How To Build an SMS Messaging Skill for Alexa

Wouldn't it be great to send a message to a friend or family member when you're thinking of them? Or maybe you just want to remind them to grab some milk on the way home from the gym? This new Alexa skill template helps you to build something that does exactly that.

In addition to the tools that we always recommend for Alexa skill development (AWS Lambda, Alexa Skills Kit (ASK), DynamoDB, and the ASK SDK), we will also be introducing you to another Amazon Web Services technology: Amazon Simple Notification Service (SNS). We provide the business logic, error handling, and help functions for your skill, you just need to provide the data and credentials.

For this example, we will be general messaging skill that sends four kinds of messages to users: Reminders, Miss You, Love You, and Hello. The user of this skill will be guided through a conversation to help them customize and send messages to their friends like these:

- “Wanted to send you a reminder about tacos! From, Alice”
- “You just popped into my head...thought I'd reach out to say hello! From, Steve”
- “I was just thinking about you, and wanted you to know I love you! Love, Jeff”

You will be able to customize all of the messages that this skill can produce, so you can create your own themed messaging skill around your favorite TV shows, movies, or anything else that would make an interesting experience!

After completing this tutorial, you'll know how to do the following:

- **Create a messaging skill** - This tutorial will walk Alexa skills developers through all the required steps involved in creating a skill that sends SMS messages.
- **Understand the basics of VUI design** - Creating this skill will help you understand the basics of creating a working Voice User Interface (VUI) while using a cut/paste approach to development. You will learn by doing, and end up with a published Alexa skill. This tutorial includes instructions on how to customize the skill and submit for certification. For guidance on designing a voice experience with Alexa you can also watch this video.
- **Use JavaScript/Node.js and the Alexa Skills Kit to create a skill** - You will use the template as a guide but the customization is up to you. For more background information on using the Alexa Skills Kit please watch this video.
- **Manage state in an Alexa skill** - As the user moves through our skill, we modify their state to track their progress, and manage which intents should be executed based on that state. Read more about state management here.
- **DynamoDB integration** - Because this skill behaves like a conversation, we also need to persist the user's data as we move through it. DynamoDB provides an excellent (and easy) way to do this.
- **Intent handoffs** - Sometimes, we want a user to complete a specific intent before they move on to the next one. This sample skill provides several examples of using the emitWithState() function to move a user from one intent to another.

- **Response randomization** - In order to simulate a real conversation, we also randomize all of Alexa's responses so that the skill doesn't appear to follow the same script every time. The user goes through the same process each time, but what Alexa says varies with each experience.
- **Card creation** In addition to sending a recipient an SMS message, we also create a nice visual card for our user as a reminder and confirmation of the message they sent in their Alexa app.
- **Get your skill published** - Once you have completed your skill, this tutorial will guide you through testing your skill and sending your skill through the certification process so it can be enabled by any Alexa user. You may even be eligible for some Alexa swag!

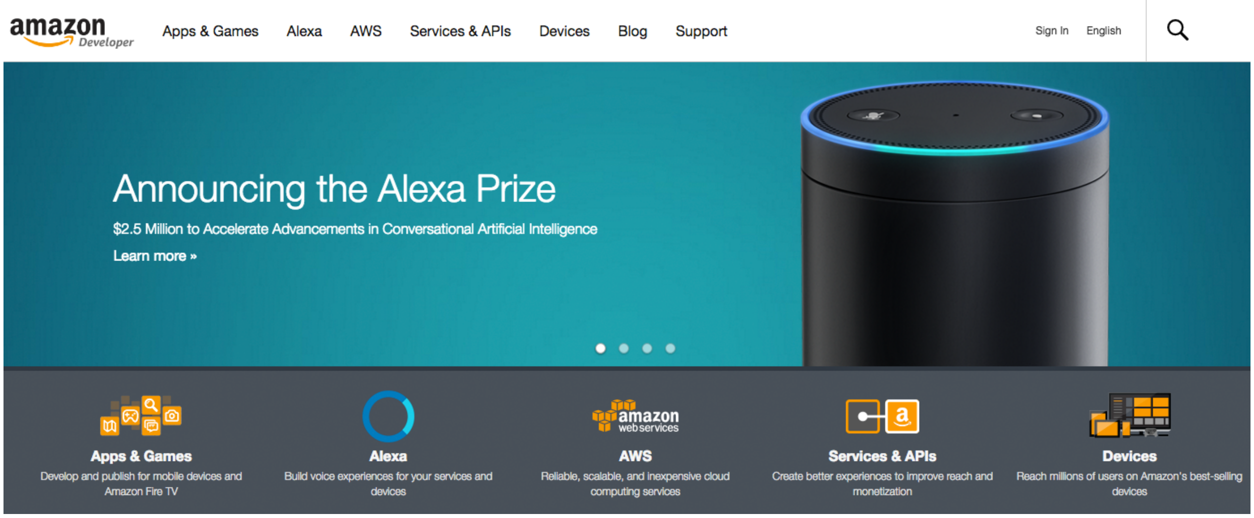
Get started and build your first - or next - Alexa skill today.

Let's Get Started

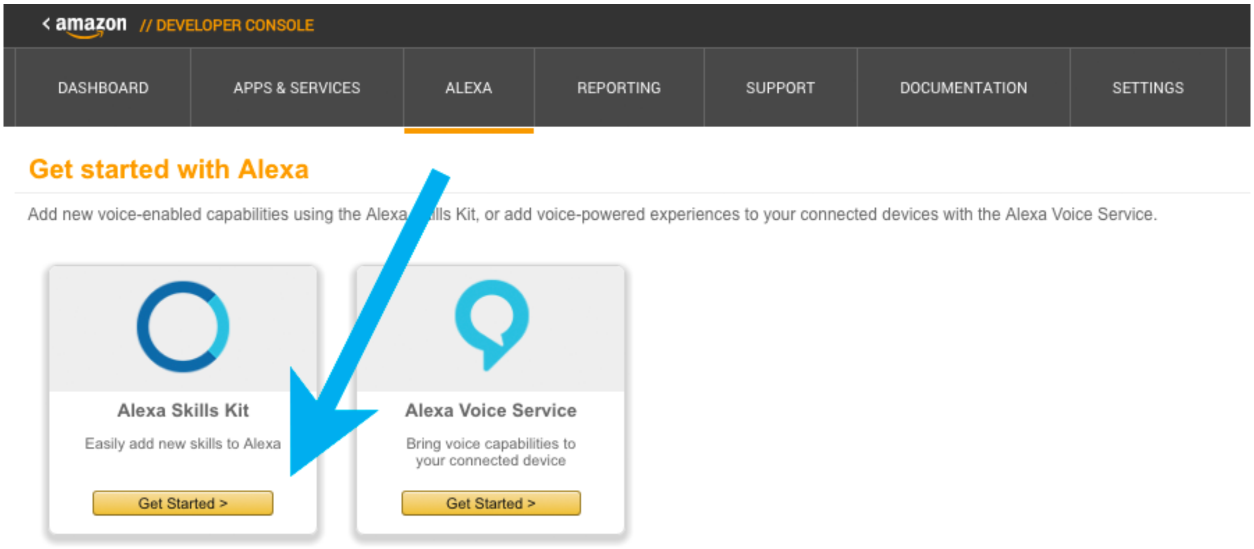
Step 1. Setting up Your Alexa Skill in the Developer Portal

Skills are managed through the Amazon Developer Portal. You'll link the Lambda function you created above to a skill defined in the Developer Portal.

1. Navigate to the Amazon Developer Portal. Sign in or create a free account (upper right). You might see a different image if you have registered already or our page may have changed. If you see a similar menu and the ability to create an account or sign in, you are in the right place.


amazon Developer
Apps & Games
Alexa
AWS
Services & APIs
Devices
Blog
Support
Sign In
English
Announcing the Alexa Prize
$2.5 Million to Accelerate Advancements in Conversational Artificial Intelligence
Learn more »
Apps & Games
Develop and publish for mobile devices and Amazon Fire TV
Alexa
Build voice experiences for your services and devices
amazon web services
AWS
Reliable, scalable, and inexpensive cloud computing services
Services & APIs
Create better experiences to improve reach and monetization
Devices
Reach millions of users on Amazon's best-selling devices

2. Once signed in, navigate to Alexa and select **"Getting Started"** under Alexa Skills Kit.



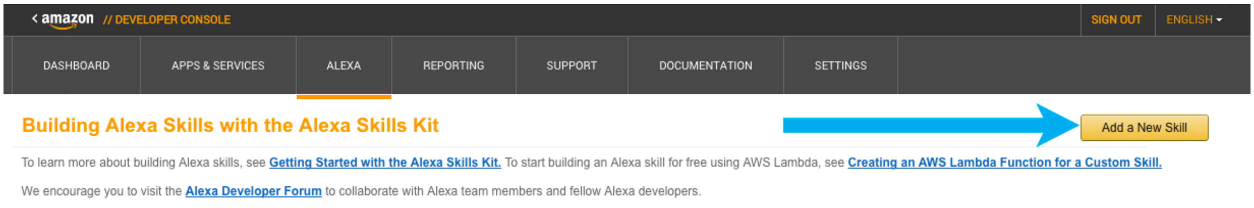
Get started with Alexa

Add new voice-enabled capabilities using the Alexa Skills Kit, or add voice-powered experiences to your connected devices with the Alexa Voice Service.

Easily add new skills to Alexa

Bring voice capabilities to your connected device

3. Here is where you will define and manage your skill. Select **"Add a New Skill"**



Building Alexa Skills with the Alexa Skills Kit

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

We encourage you to visit the Alexa Developer Forum to collaborate with Alexa team members and fellow Alexa developers.

4. There are several choices to make on this next page, so we will cover each one individually.

   i. Choose the language you want to start with. You can go back and add all of this information for each language later (this template is designed for US English, UK English, and German), but for this tutorial, we are working with "English (U.S.)"

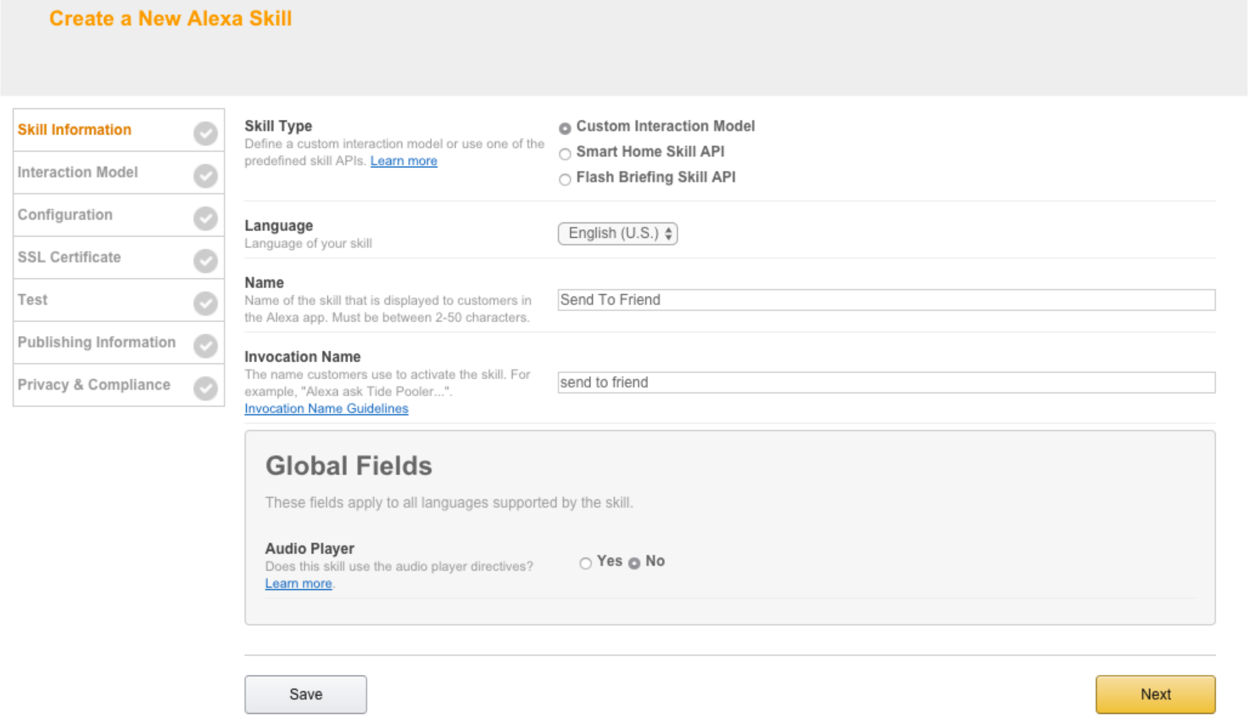
   ii. Make sure the radio button for the Custom Interaction Model is selected for “Skill Type”.

   iii. Add the name of the skill. Give your skill a name that is simple and memorable, like "Send To Friend." The name will be the one that shows up in the Alexa App (and now at amazon.com/skills) when users are looking for new skills. (Obviously, don't use "Send To Friend". Use a name that describes the kinds of messages you plan to use for your skill.)

   iv. Add the invocation name. This is what your users will actually say to start using your skill. We recommend using only two or three words, because your users will have to say this every time they want to interact with your skill.

   v. Under "Global Fields," select "no" for Audio Player, as our skill won't be playing any audio.

   vi. Select **Next**.


Create a New Alexa Skill

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.)

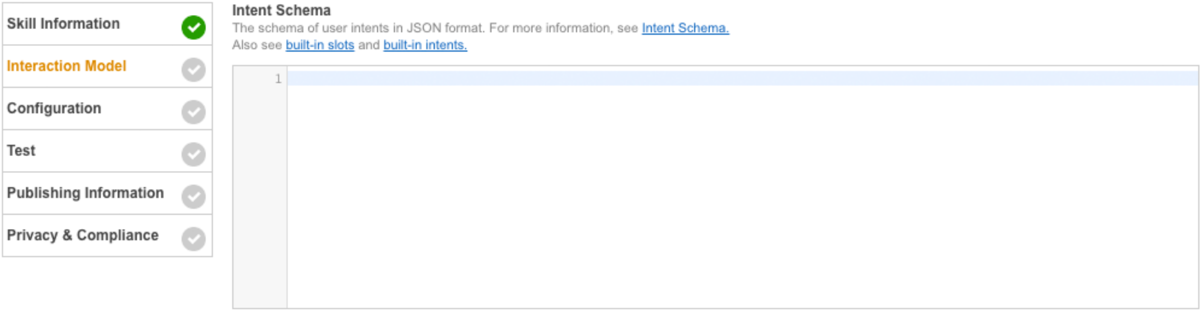
Name
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.
Send To Friend

Invocation Name
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...".
Invocation Name Guidelines
send to friend

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

Audio Player
Does this skill use the audio player directives?
Learn more.
○ Yes ● No

5. Next, we need to define our skill's interaction model. Let's begin with the intent schema. In the context of Alexa, an intent represents an action that fulfills a user's spoken request.



