

## Alexa & API.ai Integration

[Eugene Fisher](#)

Questions? Feedback? Let me know! [bfisher@ebfour.com](mailto:bfisher@ebfour.com)

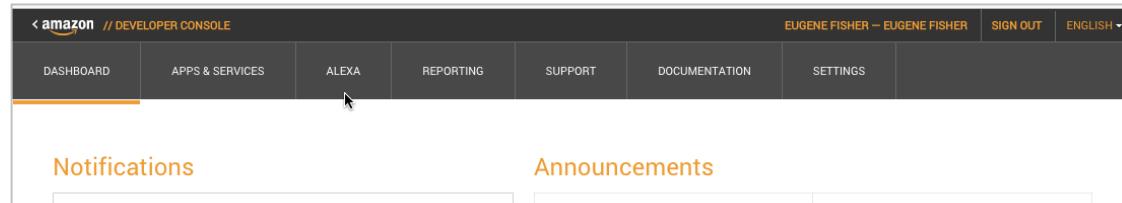
### CREATE AN AGENT ON API.AI

1. If you haven't already, create an Agent with API.ai following [their instructions here](#)

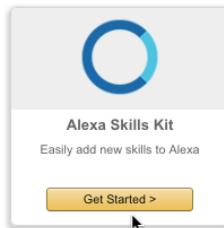
### CREATE AN ALEXA SKILL

1. Sign in to <https://developer.amazon.com>

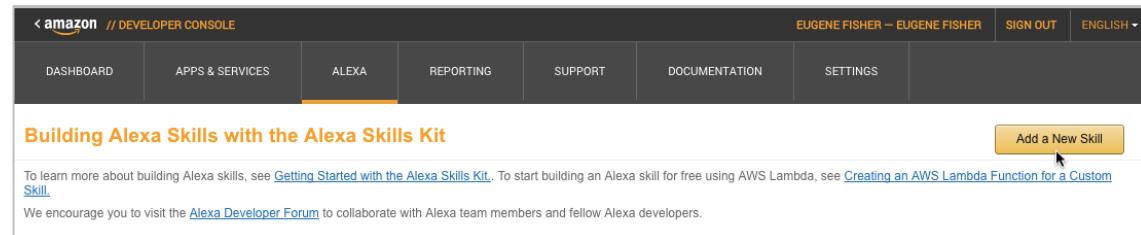
2. Click Alexa from main navigation



3. Click Alexa Skills Kit



4. Click Add a New Skill



## 5. Enter Skill Information

- Select Skill Type: Custom Interaction Model
- Select Language
- Enter Skill Name (must be between 2-50 characters, numbers and special characters allowed)
- Enter Invocation Name ([guidelines here](#))
- Select Audio Player use ([information here](#))
- Click Next

English (U.S.)  Add New Language

Skill Information	<input checked="" type="checkbox"/>
Interaction Model	<input checked="" type="checkbox"/>
Configuration	<input checked="" type="checkbox"/>
Test	<input checked="" type="checkbox"/>
Publishing Information	<input checked="" type="checkbox"/>
Privacy & Compliance	<input checked="" type="checkbox"/>

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

**Language**  
Language of your skill English (U.S.)

**Application Id**  
The ID for this skill amzn1.ask.skill.a4da702a-930c-4df4-b0ad-761490eb6f33

**Name**  
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters. Test Skill 1

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

**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](#)

Save Submit for Certification **Next**

## 6. Enter Interaction Model

- Enter the following JSON Intent into Intent Schema:

```
{"intents": [
  {
    "intent": "ApIntent",
    "slots": [
      {
        "name": "Text",
        "type": "LITERAL"
      }
    ],
    {
      "intent": "AMAZON.StopIntent"
    }
  }
]}
```

