code on the Lambda and see if you can change the flavor. Now what happens?

Kitchen Trivia!

“Alexa, ask my cookbook for cooking trivia!”

A lot of the work we’ve done in the previous exercise is plumbing, connecting different pieces of technology to get information to flow from one place to another. In this case, our data is flowing from the Alexa on our table, through our Alexa Skill, through our Lambda, and then back out to our Alexa in the form of a spoken response.

With this plumbing done, we can start adding new abilities to our Alexa by modifying our existing Alexa Skill and Lambdas. For each new ability we create for our Alexa, we’ll need to make changes in three places:


- Give our new Intent a name by modifying our Alexa Skill
- Decide on the different Utterances that we want to trigger our Lambda
- Add new instructions to our Lambda to define our behavior



Let’s jump in and start adding new abilities to our Alexa Skill! This time, we’ll create the ability for our new Skill to tell us fun facts.

1) Go back to the list of the Alexa Skills you’ve created. This was empty before but now your new Skill is listed. You can get here the same way as before: Go to <https://developer.amazon.com/alexa> then **select ‘Create an Alexa Skill now’**.

2) Click the **‘Edit’** action link next to your Skill.

Name	Language	Type	Modified	Status	Actions
 MyCookbook	English (U.S.)	Custom	4/4/17	Development ⓘ	Metrics Edit Delete

3) **Update the list of Intents** to add our new “KitchenFacts” Intent.

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": "FavoriteIceCream"
5     },
6     {
7       "intent": "KitchenFacts"
8     }
9   ]
10 }
```

4) Next let’s **update the list of Utterances** with the new ways that you can call this new Intent. See if you can come up with more ideas of ways you might ask for one of our fun facts.

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 FavoriteIceCream What is your favorite ice cream
2 FavoriteIceCream What's your favorite ice cream
3 FavoriteIceCream Do you have a favorite ice cream
4 FavoriteIceCream What is the best ice cream
5 FavoriteIceCream What ice cream is best
6 FavoriteIceCream Which ice cream is best
7 KitchenFacts Tell me a kitchen fact
8 KitchenFacts Tell me a random kitchen fact
9 KitchenFacts What is a random kitchen fact
10 KitchenFacts What's a random kitchen fact
11 KitchenFacts Say a random kitchen fact
```

5) Now all we need to do is **update our Lambda** to know how to respond when someone uses our new Intent. You can retype the example below, or copy and paste from <http://bit.ly/techstersLambda2>.

```
1 const alexa = require('alexa-sdk');
2
3 const handlers = {
4   'LaunchRequest': function () {
5     this.emit(':ask', "Welcome to your cookbook, ask me some questions and I'll try to answer!");
6   },
7   'FavoriteIceCream': function () {
8     this.emit(':ask', "I love all types of ice cream, but rocky road is my favorite!");
9   },
10  'KitchenFacts': function () {
11    const facts = [
12      'Pringles once had a lawsuit trying to prove that they weren't really potato chips.',
13      'Ripe cranberries will bounce like rubber balls.',
14      'An average ear of corn has an even number of rows, usually 16.',
15      'Apples belong to the rose family, as do pears and plums.',
16      'One of the most popular pizza toppings in Brazil is green peas.'
17    ];
18    const factNumber = Math.floor(Math.random() * facts.length);
19    this.emit(':tell', facts[factNumber]);
20  }
21 };
22
23
24 exports.handler = (event, context) => {
25   const handler = alexa.handler(event, context);
26   handler.registerHandlers(handlers);
27   handler.execute();
28 };
```

6) **Save our changes** and test out your new behavior on the Alexa!



Find That Recipe!

“Alexa, ask my cookbook for recipes with _____!”

Sometimes we want our Skill to be able to understand similar Utterances that are more flexible. For example, asking for recipes with a specific ingredient would be a lot of work if we had to list every possible question in our list of Utterances. Instead we can use what Amazon calls “Slots”.


Slots are placeholders for multiple options, they allow us to use a single Utterance to handle a lot of different but similar questions. Using Slots means we have to make slight changes to how we write our Intent, our Utterances, and the behavior inside our Lambda.

Let's jump in and start adding these new abilities to our Alexa Skill!

1) Go back to the list of the Alexa Skills you've created. This was empty before but now your new Skill is listed. You can get here the same way as before, going to <https://developer.amazon.com/alexa> then selecting 'Create an Alexa Skill now'.



2) Click the 'Edit' action link next to your Skill.