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.

6. Review the Intent Schema below. This is written in JSON and provides the information needed to map the intents we want to handle programmatically. Copy this from the intent schema in the GitHub repository here.

   Below you will see a collection of intents that we expect our users to indicate by voice. These are all mechanisms for collecting data from our user, but with this skill, there aren't multiple ways to enter this skill. Each time the user starts the skill, they start at the beginning of the conversation, by saying something like:

   "Alexa, open Send to Friend"

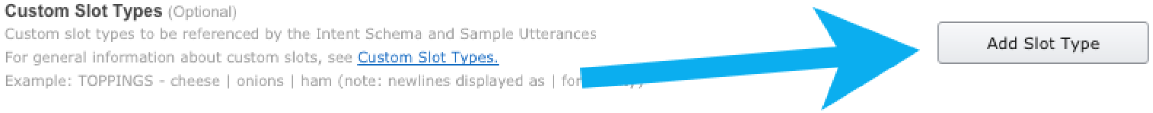
   To collect data from our users, we have to use slots. Slots are predefined data types that we expect the user to provide. This is not a closed list (like an enum), so you must anticipate that you will receive values that are not in your slot value list. For example, when the user needs to provide a phone number, we can specify an AMAZON.NUMBER slot, and they can dictate a phone number to Alexa. This data also becomes training data for Alexa's Natural Language Understanding (NLU) engine. You will see how this works more clearly when we define our sample utterances below.

   For the ReminderMessageIntent, the user will be providing something that they want to remind their contact about, like "Remind Tabitha about video games." The value "video games" is one we want to capture from our user's statements. For this, we use a custom slot, and populate it with as many nouns as we can provide.

```
{
    "intents": [
        {"intent": "AddPhoneIntent", "slots":[{"name": "phonenumber", "type": "AMAZON.NUMBER"}]},
        {"intent": "AddNameIntent", "slots":[{"name": "firstname", "type": "AMAZON.US_FIRST_NAME"}]},
        {"intent": "ReminderMessageIntent", "slots":[{"name": "noun", "type": "NOUN"}]},
        {"intent": "HelloMessageIntent"},
        {"intent": "MissYouMessageIntent"},
        {"intent": "LoveYouMessageIntent"},
        {"intent": "AMAZON.StartOverIntent"},
        {"intent": "AMAZON.RepeatIntent"},
        {"intent": "AMAZON.HelpIntent"},
        {"intent": "AMAZON.YesIntent"},
        {"intent": "AMAZON.NoIntent"},
        {"intent": "AMAZON.StopIntent"},
        {"intent": "AMAZON.CancelIntent"}
    ]
}
```

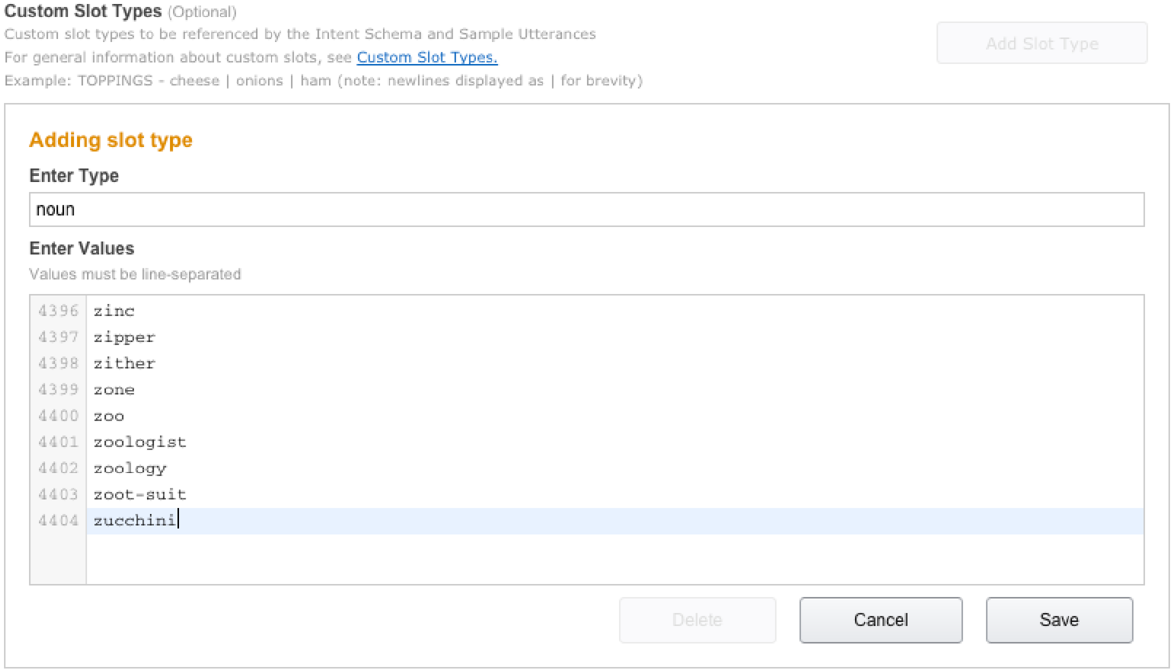
You can see that we have defined seven different built-in intents: Yes, No, Help, Stop, Cancel, Start Over and Repeat. These are built-in intents that we can use for common commands our users will indicate. For more on the use of built-in intents, go here.

7. Next, we need to define our custom slot that we called "noun." To do this, click the "Add Slot Type" button.



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.
Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for

8. For the "Enter Type" box, type the word "noun." For the "Enter Values" box, copy the contents of this giant list of nouns. You can always add more to this list if there are words you want to support.



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.
Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

Adding slot type

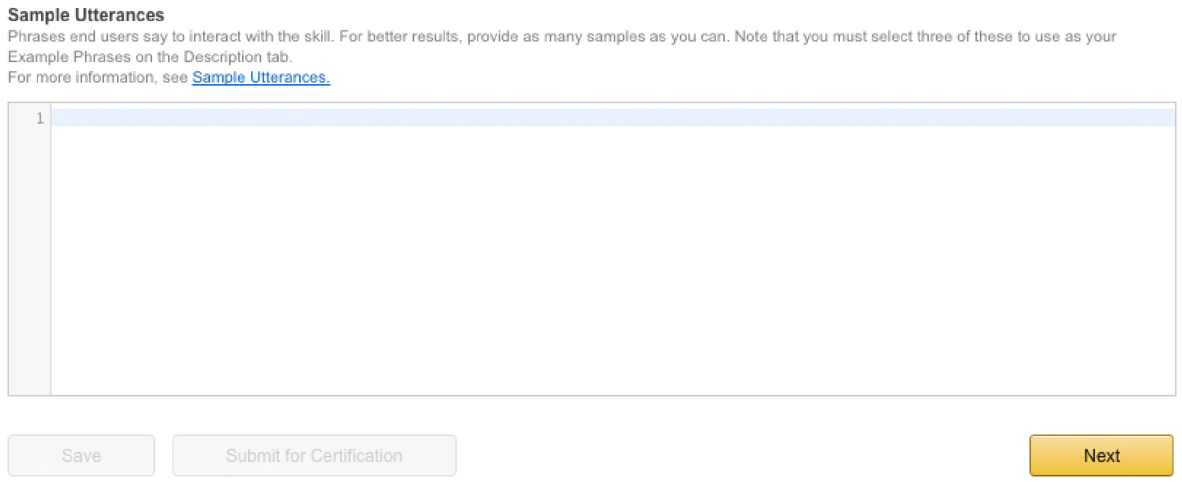
Enter Type
noun

Enter Values
Values must be line-separated

```
zinc
zipper
zither
zone
zoo
zoologist
zoology
zoot-suit
zucchini
```

Make sure to click Save before moving to the next step.

9. The next step is to build the utterance list. This is meant to be a thorough, well-thought-out list of the ways users will try to interact with your skill. You don't have to get every possible phrase, but it is important to cover a variety of utterances so that the Natural Language Understanding(NLU) engine can best interpret your user's intent.



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.

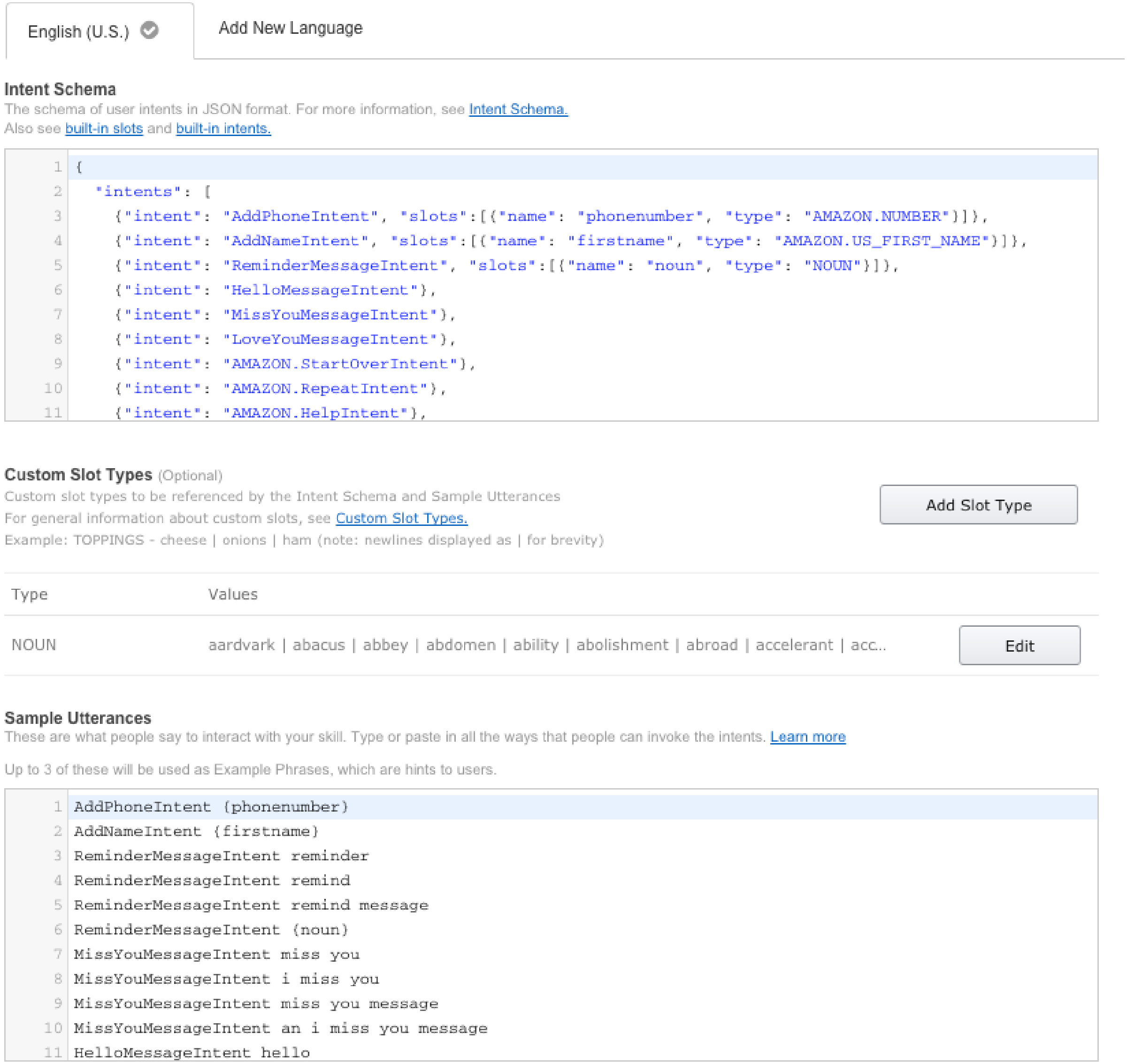
10. Given the flexibility and variation of spoken language in the real world, there will often be many different ways to express the same request. Providing these different phrases in your sample utterances will help improve voice recognition for the abilities you add to Alexa. It is important to include as wide a range of representative samples as you can -– all the phrases that you can think of that are possible in use (though do not include samples that users will never speak). Alexa also attempts to generalize based on the samples you provide to interpret spoken phrases that differ in minor ways from the samples specified.

    Now it is time to add the Utterances. Copy/paste the sample utterances from GitHub. An example of utterances is listed below.

    ```
    AddPhoneIntent {phonenumber}
    AddNameIntent {firstname}

    ReminderMessageIntent reminder
    ReminderMessageIntent remind
    ReminderMessageIntent remind message
    ReminderMessageIntent {noun}

    MissYouMessageIntent miss you
    MissYouMessageIntent i miss you
    MissYouMessageIntent miss you message
    MissYouMessageIntent an i miss you message

    HelloMessageIntent hello
    HelloMessageIntent a hello
    HelloMessageIntent hello message
    HelloMessageIntent a hello message

    LoveYouMessageIntent i love you
    LoveYouMessageIntent love you
    LoveYouMessageIntent i love you message
    LoveYouMessageIntent love you message
    ```

    As you can see in the example above, we are using our custom intents with phrases that our users might use to interact with our skill. Each example is a different way that a user might ask for that intent. AddPhoneIntent expects an AMAZON.NUMBER slot, so we have specified this in our utterances with {phonenumber}. (More information on slots can be found here.)