English (U.S.)  Add New Language

Skill Information	<input checked="" type="checkbox"/>
Interaction Model	<input checked="" type="checkbox"/>
Configuration	<input checked="" type="checkbox"/>
Test	<input checked="" type="checkbox"/>
Publishing Information	<input checked="" type="checkbox"/>
Privacy & Compliance	<input checked="" type="checkbox"/>

**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 {"intents": [
2   {
3     "intent": "ApIntent",
4     "slots": [
5       {
6         "name": "Text",
7         "type": "LITERAL"
8       }
9     ],
10    {
11      "intent": "AMAZON.StopIntent"
12    }
13  }
14]}
```

Save Submit for Certification **Next**

7. Enter Custom Slot Type
- Enter Type: list\_bot
  - Enter Values: bot

**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)

Add Slot Type

**Adding slot type**

**Enter Type**  
list\_bot

**Enter Values**  
Values must be line-separated  
1 bot

Delete Cancel Save

8. Enter Sample Utterances:
- ApIntent {bot|Text}
  - ApIntent {hello bot|Text}
  - AMAZON.StopIntent stop
  - AMAZON.StopIntent alexa stop

9. Click Next

**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.

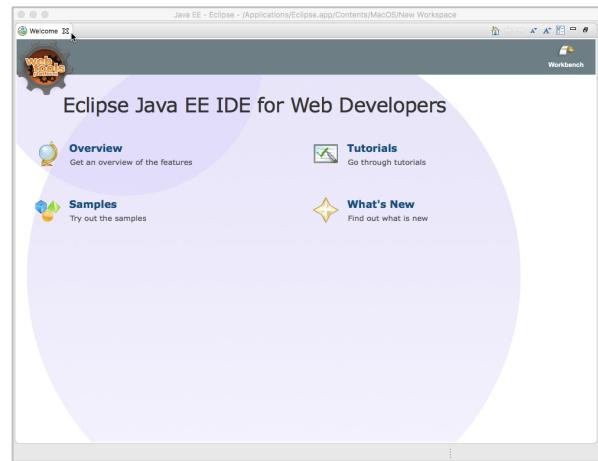
1 ApIntent {bot|Text}  
2 ApIntent {hello bot|Text}  
3 AMAZON.StopIntent stop  
4 AMAZON.StopIntent alexa stop

Save Submit for Certification Next

\*Leave window open and continue with instructions to Update JAR File and Create a Lambda Function on the following pages; you'll return to this window to complete the skill on page 10\*

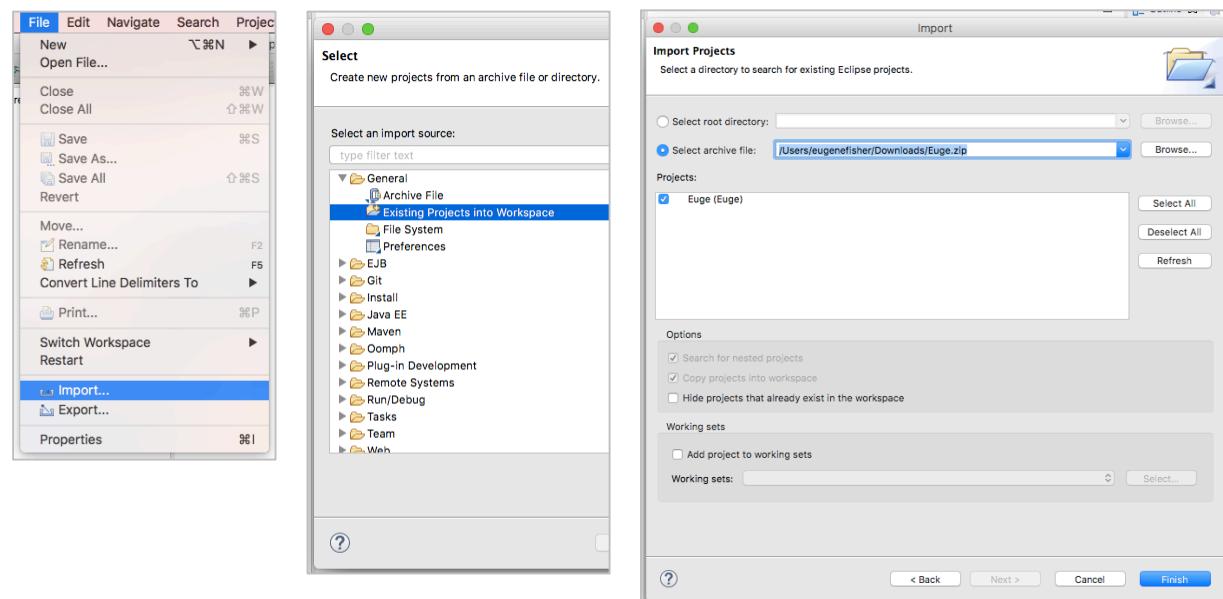
## UPDATE JAR FILE

1. If you don't have it already, download [Java Development Kit 8 \(JDK\)](#) and download an IDE ([I used Eclipse Mars 2](#)).
2. Start IDE, if Eclipse close the Welcome window.



### 3. Import Workspace

- a. Click File > Import
- b. Click General > Existing projects into Workspace
- c. Browse and choose Euge.zip file
- d. Select Euge in Projects window
- e. Click Finish

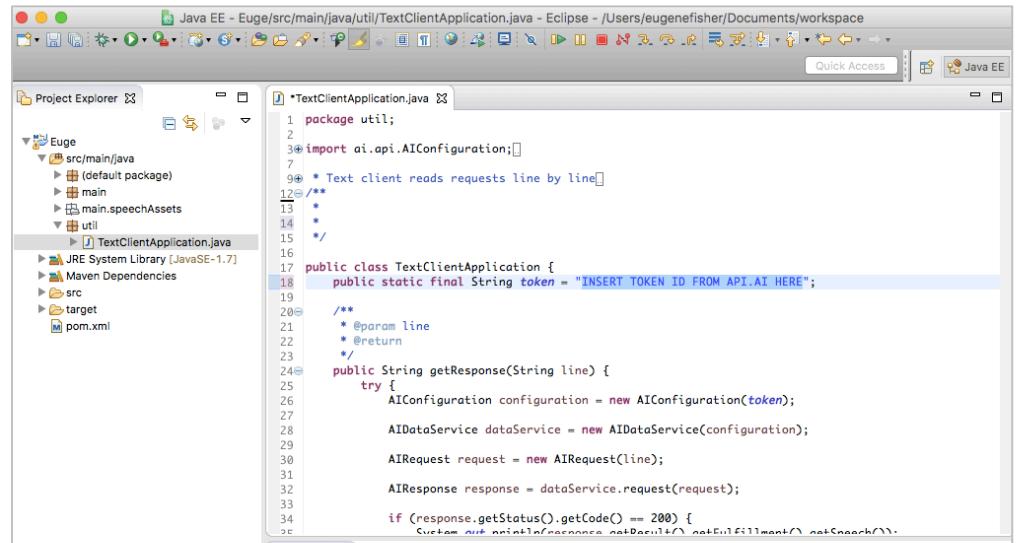


#### 4. Update API.ai Token ID

- In your API.ai account, navigate to Settings by clicking the gear icon next to your Agent Name
- Copy the Client Access Token