Name	Language	Type	Modified	Status	Actions
 MyCookbook	English (U.S.)	Custom	4/4/17	Development ⓘ	Metrics Edit Delete

3) Let's **update the list of Intents** to add our new “RecipeFinder” Intent. You'll notice for the first time we are also telling Amazon about a new Slot that we'll use in our Utterances. Let's call this “Ingredient” and it will be one of the values from our “LIST_OF_INGREDIENTS” that we will define next. You can retype the below example or copy from <http://bit.ly/techstersIntent3>.

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      "intent": "KitchenFacts",
2    },
3  ],
4  {
5    "intent": "RecipeFinder",
6    "slots": [
7      {
8        "name": "Ingredient",
9        "type": "LIST_OF_INGREDIENTS"
10     }
11   ]
12 }
```

4) Since we are using a special Slot that we are creating for the ingredient, we'll have to define this “LIST_OF_INGREDIENTS” so Alexa knows what to expect us to say. This is done in the

next step, which we previously skipped over. **Create your custom Slot type like the example, then click the 'Add' button.** You can add any other ingredients you think of to this list.

Custom Slot Types (Optional)

Custom slot types to be referenced by the Intent Schema and Sample Utterances. For general information about custom slots, see [Custom Slot Types](#).

Enter Type

LIST_OF_INGREDIENTS

Enter Values

Values must be line-separated

```
1 chocolate
2 vanilla
3 strawberries
4 almond
5
6
7
8
9
10
```

Cancel

Add

5) Next let's **update the list of Utterances** with the new ways that you can call this new Intent. We can put {Ingredient} in our Utterance any place we expect someone to say one of the items from our ingredient list. For example, "What is a recipe that includes {Ingredient}".

See if you can come up with more ideas of ways you might ask for one of our fun facts. You can retype the example below, or copy and paste from <http://bit.ly/techstersUtterances3>.

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.

```
6 FavoriteIceCream Which ice cream is best
7 KitchenFacts Tell me a kitchen fact
8 KitchenFacts Tell me a random kitchen fact
9 KitchenFacts What is a random kitchen fact
10 KitchenFacts What's a random kitchen fact
11 KitchenFacts Say a random kitchen fact
12 RecipeFinder What is a recipe that includes {Ingredient}
13 RecipeFinder What can I make with {Ingredient}
14 RecipeFinder What is something I can make with {Ingredient}
15 RecipeFinder What is a recipe I can make with {Ingredient}
16 RecipeFinder What is a dish I can make with {Ingredient}
```

5) Now all we need to do is **update our Lambda** to know how to respond when someone uses our new Intent. We can provide a different response for each ingredient. Later, if we want to get fancy, we can use what learned in the previous exercise to return a different random recipe each time an ingredient is requested. You can retype the example below, or copy and paste from <http://bit.ly/techstersLambda3>.