11. Select **Save**. You should see the interaction model being built (this might take a minute or two). If you select Next, your changes will be saved and you will go directly to the Configuration screen. After selecting Save, it should now look like this:



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.

```
{
  "intents": [
    {"intent": "AddPhoneIntent", "slots":[{"name": "phonenumber", "type": "AMAZON.NUMBER"}]},
    {"intent": "AddNameIntent", "slots":[{"name": "firstname", "type": "AMAZON.US_FIRST_NAME"}]},
    {"intent": "ReminderMessageIntent", "slots":[{"name": "noun", "type": "NOUN"}]},
    {"intent": "HelloMessageIntent"},
    {"intent": "MissYouMessageIntent"},
    {"intent": "LoveYouMessageIntent"},
    {"intent": "AMAZON.StartOverIntent"},
    {"intent": "AMAZON.RepeatIntent"},
    {"intent": "AMAZON.HelpIntent"},
```

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.
Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

Type	Values	
NOUN	aardvark | abacus | abbey | abdomen | ability | abolishment | abroad | accelerant | acc...	Edit

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

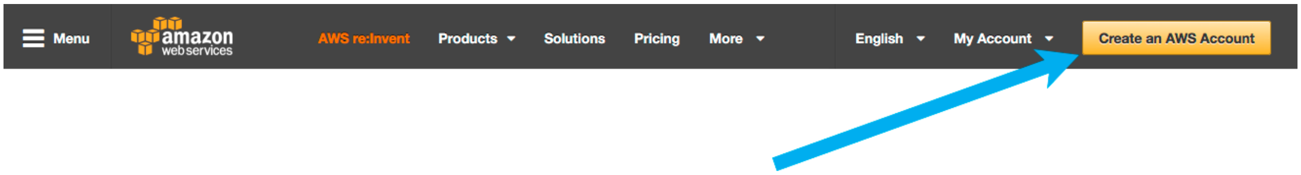
Up to 3 of these will be used as Example Phrases, which are hints to users.

```
AddPhoneIntent {phonenumber}
AddNameIntent {firstname}
ReminderMessageIntent reminder
ReminderMessageIntent remind
ReminderMessageIntent remind message
ReminderMessageIntent {noun}
MissYouMessageIntent miss you
MissYouMessageIntent i miss you
MissYouMessageIntent miss you message
MissYouMessageIntent an i miss you message
HelloMessageIntent hello
```

Next we will configure the AWS Lambda function that will host the logic for our skill.

Step 2: Creating Your Skill Logic using AWS Lambda

Create an AWS Account



Note: If you already have an AWS account, you can skip this section. Just sign in to your console.

1. Open aws.amazon.com and then choose **'Create an AWS Account'**
   i. Follow the online instructions. Do not worry about the IAM role, we will do that later.
   ii. You will need a Valid Credit Card to set up your account (note the AWS Free Tier will suffice however. You can find out more about the free tier here.)
   iii. Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.
2. Sign in to the AWS Console

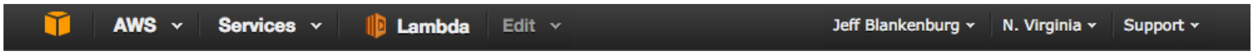
3. It can sometimes take a couple minutes for your new AWS account to go live. You will receive an e-mail when your account is active.

Create an AWS Lambda Function

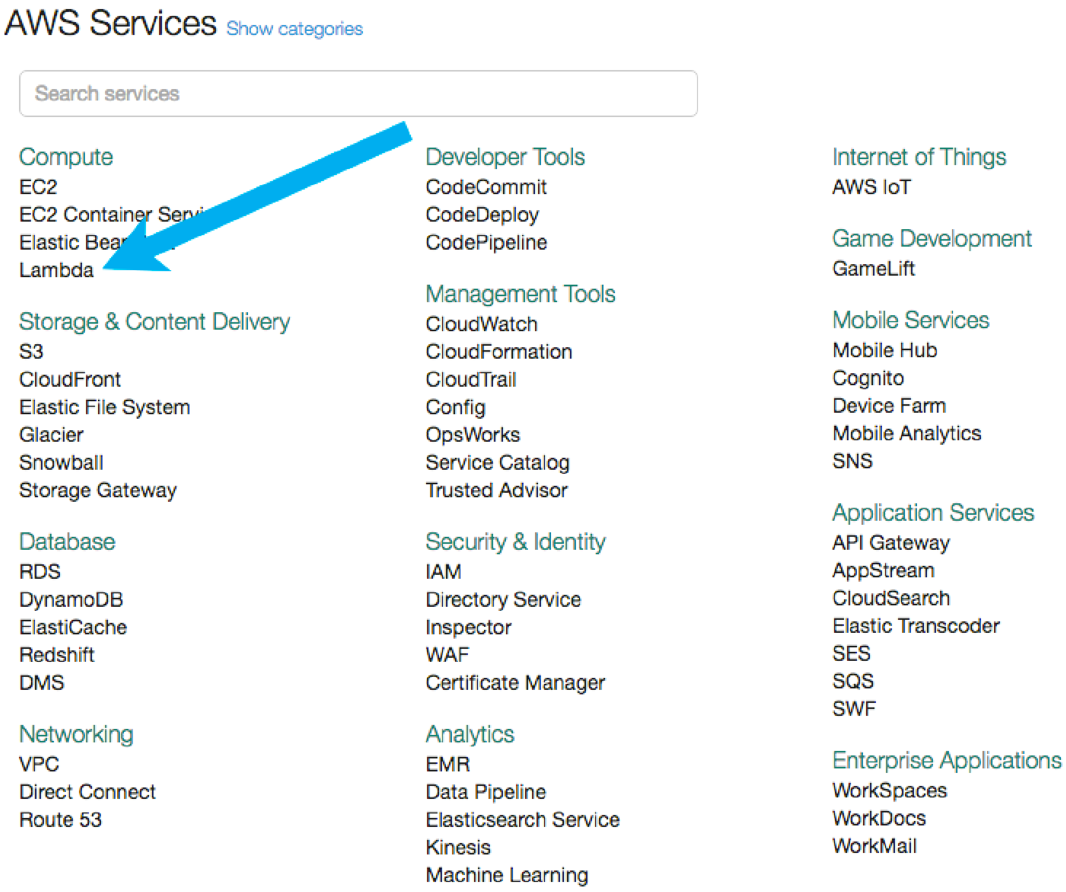
AWS Lambda lets you run code without provisioning or managing servers. The free tier of AWS Lambda provides you with 1,000,000 invocations of your function for free each month.

Note: If you are new to Lambda and would like more information, visit the Lambda Getting Started Guide.

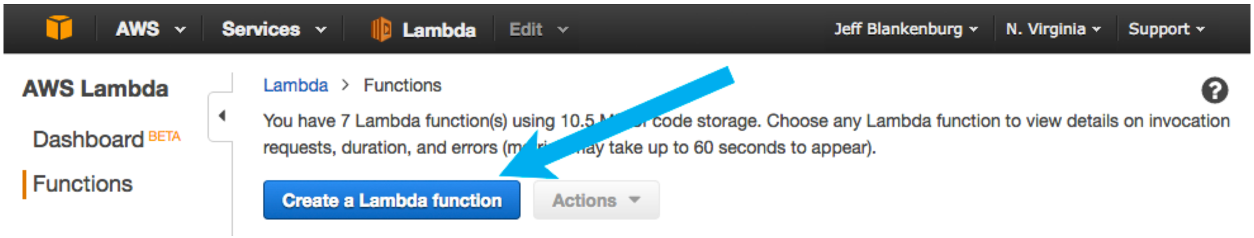
1. **IMPORTANT**: Select **US East (N. Virginia)** region, or the **EU (Ireland)** region (upper right corner). You should choose the region geographically closest to your audience. These are currently the only regions that currently support Alexa skill development.



2. Select **Lambda** from AWS Services (under Compute)

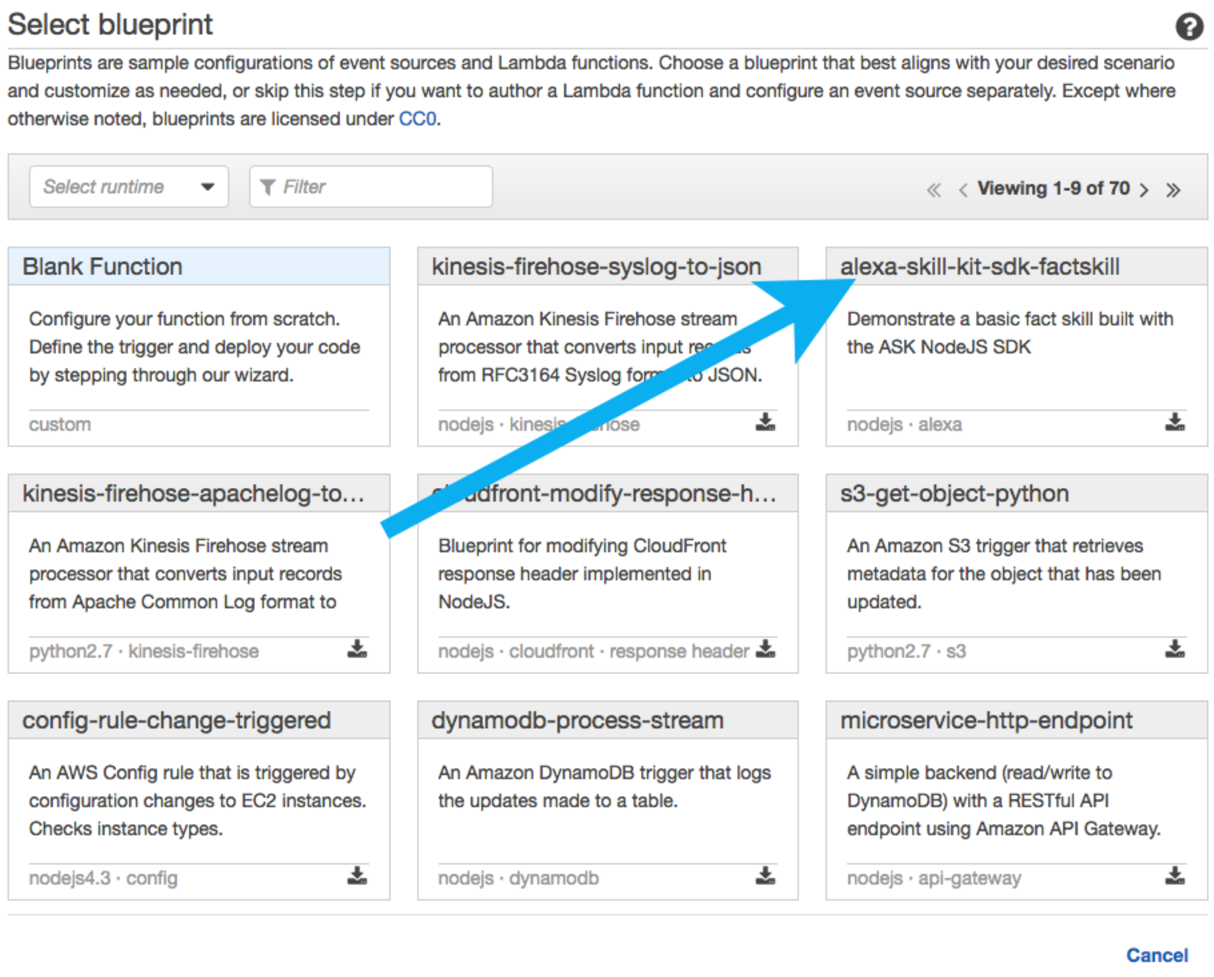


3. Select **"Create a Lambda Function"** to begin the process of defining your Lambda function.



You have 7 Lambda function(s) using 10.5 MB of code storage. Choose any Lambda function to view details on invocation requests, duration, and errors (metrics may take up to 60 seconds to appear).

4. Select the **alexa-skill-kit-sdk-factskill** option on the Select Blueprint screen.



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 70 › »

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

kinesis-firehose-apachelog-to…
An Amazon Kinesis Firehose stream processor that converts input records from Apache Common Log format to
python2.7 · kinesis-firehose

cloudfront-modify-response-h…
Blueprint for modifying CloudFront response header implemented in NodeJS.
nodejs · cloudfront · response header

s3-get-object-python
An Amazon S3 trigger that retrieves metadata for the object that has been updated.
python2.7 · s3

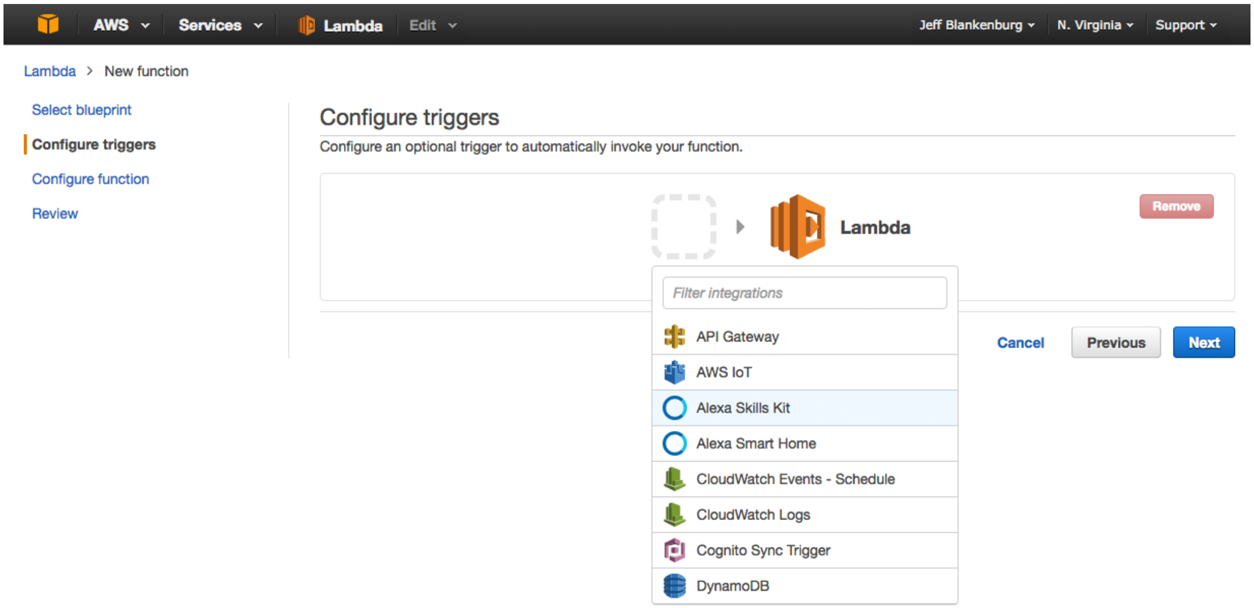
config-rule-change-triggered
An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types.
nodejs4.3 · config

dynamodb-process-stream
An Amazon DynamoDB trigger that logs the updates made to a table.
nodejs · dynamodb

microservice-http-endpoint
A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.
nodejs · api-gateway

Cancel

5. Now, you need to configure the event that will trigger your function to be called. As we are building skills with the Alexa Skills Kit, click on the gray dash-lined box and select Alexa Skills Kit from the dropdown menu. (If you don't see this option, go back to Step #1 and select the appropriate AWS region). This gives the Alexa service permission to invoke your skill's function.