The screenshot shows the 'EugeBot' agent settings in the API.ai interface. The 'General' tab is selected. A 'Client access token' field contains the value '36ecd03aefea4a768d9577bded6e9ae7'. Below it is a 'Developer access token' field containing '963c29da5b794a8992c5cd626108b763'.

- In Eclipse, navigate to Euge > src/main/java > util package > TextClientApplication.java
- Paste Client Access Token over INSERT TOKEN ID FROM API.AI HERE (keep double quote marks)



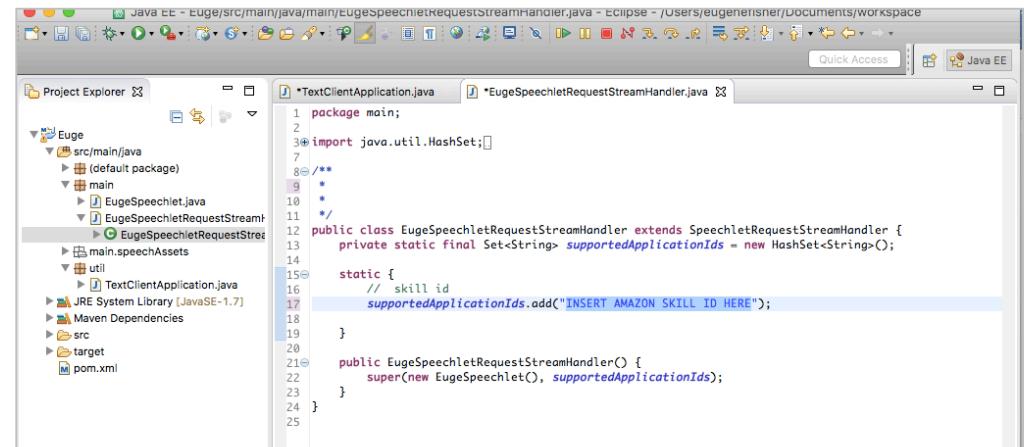
#### 5. Update Alexa Skill ID

- In your Amazon Developer Account navigate to Skill Information tab
- Copy Application ID

The screenshot shows the 'Skill Information' section of the Amazon Developer Skills & Games console. The 'Skill Type' dropdown is set to 'Custom'. The 'Language' is listed as 'English (U.S.)'. The 'Application Id' is 'amzn1.ask.skill.98cb839c-bd93-4e6d-bb81-61e2841c64b4'. The 'Name' field contains 'testearfd'.

## 6. Continued

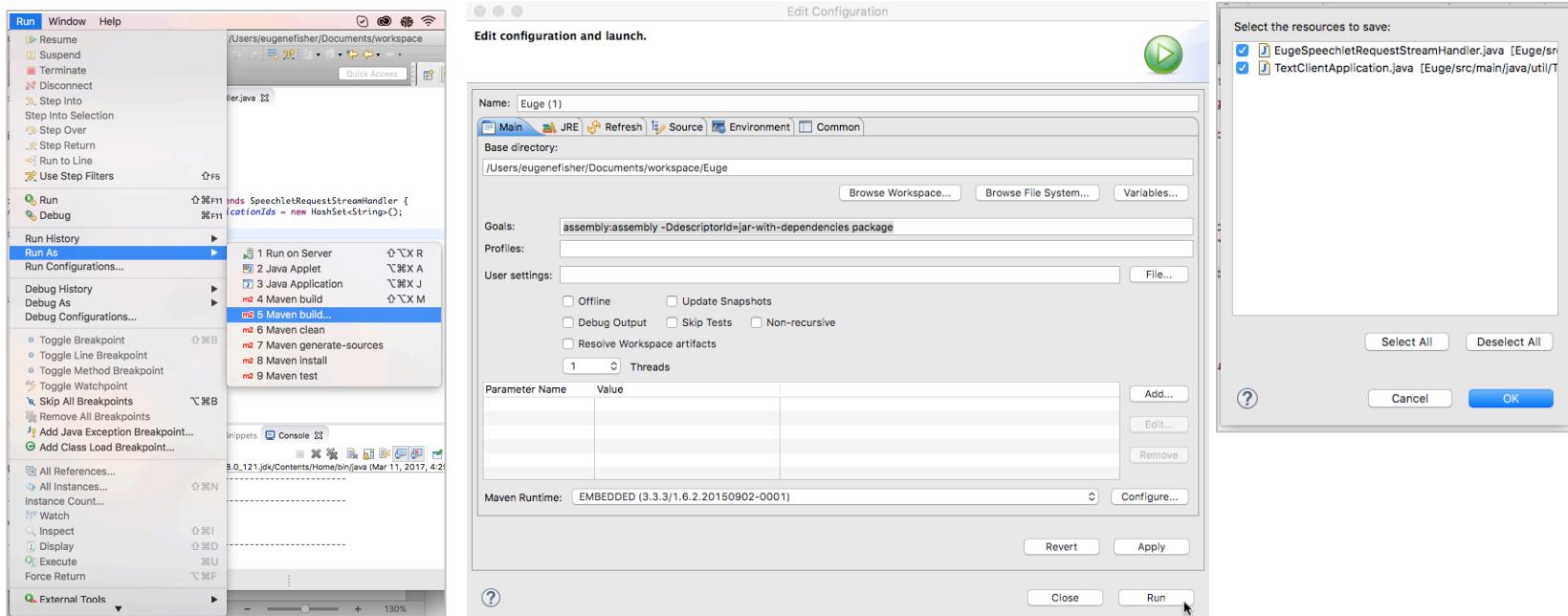
- c. In Eclipse navigate to Euge > Main > EugeSpeechletRequestStreamHandler.java