```

1  const alexa = require('alexa-sdk');
2
3  const handlers = {
4    'LaunchRequest': function () {
5      this.emit(':tell', "Welcome to your cookbook, ask me some questions and I'll try to answer!");
6    },
7    'FavoriteIceCream': function () {
8      this.emit(':tell', "I love all types of ice cream, but rocky road is my favorite!");
9    },
10   'KitchenFacts': function () {
11     const facts = [
12       'Pringles once had a lawsuit trying to prove that they weren't really potato chips.',
13       'Ripe cranberries will bounce like rubber balls.',
14       'An average ear of corn has an even number of rows, usually 16.',
15       'Apples belong to the rose family, as do pears and plums.',
16       'One of the most popular pizza toppings in Brazil is green peas.'
17     ];
18     const factNumber = Math.floor(Math.random() * facts.length);
19
20     this.emit(':tell', facts[factNumber]);
21   },
22   'RecipeFinder': function() {
23     const ingredient = this.event.request.intent.slots.Ingredient.value;
24     const recipes = {
25       'chocolate': 'whipped chocolate pie',
26       'vanilla': 'home made vanilla bean ice cream',
27       'strawberries': 'strawberry custard',
28       'almond': 'almond crust cheesecake'
29     };
30     this.emit(':tell', recipes[ingredient]);
31   }
32 };
33
34 exports.handler = (event, context) => {
35   const handler = alexa.handler(event, context);
36   handler.registerHandlers(handlers);
37   handler.execute();
38 };

```

6) **Save your changes and test out your new behavior on the Alexa!**

The Internet Always knows!

“Alexa, ask my cookbook for recipes with _____!”

In the last exercise, we provided a list of ingredients, and matching recipes that we listed in our Lambda. Wouldn't it be great if our responses could be smarter, using information from the Internet to lookup answers and provide a seemingly infinite number of recipe answers?

A surprising amount of Internet traffic isn't people browsing the web, but computers talking with each other. Companies around the world create “web services” which are much like web pages for computers, that provide data and allow different systems to communicate with each other over the Internet - just without pictures, colors, and cat videos.

We can use a recipe web service to request information about recipes with different ingredients, and use information over the Internet to make our Alexa Skill even smarter! Let's modify our recipe finder to use the power of web services to give better answers for more ingredients.



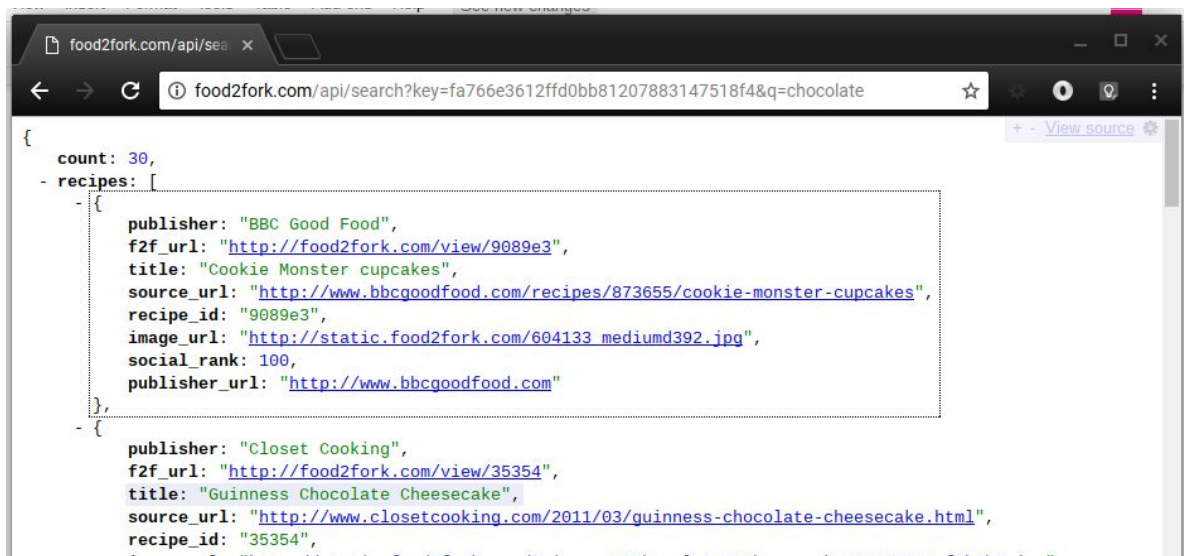
1) Since we already have an Alexa Intent that looks up recipes, let's reuse as much of it as possible. The only two things we'll want to modify are the instructions in the Lambda for this behavior, and add new ingredients to our list.

First, **go back to your Lambda** by navigating through the AWS Console.

2) The web service that we are going to use will allow us to call a special website address that includes the name of our ingredient, and get back information about recipes. This information is written in a way that makes it easy for computers to read and understand - and not as easy for us humans. You can **visit this web service in your computer's browser by entering the following address** to see what the response looks like, and what information it returns, or just look at the example below.

<http://food2fork.com/api/search?key=fa766e3612ffd0bb81207883147518f4&cq=chocolate>

Notice that in this example, we included 'chocolate' in the website address, and the information that comes back is all about recipes that use chocolate. Although we are specifically looking for the recipe name to speak back from our Alexa Skill, the web service response also comes back with information on who published the recipe, where we can get the recipe, and a ranking of how it compares to other recipes.



3) By using this information in our Alexa Skill, we'll return better recipe suggestions than the list we entered earlier. We will be pulling from thousands of recipe ideas available on the Internet! **Update your Lambda's Recipe Finder instructions** to match what you see below. You can retype (not recommended) or just copy from <http://bit.ly/techstersLambda4> and paste into your code.