AWS ▾ | Services ▾ | Lambda | Edit ▾ | Jeff Blankenburg ▾ | N. Virginia ▾ | Support ▾

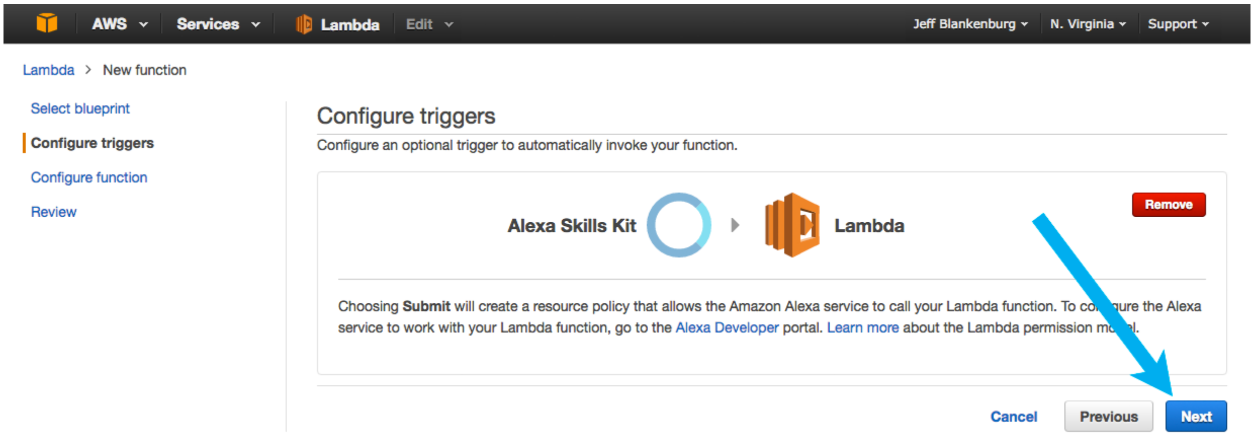
Lambda › New function

- Select blueprint
- **Configure triggers**
- Configure function
- Review

Configure triggers

Configure an optional trigger to automatically invoke your function.

6. Choose **Next** to continue.

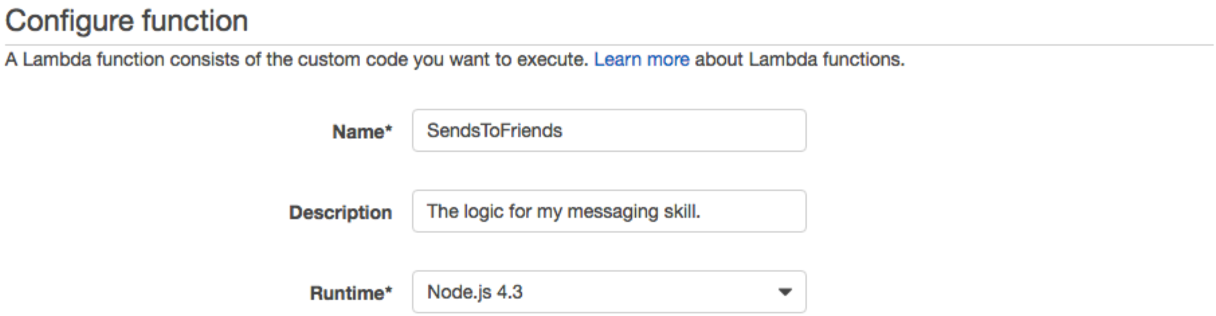


Configure triggers

Configure an optional trigger to automatically invoke your function.

Choosing **Submit** will create a resource policy that allows the Amazon Alexa service to call your Lambda function. To configure the Alexa service to work with your Lambda function, go to the Alexa Developer portal. Learn more about the Lambda permission model.

7. You should now be in the **"Configure Function"** section. Enter the Name, Description, and Runtime for your skill as in the example below. Your runtime should be "Node.js 4.3."



Configure function

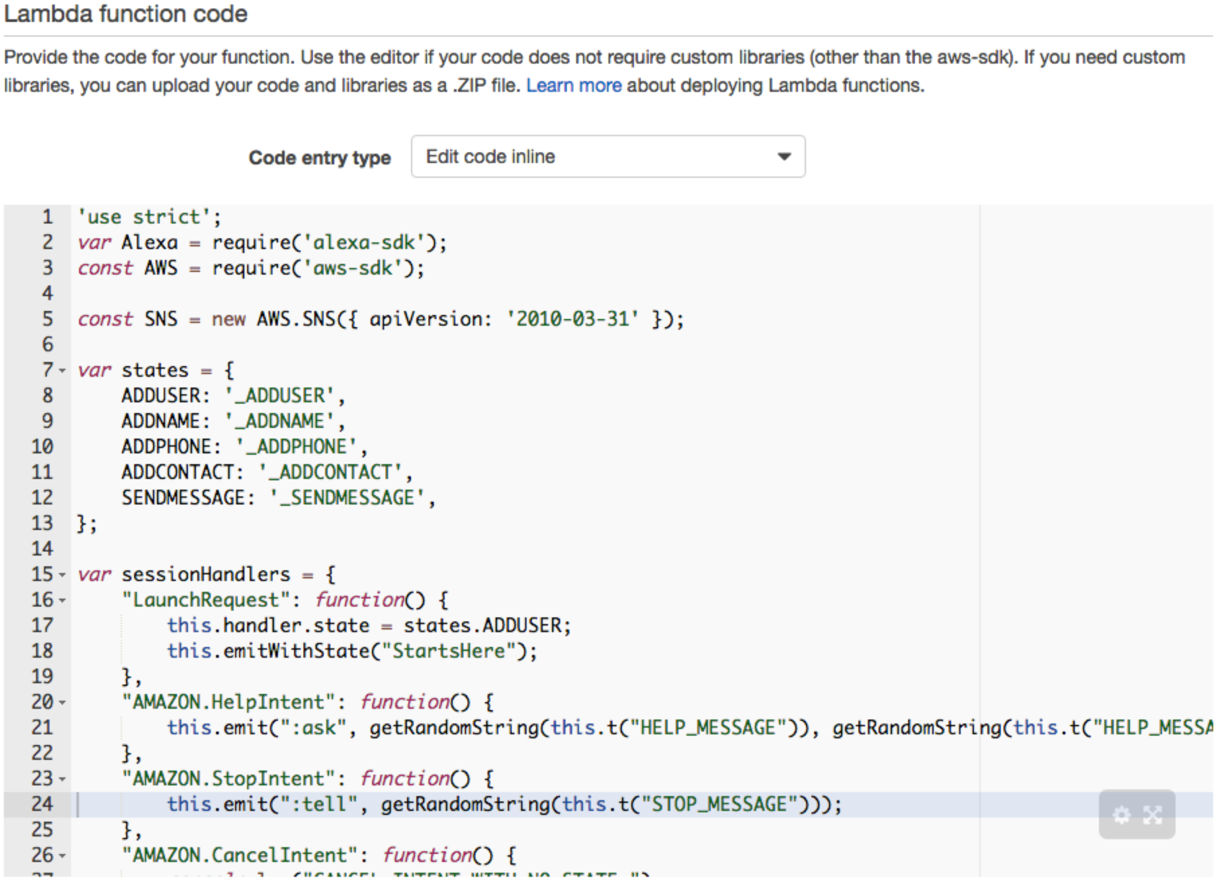
A Lambda function consists of the custom code you want to execute. Learn more about Lambda functions.

Name* SendsToFriends

Description The logic for my messaging skill.

Runtime* Node.js 4.3

8. Delete the code in the **Lambda function code** box, and replace it with the code from this tutorial. You can get this file on GitHub.



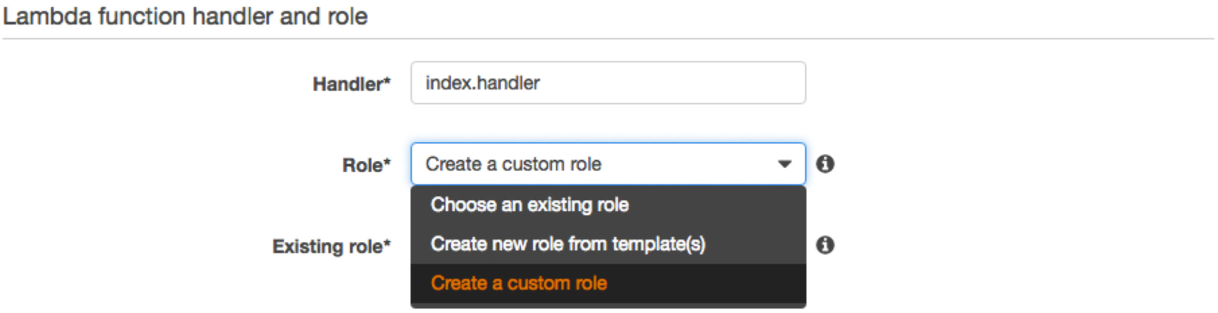
Lambda function code

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 Edit code inline

```
'use strict';
var Alexa = require('alexa-sdk');
const AWS = require('aws-sdk');

const SNS = new AWS.SNS({ apiVersion: '2010-03-31' });

var states = {
    ADDUSER: '_ADDUSER',
    ADDNAME: '_ADDNAME',
    ADDPHONE: '_ADDPHONE',
    ADDCONTACT: '_ADDCONTACT',
    SENDMESSAGE: '_SENDMESSAGE',
};

var sessionHandlers = {
    "LaunchRequest": function() {
        this.handler.state = states.ADDUSER;
        this.emitWithState("StartsHere");
    },
    "AMAZON.HelpIntent": function() {
        this.emit(":ask", getRandomString(this.t("HELP_MESSAGE")), getRandomString(this.t("HELP_MESSA
    },
    "AMAZON.StopIntent": function() {
        this.emit(":tell", getRandomString(this.t("STOP_MESSAGE")));
    },
    "AMAZON.CancelIntent": function() {
        console.log("CANCEL INTENT WITH NO STATE.")
```

9. Set your handler and role as follows:
   - Keep Handler as ‘index.handler’
   - Drop down the “Role” menu and select **“Create a custom role”**. (Note: if you have already used Lambda you may already have a ‘lambda_basic_execution’ role created that you can use.) This will launch a new tab in the IAM Management Console.



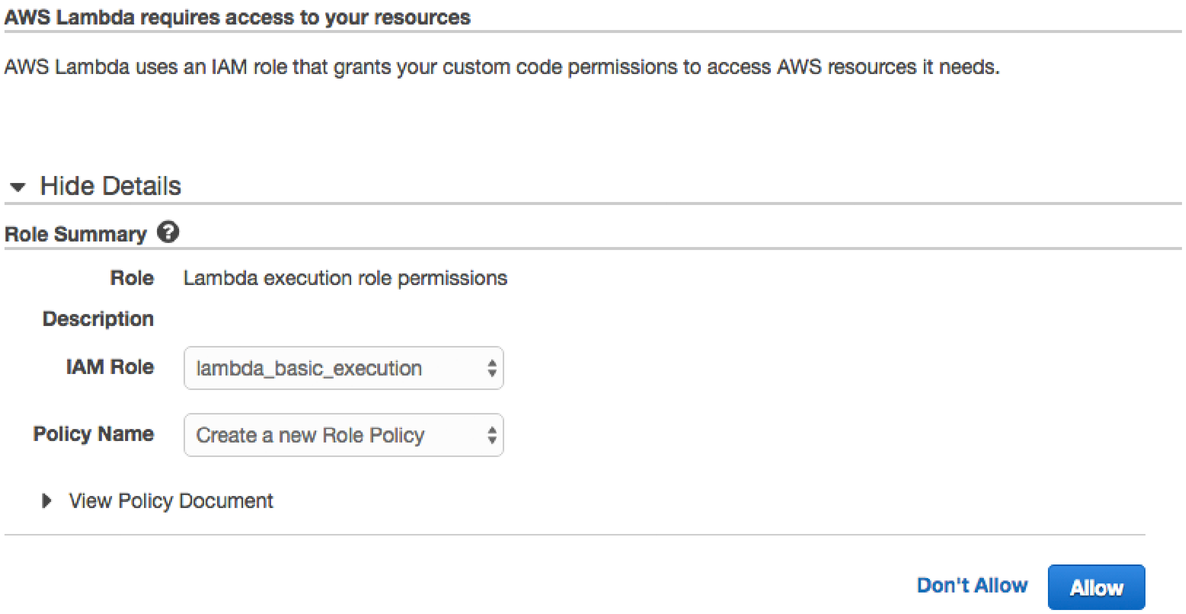
Lambda function handler and role

Handler* index.handler

Role* Create a custom role

Existing role*

10. You will be asked to set up an Identity and Access Management or “IAM” role if you have not done so. AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. The IAM role will give your Lambda function permission to use other AWS Services at runtime, such as Cloudwatch Logs, the AWS logs collection and storage service. In the Role Summary section, select "Create a new IAM Role" from the IAM Role dropdown menu. The Role Name and policy document 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