- d. Paste Alexa Skill ID over INSERT AMAZON SKILL ID HERE (keep double quote marks)



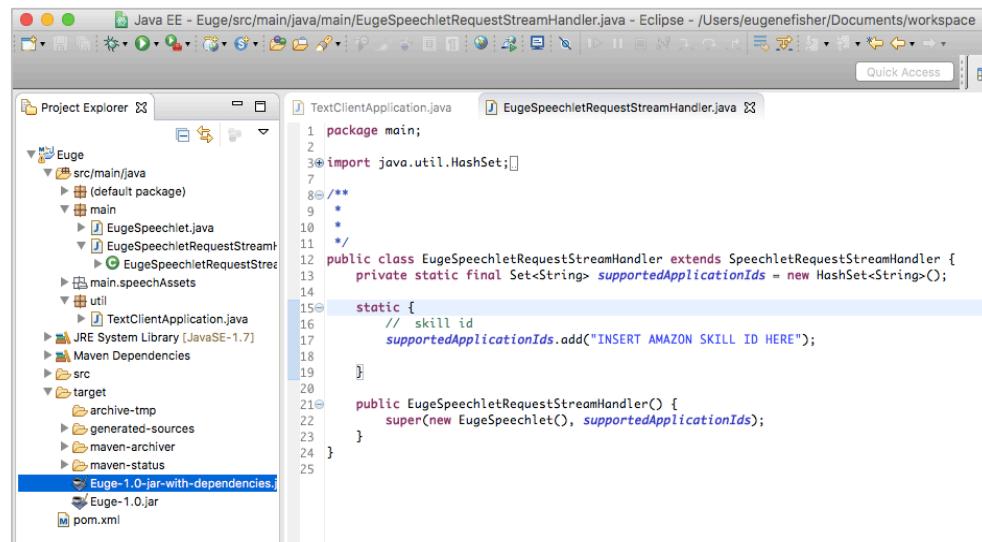
```
1 package main;
2
3 import java.util.HashSet;
4
5 /**
6 * 
7 */
8
9
10 /**
11 */
12
13 public class EugeSpeechletRequestStreamHandler extends SpeechletRequestStreamHandler {
14     private static final Set<String> supportedApplicationIds = new HashSet<String>();
15
16     static {
17         // skill id
18         supportedApplicationIds.add("INSERT AMAZON SKILL ID HERE");
19     }
20
21     public EugeSpeechletRequestStreamHandler() {
22         super(new EugeSpeechlet(), supportedApplicationIds);
23     }
24 }
```

## 6. Build Application

- a. Click Run > Run As > Maven build...
- b. In new window, enter “assembly:assembly -DdescriptorId=jar-with-dependencies package” in the Goals field (without double quote marks) and click Run
- c. In new window, select both resources and click OK



7. Remove JAR file
  - a. Navigate to Euge > target > maven-status > Euge-1.0-jar-with-dependencies.jar
  - b. Copy JAR file and place in desired system directory folder
  
8. Close Eclipse