```

20     this.emit(':tell', facts[factNumber]);
21   },
22   'RecipeFinder': function() {
23     const ingredient = this.event.request.intent.slots.Ingredient.value;
24     var handler = this;
25     var http = require('http');
26
27     var url='http://food2fork.com/api/search?key=fa766e3612ffd0bb81207883147518f4&cq='+ingredient;
28
29     http.get(url, function(res) {
30       var body = '';
31       res.on('data', function(chunk) { body += chunk; });
32       res.on('end', function() {
33         const recipeList = JSON.parse(body);
34         const recipeNumber = Math.floor(Math.random() * recipeList.count);
35         handler.emit(':tell', recipeList.recipes[recipeNumber].title);
36       });
37     });
38   }
39 };
40
41 exports.handler = (event, context) => {

```

4) Save your changes, and test out your new behavior on the Alexa! *If you keep asking it for recipes with the same ingredient, you will get new and different answers each time!*

5) There are web services that can provide and do all sorts of things. Some are available for free and others require a small fee each time they are used. What sort of useful information would you like to be able to get from your Alexa? Can you find a web service that can provide

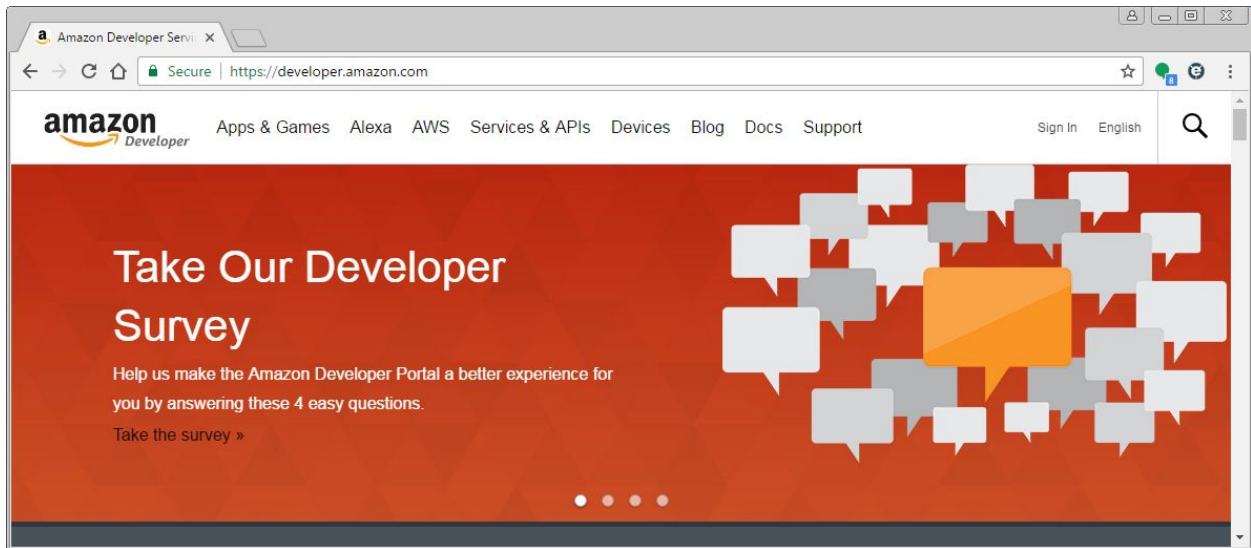
the information needed to make an Alexa Skill? **Take a look at the websites below** for web services you can use!

<http://market.mashape.com>

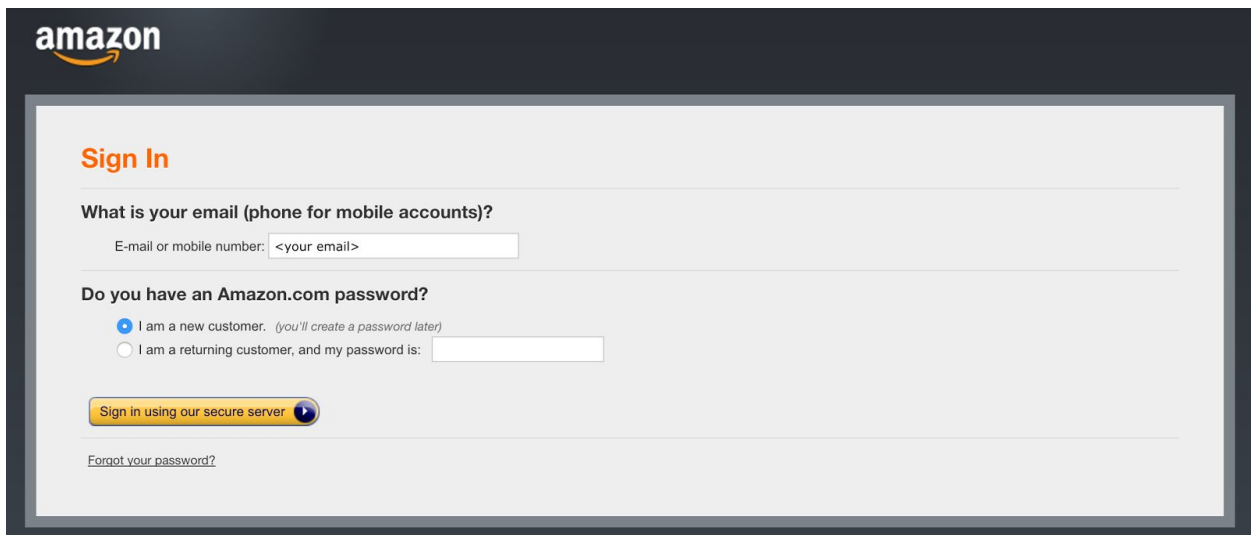
Appendix: Setting Up A Developer Account

Today we'll be developing new Skills for the Amazon Alexa. We'll be real software developers, writing real software, making our home devices smarter. To get started, the first step is registering as a developer with Amazon. This will give us access to all of the great developers tools, servers, and automation that Amazon makes available.

- 1) Our first step is to go to the Amazon developer website: <https://developer.amazon.com>

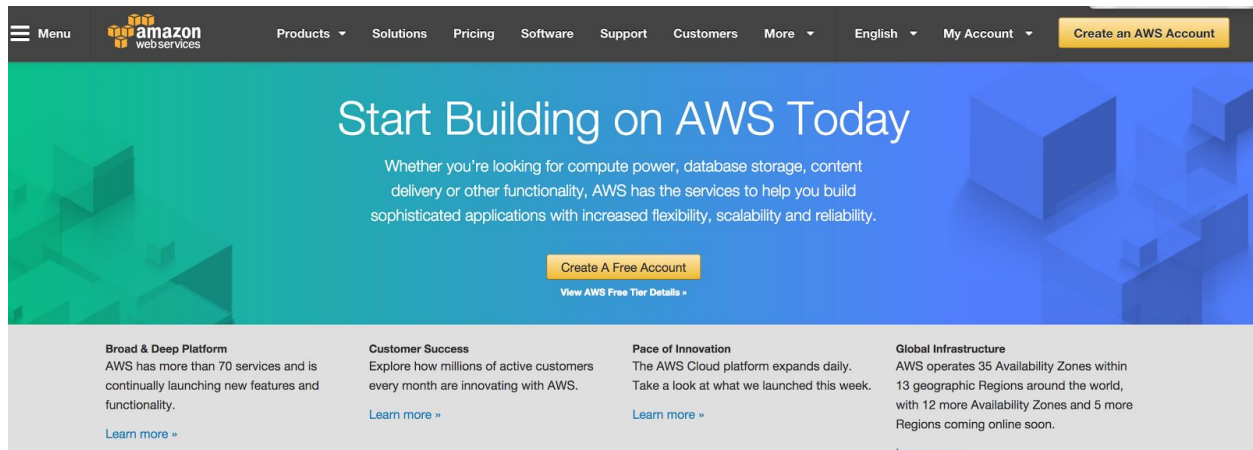