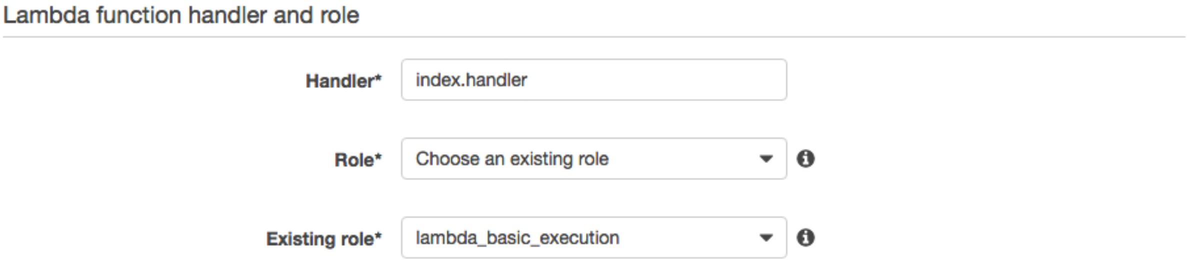
Role Summary

Role Description: Lambda execution role permissions

IAM Role: lambda_basic_execution

Policy Name: Create a new Role Policy

11. Select **“Allow”** in the lower right corner and you will be returned to your Lambda function.



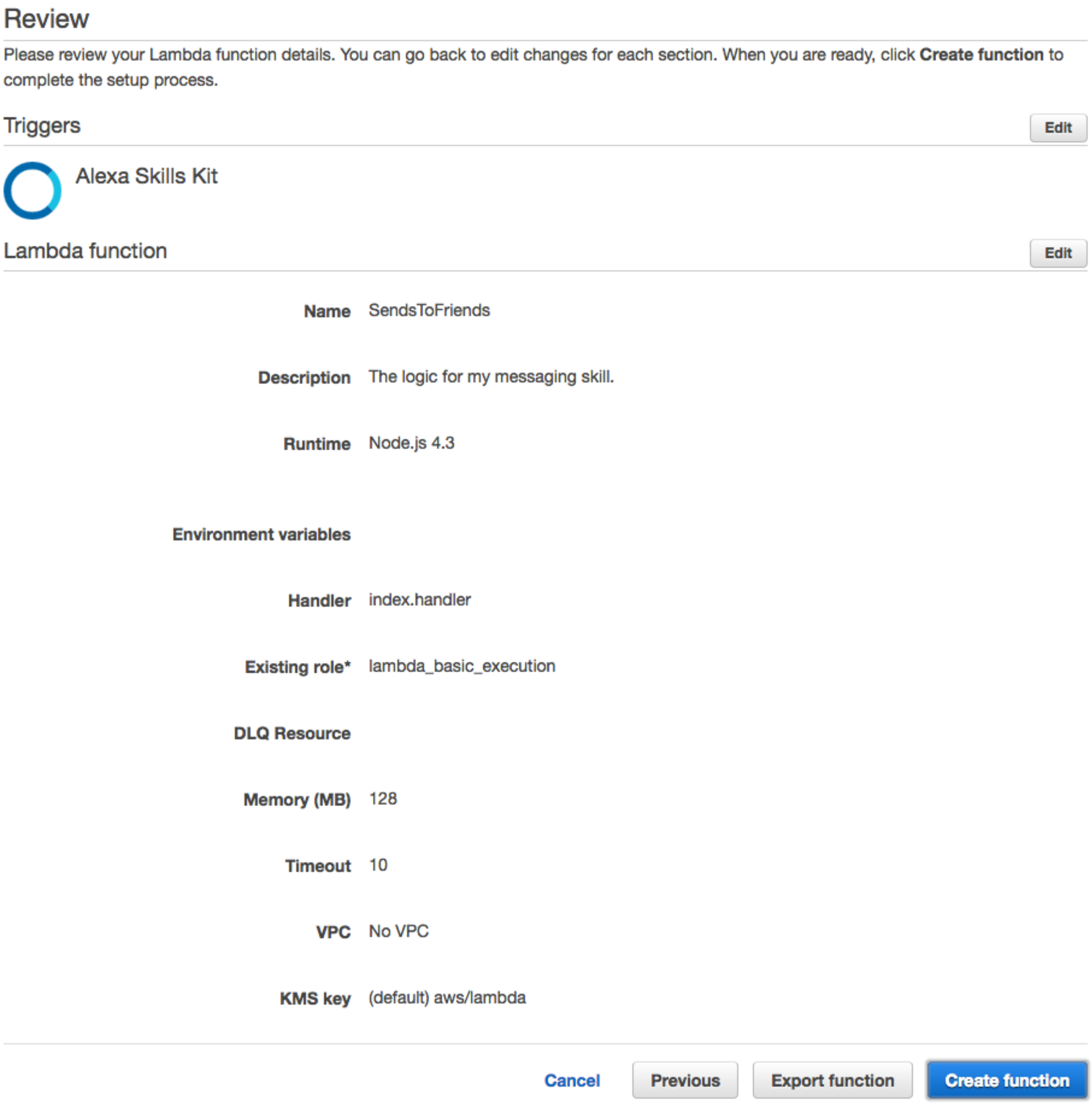
Lambda function handler and role

Handler* index.handler

Role* Choose an existing role

Existing role* lambda_basic_execution

12. Keep the Advanced settings as default. Select **'Next'** and review. You should see something like below. Then select **'Create Function'**:



Review

Please review your Lambda function details. You can go back to edit changes for each section. When you are ready, click **Create function** to complete the setup process.

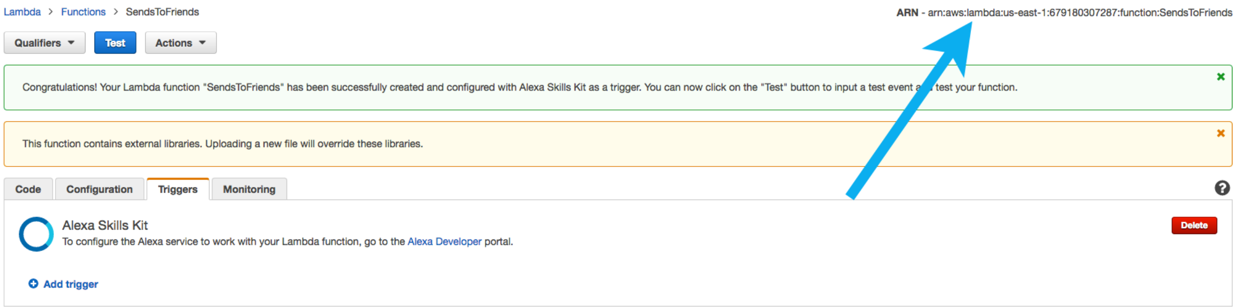
Triggers

Alexa Skills Kit

Lambda function

Name	SendsToFriends
Description	The logic for my messaging skill.
Runtime	Node.js 4.3
Environment variables	
Handler	index.handler
Existing role*	lambda_basic_execution
DLQ Resource	
Memory (MB)	128
Timeout	10
VPC	No VPC
KMS key	(default) aws/lambda

13. Congratulations, you have created your AWS Lambda function. **Copy** the Amazon Resource Name (ARN) for use in the Configuration section of the Amazon Developer Portal.



Lambda > Functions > SendsToFriends

ARN - arn:aws:lambda:us-east-1:679180307287:function:SendsToFriends

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

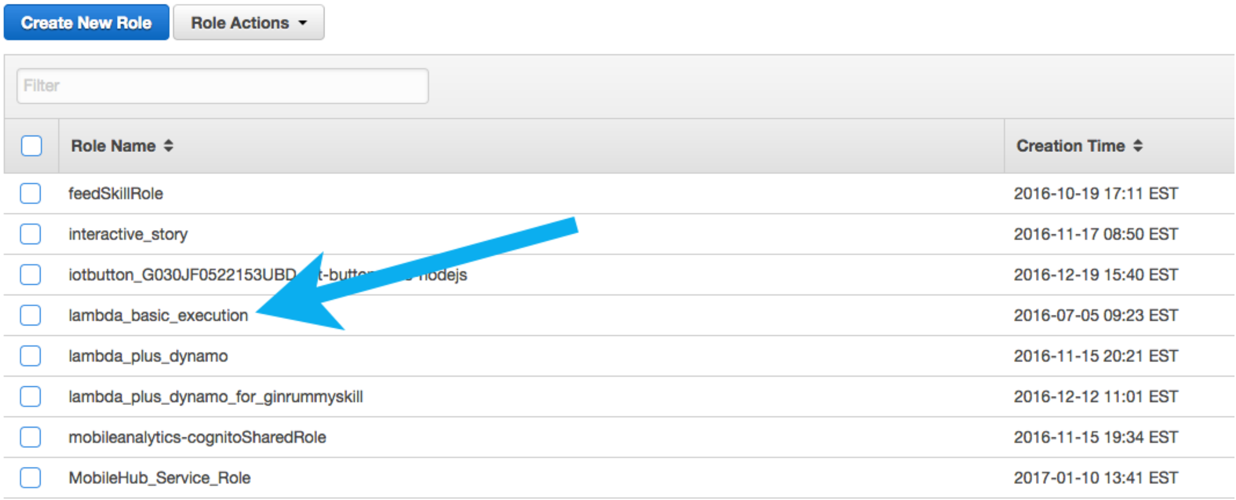
This function contains external libraries. Uploading a new file will override these libraries.

Alexa Skills Kit

To configure the Alexa service to work with your Lambda function, go to the Alexa Developer portal.

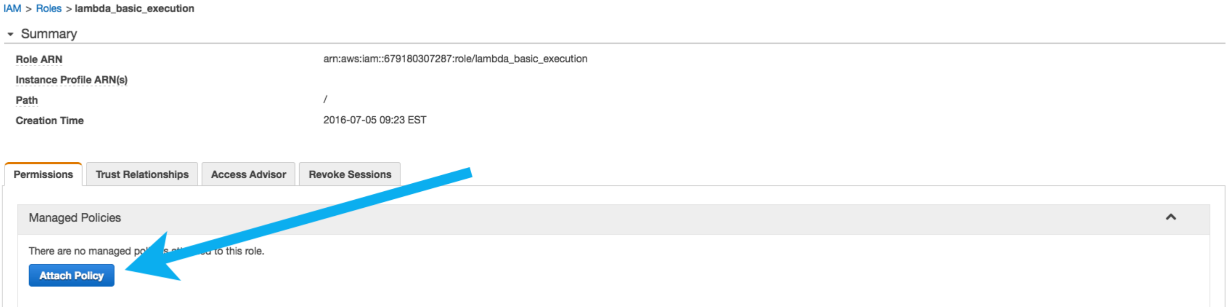
14. Before we move on, we have one additional step we need to take in this tutorial, because we will be using the Simple Notification Service (SNS). We need to modify the IAM role we created earlier so that it has permissions to use the SNS service. To do this, go to the IAM section of the AWS console. You should see your

"lambda_basic_execution" role in a list. Click it to open that role.



□	Role Name ⬍	Creation Time ⬍
□	feedSkillRole	2016-10-19 17:11 EST
□	interactive_story	2016-11-17 08:50 EST
□	iotbutton_G030JF0522153UBD-iot-butto[illegible]nodejs	2016-12-19 15:40 EST
□	lambda_basic_execution	2016-07-05 09:23 EST
□	lambda_plus_dynamo	2016-11-15 20:21 EST
□	lambda_plus_dynamo_for_ginrummyskill	2016-12-12 11:01 EST
□	mobileanalytics-cognitoSharedRole	2016-11-15 19:34 EST
□	MobileHub_Service_Role	2017-01-10 13:41 EST

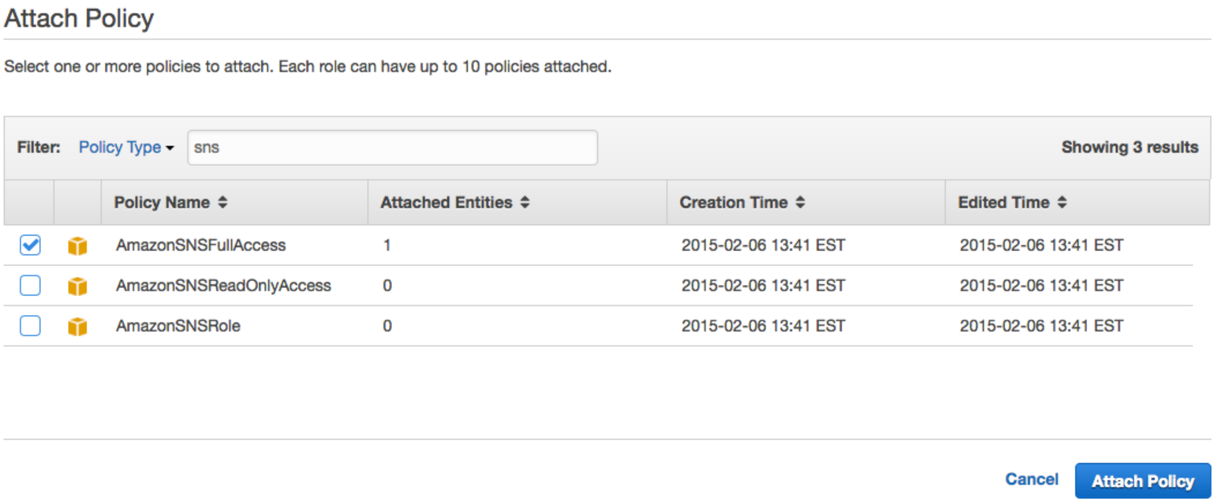
15. On the next screen, we need to add a new managed policy to our IAM role. To do this, click the "Attach Policy" button



Role ARN	arn:aws:iam::679180307287:role/lambda_basic_execution
Instance Profile ARN(s)	
Path	/
Creation Time	2016-07-05 09:23 EST

There are no managed policies attached to this role.

16. On the "Attach Policy" screen, you are looking for the "AmazonSNSFullAccess" policy. (Typing SNS in the filter box makes it easy to find.) Check the box, and click "Attach Policy" at the bottom of the page. This will apply the SNS policy to your IAM role, making it possible for your skill to send SMS messages using this service.



Attach Policy

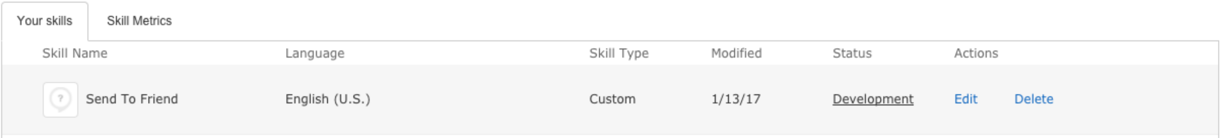
Select one or more policies to attach. Each role can have up to 10 policies attached.

		Policy Name ⬍	Attached Entities ⬍	Creation Time ⬍	Edited Time ⬍
☑	📦	AmazonSNSFullAccess	1	2015-02-06 13:41 EST	2015-02-06 13:41 EST
□	📦	AmazonSNSReadOnlyAccess	0	2015-02-06 13:41 EST	2015-02-06 13:41 EST
□	📦	AmazonSNSRole	0	2015-02-06 13:41 EST	2015-02-06 13:41 EST

You can close this browser tab when you have completed these steps.