```

Java EE - Euge/src/main/java/main/EugeSpeechletRequestStreamHandler.java - Eclipse - /Users/eugenefisher/Documents/workspace

Project Explorer
Euge
  src/main/java
    main
      EugeSpeechlet.java
      EugeSpeechletRequestStreamHandler.java
      main.speechAssets
      util
        TextClientApplication.java
      JRE System Library [JavaSE-1.7]
      Maven Dependencies
      src
      target
        archive-tmp
        generated-sources
        maven-archiver
        maven-status
        Euge-1.0-jar-with-dependencies.jar
        Euge-1.0.jar
      pom.xml

TextClientApplication.java
EugeSpeechletRequestStreamHandler.java

1 package main;
2
3 import java.util.HashSet;
4
5 /**
6 * 
7 */
8
9 public class EugeSpeechletRequestStreamHandler extends SpeechletRequestStreamHandler {
10     private static final Set<String> supportedApplicationIds = new HashSet<String>();
11
12     static {
13         // skill id
14         supportedApplicationIds.add("INSERT AMAZON SKILL ID HERE");
15     }
16
17     public EugeSpeechletRequestStreamHandler() {
18         super(new EugeSpeechlet(), supportedApplicationIds);
19     }
20
21 }
22
23
24
25

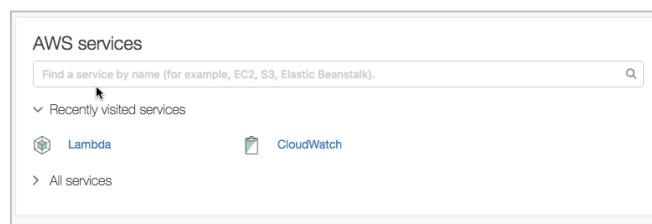
```

## CREATE A LAMBDA FUNCTION

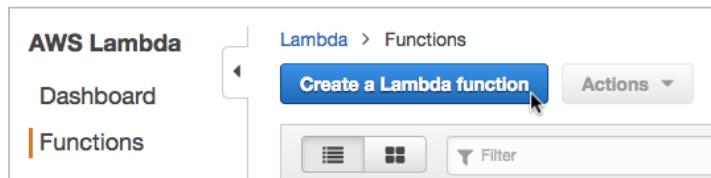
1. Sign in to <https://aws.amazon.com/>
  
2. Select Region (must be either N. Virginia or Ireland for Alexa functions)



3. Search and select Lambda Service



4. Select Create a Lambda Function



## 5. Select Blank Function Blueprint

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 78

<b>Blank Function</b> 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
---	---	---

## 6. Click grey dashed box to Configure Triggers

- Select Alexa Skills Kit
- Click Next

Configure triggers

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

Remove

Cancel Previous Next

Filter integrations

- API Gateway
- AWS IoT