- 2) In the top menu, click on the 'Alexa' item.
- 3) Click Sign In
- 4) Enter your email and then select the button for 'I am a new customer'.



5) Once we have logged in as an Amazon developer, we will have access to everything Amazon has to offer for creating and innovating on their developer platform. Let's go see all of the tools that are available by going to the Amazon Web Services console: <https://aws.amazon.com>

6) Amazon provides "Amazon Web Services" (AWS) as a set of products which help developers create new things. Now that we have a developer account, let's get access to Amazon Web Services by clicking on the 'Create an AWS Account' button on the upper right of the page



FAQ

- 1)
- 2)

TODO:

Exercise 2: Facts Skill

"Alexa, ask Kitchen Facts for a cooking trivia fact!"

Fact based Skills can be a one to many algorithm.

- a. **One-to-one Algorithm:** Draw on whiteboard the ice flavor exercise, and show that for each person there is one correct response- it was a one-to-one algorithm. Ask for other examples where there is only one correct response:
 - i. How much money do you have?
 - ii. What color are my eyes?
 - iii. What is the current temperature in Richmond, Virginia?

- b. **One to Many Algorithm:** In the next exercise, we're going to look at a one-to-many algorithm, where one question can have many different answers like name a color. What are other examples?
- Name a type of candy
 - Name a recipe with chocolate chips
 - What's a city that gets lots of rain?

- 1) Just like in the previous exercise, go to the AWS Console and open up Lambda. Create a new function. This time, name it 'kitchenFacts.'
- 2) Copy the code from here.
- 3) On the Developer Portal, create a new Skill. Name it 'Kitchen Facts' and use 'Kitchen Facts' or something similar as the invocation name.

If all of this is working, try entering other commands into the utterance box, or add to your Lambda function. Here are some ideas:

- Add more facts
- Add more intents
- Add more utterances for each intent