Step 3: Add Your Lambda Function to Your Skill

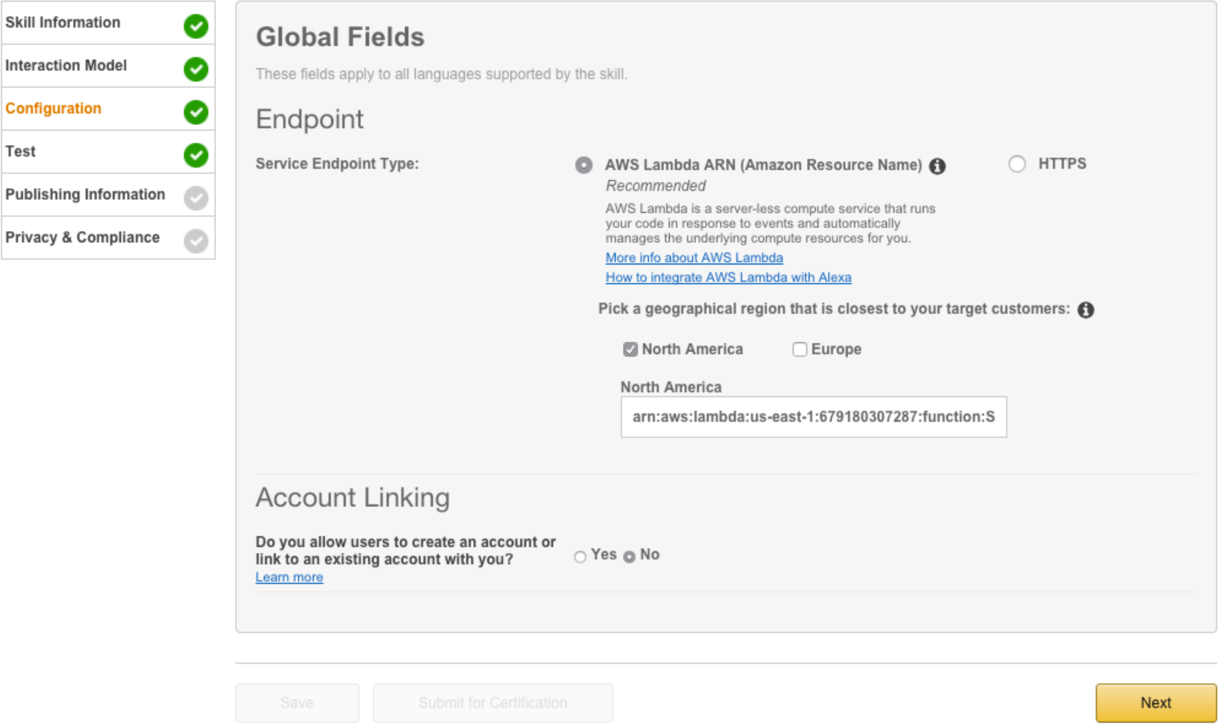
1. Navigate back to developer.amazon.com and select your skill from the list. You can select the skill name or the edit button.



Your skills | Skill Metrics

Skill Name	Language	Skill Type	Modified	Status	Actions	
Send To Friend	English (U.S.)	Custom	1/13/17	Development	Edit	Delete

2. Select the Configuration section, and make sure to choose the AWS Lambda ARN region that corresponds to your AWS Lambda function's region. Add the ARN from the Lambda function you created in the AWS Console earlier. Select the **Lambda ARN (Amazon Resource Name)** radio button. Then, select **"No"** for account linking since we will not be connecting to an external account for this tutorial. Paste the ARN you copied earlier into the Endpoint field. Then select **Next**.



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

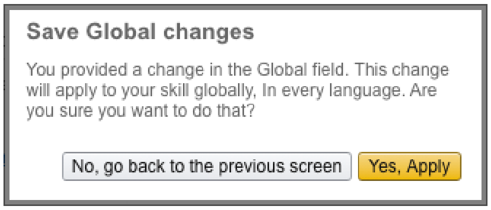
arn:aws:lambda:us-east-1:679180307287:function:S

Account Linking

Do you allow users to create an account or link to an existing account with you? ○ Yes ● No

Learn more

3. You will be asked if you want to "Save Global Changes." This happens because you are changing values that would apply to every version of your skill (in every language.) You can click "Yes, Apply" to complete this step.



Save Global changes

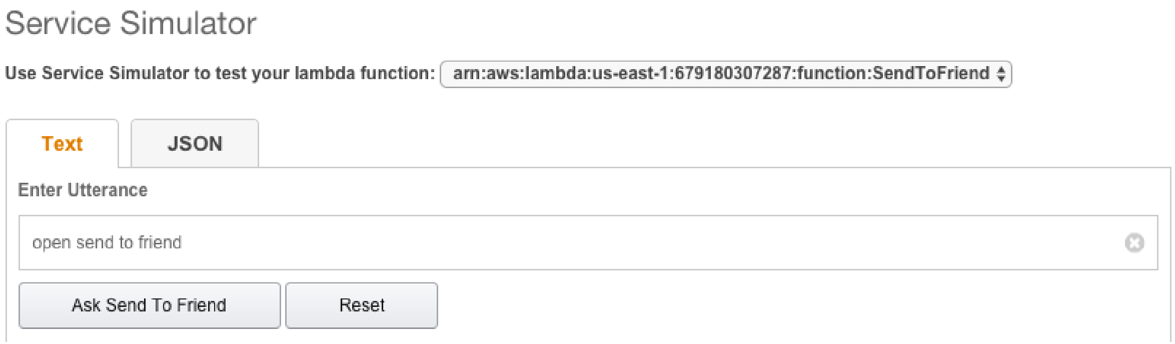
You provided a change in the Global field. This change will apply to your skill globally, In every language. Are you sure you want to do that?

4. You have now completed the initial development of your skill. Now it is time to test.

Step 4: Testing Your Skill

1. In the Test area, we are going to enter a sample utterance in the service simulator section and see how Alexa will respond. In this example, we have called the skill "Send To Friend." This is the "Invocation Name" we set up on the Skill Information line in the “Skill Information” section.

   - In the Service Simulator, type **‘open send to friend’** and click the **“Ask Send to Friend”** button.



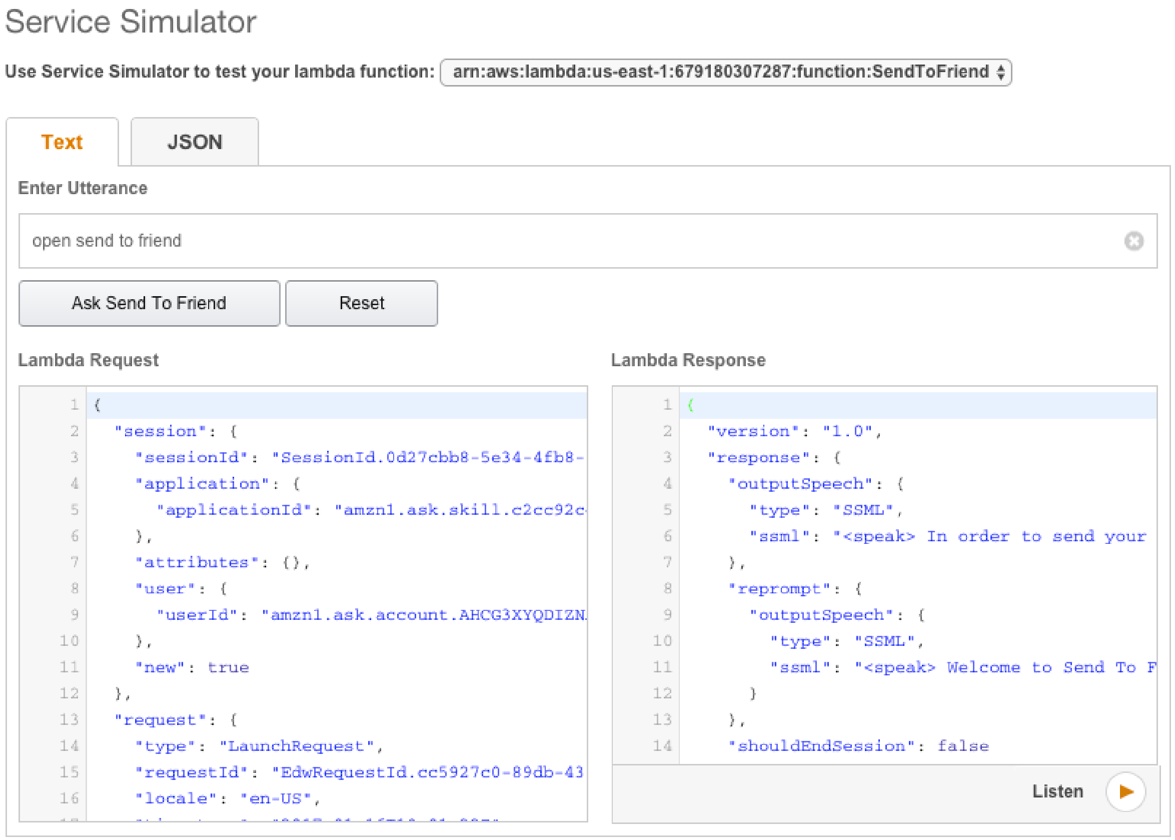
Service Simulator

Use Service Simulator to test your lambda function: arn:aws:lambda:us-east-1:679180307287:function:SendToFriend

Enter Utterance

open send to friend

2. You should see the formatted JSON request from the Alexa service and the response coming back from your Lambda function. Verify that you get a correct Lambda response.



Service Simulator

Use Service Simulator to test your lambda function: arn:aws:lambda:us-east-1:679180307287:function:SendToFriend

Enter Utterance

open send to friend

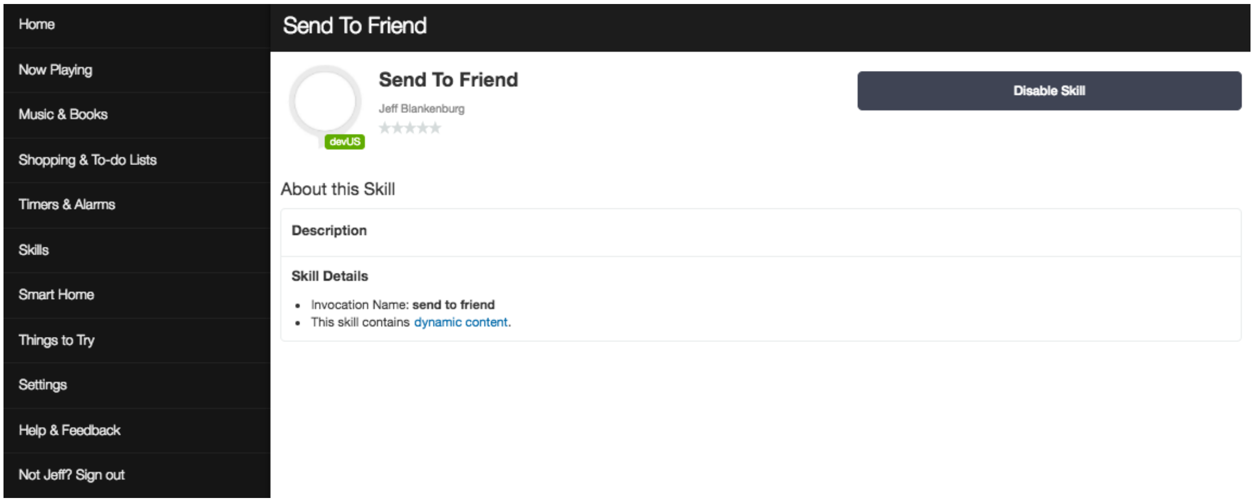
Lambda Request

```
{
  "session": {
    "sessionId": "SessionId.0d27cbb8-5e34-4fb8-
    "application": {
      "applicationId": "amzn1.ask.skill.c2cc92c
    },
    "attributes": {},
    "user": {
      "userId": "amzn1.ask.account.AHCG3XYQDIZN
    },
    "new": true
  },
  "request": {
    "type": "LaunchRequest",
    "requestId": "EdwRequestId.cc5927c0-89db-43
    "locale": "en-US",
```

Lambda Response

```
{
  "version": "1.0",
  "response": {
    "outputSpeech": {
      "type": "SSML",
      "ssml": "<speak> In order to send your
    },
    "reprompt": {
      "outputSpeech": {
        "type": "SSML",
        "ssml": "<speak> Welcome to Send To F
      }
    },
    "shouldEndSession": false
```

3. (Optional) Testing with your device. This is optional as you can do all the testing in the portal. Assuming your Echo device is on-line (and logged in with the same account as your developer account), you should now see your skill enabled in the Alexa app (under "Your Skills," in the top right corner) and ask Alexa to launch your skill. For more information on testing an Alexa skill and registering an Alexa-enabled device, check here.



Send To Friend

Jeff Blankenburg

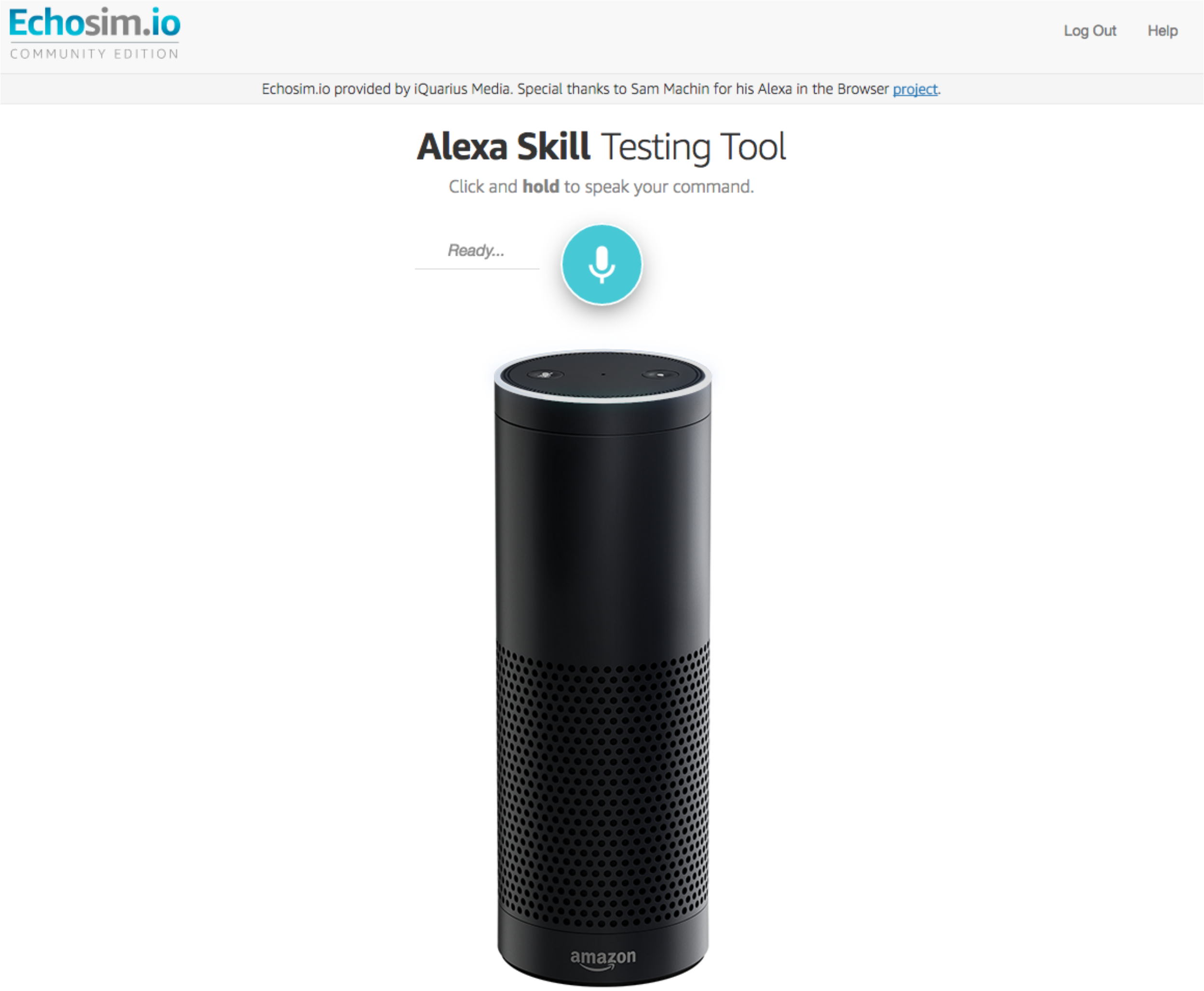
About this Skill

Description

Skill Details

- Invocation Name: **send to friend**
- This skill contains dynamic content.

Another option for testing your device with your voice is Echosim.io. This is a virtual Alexa device in your browser, created and hosted by iQuarius Media, that you can speak to and get responses from, just like having a physical device in front of you.



Echosim.io provided by iQuarius Media. Special thanks to Sam Machin for his Alexa in the Browser project.

Alexa Skill Testing Tool

Click and **hold** to speak your command.

Skills / Lambda Troubleshooting (getting an invalid response)?

- Do you have the right ARN copied from your Lambda function into your Developer Portal / skill?
- Are you calling the right invocation name?
- Are you saying launch, start or open (followed by your invocation name)?
- Are you sure you have no other skills in your accounts with the same invocation name?

Step 5: Make it Yours

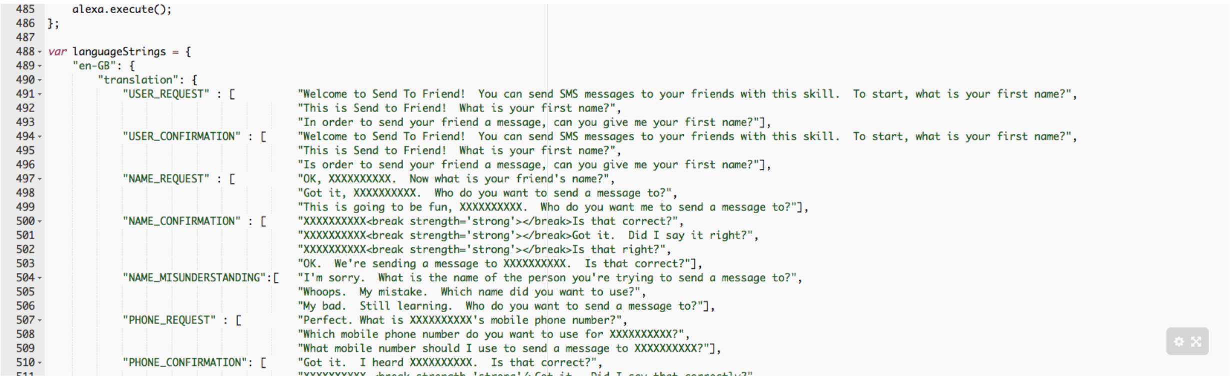
1. In the Skill Information section in the Developer Console, edit the Skill Information Tab to reflect your new messaging skill:

   i. Provide a skill name that represents the new skill you are creating.
   ii. Come up with a cool Invocation Name that users will use to invoke your skill. Make sure to read the rules for creating invocation names, as this is a common failure point during certification.

   iii. Create a fun icon. Be sure you have the rights to whatever icons you are uploading – you will need to provide both 108x108px and 512x512px images. Need help finding an image? Try The Noun Project or Pixabay as a possible source for royalty-free images. Use an image editor (such as Paint on Windows or Preview on Mac) to change the size of the image.

   Everything else can stay as-is for now in the Developer Portal.

2. Head back to your Lambda function now. This is where you can customize the kinds of things that Alexa says and does. Feel free to peruse the code in the "Code" tab, but the important parts for customization start on line 488. There is where you will find a large collection of language strings broken up into three languages. They are in order alphabetically, en-GB is for UK English, en-US is for United States English, and de-DE is for German.



```
485      alexa.execute();
486  };
487
488  var languageStrings = {
489      "en-GB": {
490          "translation": {
491              "USER_REQUEST" : [          "Welcome to Send To Friend!  You can send SMS messages to your friends with this skill.  To start, what is your first name?",
492                                          "This is Send to Friend!  What is your first name?",
493                                          "In order to send your friend a message, can you give me your first name?"],
494              "USER_CONFIRMATION" : [     "Welcome to Send To Friend!  You can send SMS messages to your friends with this skill.  To start, what is your first name?",
495                                          "This is Send to Friend!  What is your first name?",
496                                          "Is order to send your friend a message, can you give me your first name?"],
497              "NAME_REQUEST" : [          "OK, XXXXXXXXXX.  Now what is your friend's name?",
498                                          "Got it, XXXXXXXXXX.  Who do you want to send a message to?",
499                                          "This is going to be fun, XXXXXXXXXX.  Who do you want me to send a message to?"],
500              "NAME_CONFIRMATION" : [     "XXXXXXXXXX<break strength='strong'></break>Is that correct?",
501                                          "XXXXXXXXXX<break strength='strong'></break>Got it.  Did I say it right?",
502                                          "XXXXXXXXXX<break strength='strong'></break>Is that right?",
503                                          "OK.  We're sending a message to XXXXXXXXXX.  Is that correct?"],
504              "NAME_MISUNDERSTANDING":[   "I'm sorry.  What is the name of the person you're trying to send a message to?",
505                                          "Whoops.  My mistake.  Which name did you want to use?",
506                                          "My bad.  Still learning.  Who do you want to send a message to?"],
507              "PHONE_REQUEST" : [         "Perfect. What is XXXXXXXXXX's mobile phone number?",
508                                          "Which mobile phone number do you want to use for XXXXXXXXXX?",
509                                          "What mobile number should I use to send a message to XXXXXXXXXX?"],
510              "PHONE_CONFIRMATION": [     "Got it.  I heard XXXXXXXXXX.  Is that correct?",
511                                          "XXXXXXXXXX <break strength 'strong'/>Got it.  Did I say that correctly?"
```

3. You will notice that there are three different strings for each type of response. This is done so that Alexa can vary her responses to the user. Each time that Alexa needs to respond to the user, she will select one of the appropriate values at random from the list. You should definitely expand her response list to make her responses even more conversational.

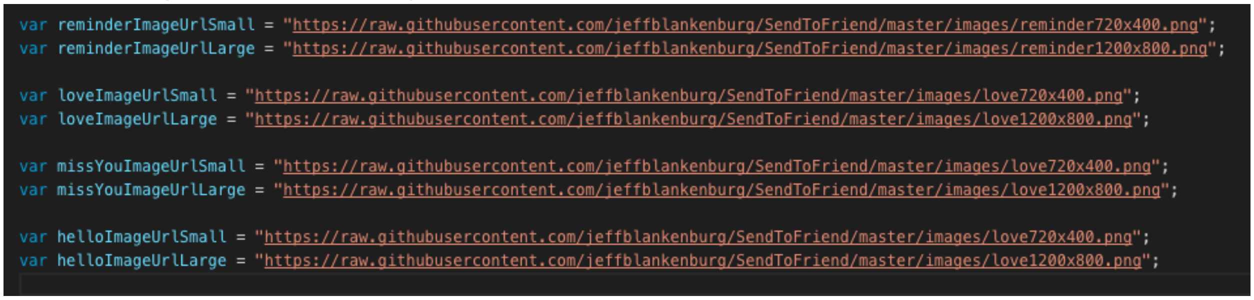
4. To clarify each of the response types, here's a quick guide:

   - **USER_REQUEST**: This welcomes the user to the skill, and prompts them for their first name.
   - **USER_CONFIRMATION**: This message repeats the user's name back to them, to make sure that she heard it correctly.
   - **NAME_REQUEST**: This repeats the user's name, and asks for the name of the person they are sending a message to.
   - **NAME_CONFIRMATION**: This repeats the intended recipient's name back to the user, asking for confirmation that Alexa heard it correctly.
   - **NAME_MISUNDERSTANDING**: If Alexa got the name wrong, she apologizes, and then asks for the name again.
   - **PHONE_REQUEST**: This asks the user for the mobile phone number that the message should be sent to.
   - **PHONE_CONFIRMATION**: Alexa repeats the phone number back to the user, and confirms that it is correct.
   - **PHONE_MISUNDERSTANDING**: When a user provides a value that is not 10 digits (the format of a US-based mobile phone number), she repeats the number, and asks the user to say the number again.
   - **PHONE_RETRY**: If the user indicates that the phone number was received incorrectly, Alexa apologizes and asks for it again.
   - **MESSAGE_SENT**: This is confirmation to the user that a message has been sent to their intended user. This is paired with an option from one of the next four categories that the user selected.

- **REMINDER_MESSAGE**: These are messages meant to remind the recipient about something that the user indicated. This uses the data from our custom slot we called "noun" earlier.
- **LOVE_MESSAGE**: These are messages to communicate that the user was thinking about someone they love.
- **MISSYOU_MESSAGE**: These are messages to communicated that the user misses the recipient.
- **HELLO_MESSAGE**: These are messages to send a friendly "Hello" message to their recipient.
- **HELP_MESSAGE**: This is message the user will hear when they ask for help. It tells them what they can do, and prompts them to say "Start Over" or "Quit."
- **UNHANDLED_MESSAGE**: If the user somehow tries something that isn't handled, this is the message they will receive. Similar to help, it asks them to "Start Over" or "Quit."
- **STOP_MESSAGE**: When a user indicates that they want our skill to stop, this should be a very brief message saying goodbye.

5. You will definitely notice that some of the responses have "XXXXXXXXXX" or "YYYYYYYYYY" in them. We built this as a way to easily collect all of your strings in one place, but also insert some of the data the user provides into those strings. For example, one of the USER_CONFIRMATION strings is "Perfect. I heard your name as XXXXXXXXXX. Is that right?" The code earlier in our Lambda function will look for a string of 10 Xs and replace them with the name that the user provides to us. The same goes for phone numbers, recipient names, etc.

6. The final change you need to make relates to the cards that get created when a user sends a message. Each of these cards has two images associated with it. You need one image that is 1200x800 pixels, and one that is 720x400 for each message type. These images also need to be hosted somewhere on the internet where they can be publicly available. You can change each of these images in the variables immediately below all of the response strings you've been working on.



```
var reminderImageUrlSmall = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/reminder720x400.png";
var reminderImageUrlLarge = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/reminder1200x800.png";

var loveImageUrlSmall = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love720x400.png";
var loveImageUrlLarge = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love1200x800.png";

var missYouImageUrlSmall = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love720x400.png";
var missYouImageUrlLarge = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love1200x800.png";

var helloImageUrlSmall = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love720x400.png";
var helloImageUrlLarge = "https://raw.githubusercontent.com/jeffblankenburg/SendToFriend/master/images/love1200x800.png";
```

7. Once you have successfully updated all of your strings, scroll back to the top of the page and select “Save”.

8. Repeat the tests you performed earlier to ensure your changes are functioning properly. See Step #4 for a review of how to perform functional tests.

Step 6: Publish Your Skill

Now we need to go back to our Developer Portal to test and edit our skill and we will be ready for certification.

1. In your skill's Test section, enter your Utterances into the Simulator to make sure everything is working with your new messaging skill.

2. Optionally, you can test with your Alexa-enabled device to make sure everything is working correctly. You may find a few words that need to be changed for a better user experience.

   Some things to think about:

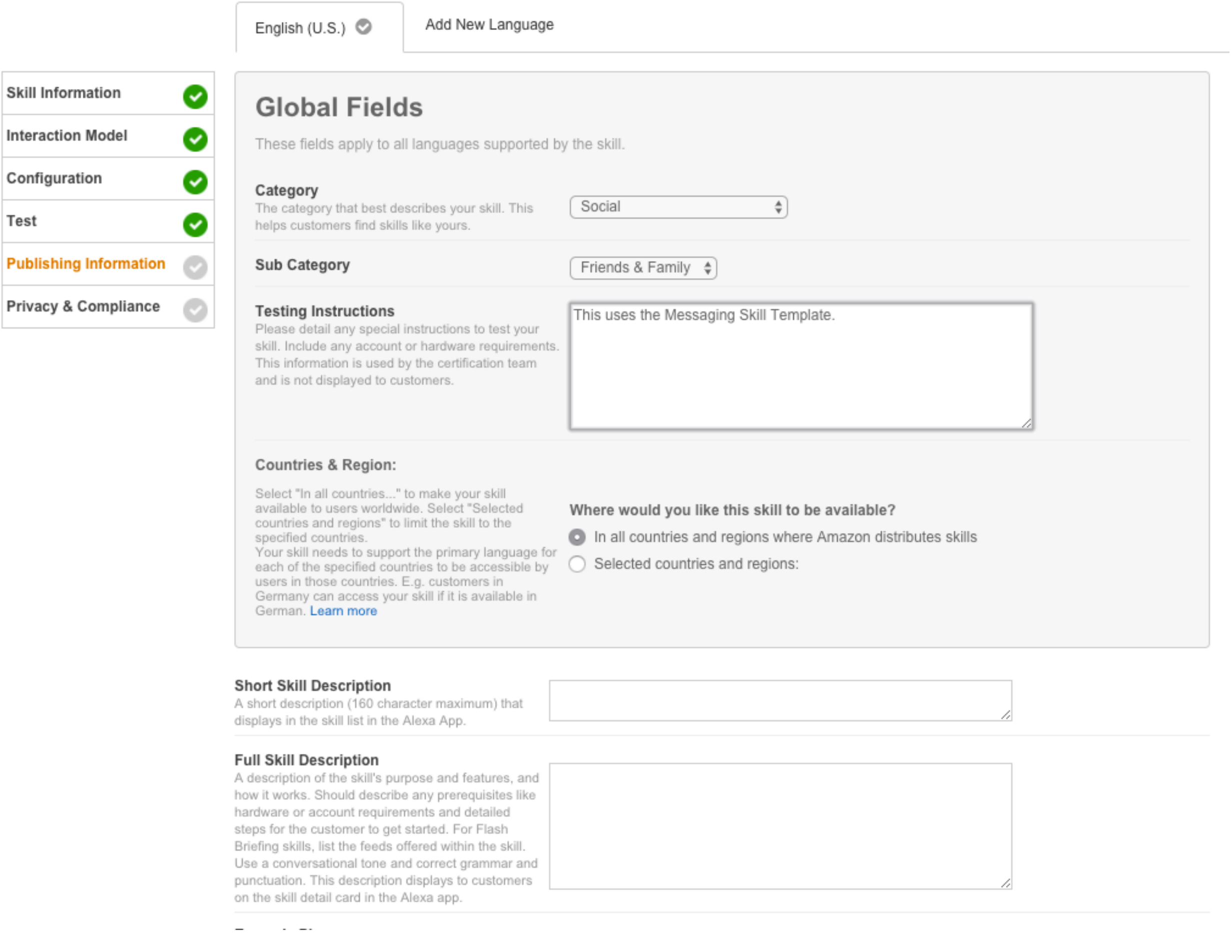
   - Does every word in Alexa's responses sound correct?
   - Do you need to change any words to make them sound correct?

   Since we get to choose our responses, make sure that Alexa pronouces the words you chose properly. You can use the Voice Simulator in the Test section to simulate Alexa’s responses. In the Voice Simulator, type in each location name that you are using to test how Alexa will say it. Use additional punctuation or possibly SSML if you need to better control how Alexa responds. You can find out more about SSML here.

Read more about functional testing for Alexa skills.

Read more about building effective voice user interfaces (VUI).

3. Select the Publishing Information area of your skill next:



English (U.S.) ✓ | Add New Language

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

Global Fields

These fields apply to all languages supported by the skill.

Category
The category that best describes your skill. This helps customers find skills like yours.
Social

Sub Category
Friends & Family

Testing Instructions
Please detail any special instructions to test your skill. Include any account or hardware requirements. This information is used by the certification team and is not displayed to customers.
This uses the Messaging Skill Template.

Countries & Region:

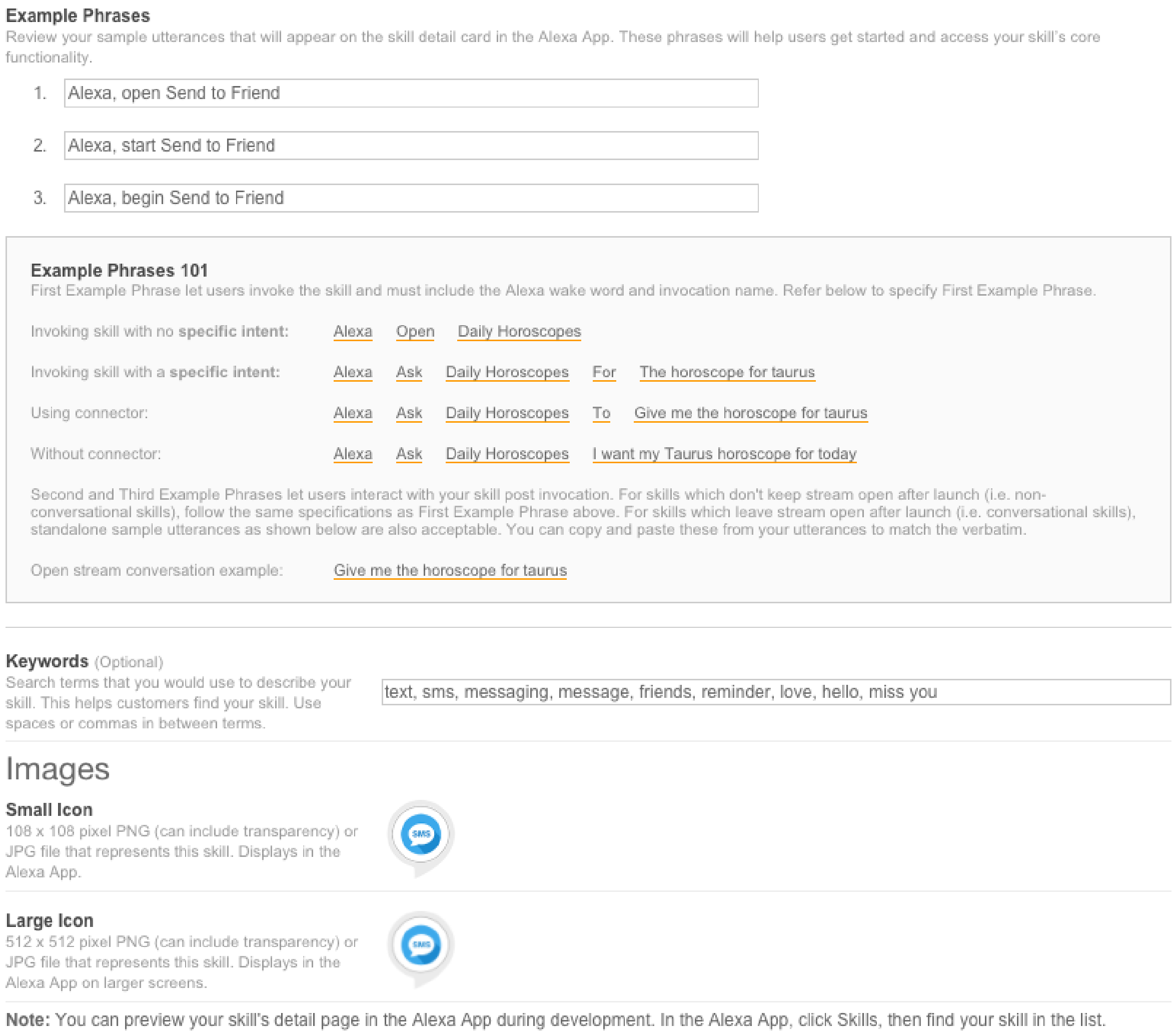
Select "In all countries..." to make your skill available to users worldwide. Select "Selected countries and regions" to limit the skill to the specified countries.
Your skill needs to support the primary language for each of the specified countries to be accessible by users in those countries. E.g. customers in Germany can access your skill if it is available in German. Learn more

Where would you like this skill to be available?
● In all countries and regions where Amazon distributes skills
○ Selected countries and regions:

Short Skill Description
A short description (160 character maximum) that displays in the skill list in the Alexa App.

Full Skill Description
A description of the skill's purpose and features, and how it works. Should describe any prerequisites like hardware or account requirements and detailed steps for the customer to get started. For Flash Briefing skills, list the feeds offered within the skill. Use a conversational tone and correct grammar and punctuation. This description displays to customers on the skill detail card in the Alexa app.

- Spend some time coming up with an enticing, succinct description. This is an excellent way to attract new users. These descriptions show up on the list of skills available in the Alexa app, or at amazon.com/skills.
- In your example phrases, be sure that the examples you use exactly match the utterances that you created in the Interaction Model section. The first example should be "Alexa, open {your invocation name}" and no utterance. Remember, there are built-in intents such as help and cancel. You can learn more about built-in intents here. You can also review the list of supported phrases to begin a conversation.
- Be sure you have the rights to whatever icons you are uploading – you will need to provide both 108x108px and 512x512px images. If there is any question, the Amazon certification team will fail your Alexa skill submission. In the event your skill fails certification, you will receive an email from Amazon's testing team with information about your certification results.
- IMPORTANT: Add the text “This is based on the Messaging Skill Template” to the Testing Instructions section. This alerts the Certification team of your submission using this standardized template, smoothing the road to a faster certification process.



Example Phrases

Review your sample utterances that will appear on the skill detail card in the Alexa App. These phrases will help users get started and access your skill's core functionality.

1. Alexa, open Send to Friend
2. Alexa, start Send to Friend
3. Alexa, begin Send to Friend

Example Phrases 101

First Example Phrase let users invoke the skill and must include the Alexa wake word and invocation name. Refer below to specify First Example Phrase.

Invoking skill with no **specific intent:** Alexa Open Daily Horoscopes

Invoking skill with a **specific intent:** Alexa Ask Daily Horoscopes For The horoscope for taurus

Using connector: Alexa Ask Daily Horoscopes To Give me the horoscope for taurus

Without connector: Alexa Ask Daily Horoscopes I want my Taurus horoscope for today

Second and Third Example Phrases let users interact with your skill post invocation. For skills which don't keep stream open after launch (i.e. non-conversational skills), follow the same specifications as First Example Phrase above. For skills which leave stream open after launch (i.e. conversational skills), standalone sample utterances as shown below are also acceptable. You can copy and paste these from your utterances to match the verbatim.

Open stream conversation example: Give me the horoscope for taurus

Keywords (Optional)

Search terms that you would use to describe your skill. This helps customers find your skill. Use spaces or commas in between terms.

text, sms, messaging, message, friends, reminder, love, hello, miss you

Images

Small Icon

108 x 108 pixel PNG (can include transparency) or JPG file that represents this skill. Displays in the Alexa App.

Large Icon

512 x 512 pixel PNG (can include transparency) or JPG file that represents this skill. Displays in the Alexa App on larger screens.

Note: You can preview your skill's detail page in the Alexa App during development. In the Alexa App, click Skills, then find your skill in the list.

Once you have uploaded your icons, you should see a success message at the bottom of the screen.

4. Privacy and Compliance.

   i. On the Privacy and Compliance section, make sure to consider each answer.

      a. Since our skill does not require purchases or spending of money, choose "No" for the first question.

      b. We are collecting personal information in this skill when we ask for names and mobile phone numbers. Answer yes for the second question.

      c. If your messaging skill is not directed to children under the age of 13 or is not intended to target children under the age of 13, choose ‘No’ for the third question as well.

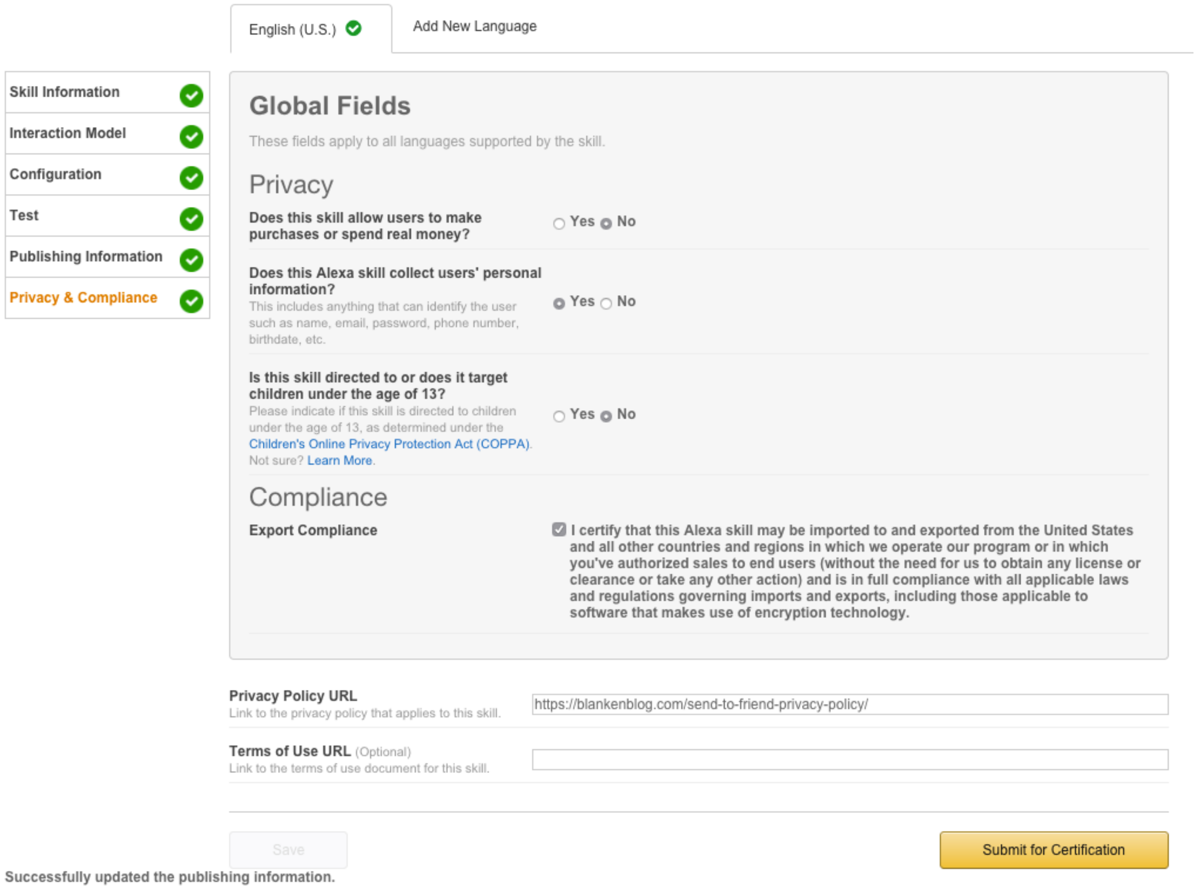
      d. Choose to certify that your skill can be imported to and exported from the countries and regions that you operate the skill.

      e. You must provide a simple privacy policy in order to publish this skill. You can set up a free Wordpress site to host this, a GitHub repository, or post a page on a website of your choosing. You can see my example here.

      f. A Terms of Use is optional, but always recommended as well.

   ii. Select **“Save”**.

   iii. Select “Submit for Certification”



English (U.S.) ✓ | Add New Language

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

Global Fields

These fields apply to all languages supported by the skill.

Privacy

Does this skill allow users to make purchases or spend real money? ○ Yes ● No

Does this Alexa skill collect users' personal information? ● Yes ○ No
This includes anything that can identify the user such as name, email, password, phone number, birthdate, etc.

Is this skill directed to or does it target children under the age of 13? ○ Yes ● No
Please indicate if this skill is directed to children under the age of 13, as determined under the Children's Online Privacy Protection Act (COPPA).
Not sure? Learn More.

Compliance

Export Compliance ☑ I certify that this Alexa skill may be imported to and exported from the United States and all other countries and regions in which we operate our program or in which you've authorized sales to end users (without the need for us to obtain any license or clearance or take any other action) and is in full compliance with all applicable laws and regulations governing imports and exports, including those applicable to software that makes use of encryption technology.

Privacy Policy URL
Link to the privacy policy that applies to this skill.
https://blankenblog.com/send-to-friend-privacy-policy/

Terms of Use URL (Optional)
Link to the terms of use document for this skill.

Successfully updated the publishing information.

iv. Finally, confirm your submission. Select “Yes” to submit your skill.

Congratulations! You have successfully submitted your skill for publication. You will receive progress e-mails and possibly other suggestions from the Alexa certification team on how you can make your skill even better. You will typically receive news back from the certification team within 4-5 business days. You can update your skills at any time, except while they are being reviewed for certification.

Did you like this tutorial? You can find more on our Alexa Skills Kit training page

Check out These Other Developer Resources

- Alexa Skills Kit (ASK)
- Alexa Skill Submission Checklist
- Alexa Developer Forums
- Knowledge Base
- Intro to Alexa Skills Kit - On Demand Webinar
- Voice Design 101 - On Demand Webinar
- Developer Office Hours