- Alexa Skills Kit
- Alexa Smart Home
- CloudFront
- CloudWatch Events - Schedule
- CloudWatch Logs
- CodeCommit

Configure triggers

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

Alexa Skills Kit → Lambda Remove

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.

Cancel Previous Next

## 7. Configure Function

- Enter Function Name
- Enter Function Description
- Select Runtime: Java 8

Configure function

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

Name\*

Description

Runtime\*  



## 8. Upload Function Code

- Select Upload a .ZIP or JAR file
- Click Upload and select file
- Leave Enable Encryption Helpers unchecked
- Leave Environment Variables blank

Lambda function code

Provide the code for your function. [Learn more](#) about deploying Lambda functions.

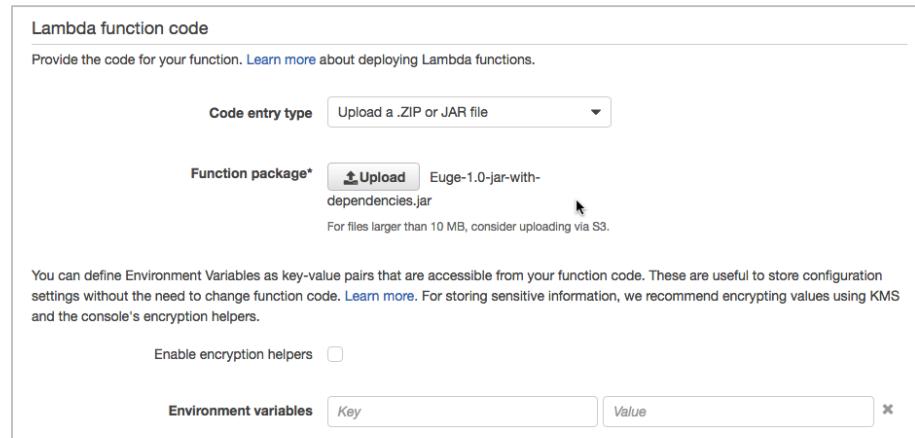
Code entry type  

Function package\*    Euge-1.0-jar-with-dependencies.jar  
For files larger than 10 MB, consider uploading via S3.

You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#). For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.

Enable encryption helpers

Environment variables   



## 9. Specify Function Handler and Role

- Paste this in Handler field:  
`EugeSpeechletRequestStreamHandler.java`
- Role: Choose an existing role
- Existing Role: `lambda_basic_execution`
- Ignore Advanced Settings
- Click Next

Lambda function handler and role

Handler\*  

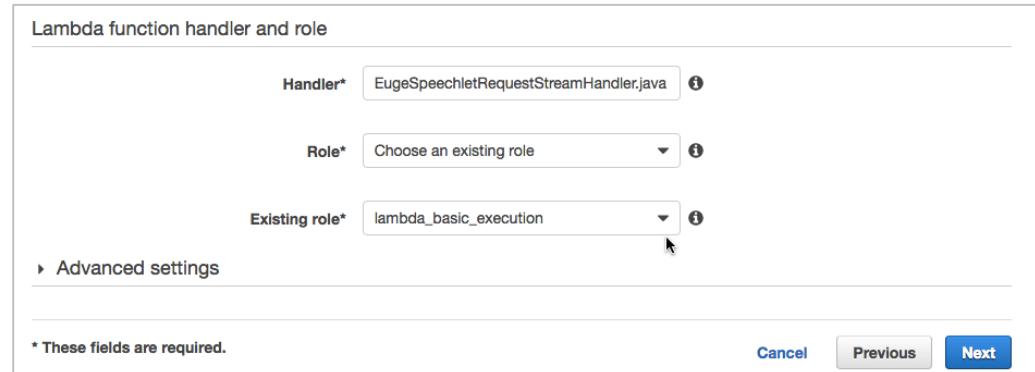
Role\*   

Existing role\*   

 Advanced settings

\* These fields are required.

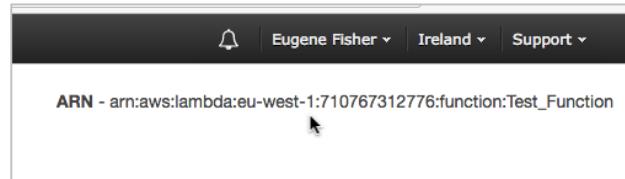
[Cancel](#) [Previous](#) [Next](#)



10. Review Function Details  
 a. Click Create Function



11. Copy function ARN to clipboard and



12. Close Lambda

## COMPLETE ALEXA SKILL

*\*navigate back to Amazon Developer Portal\**

10. Endpoint Information
- Service Endpoint Type: AWS Lambda ARN
  - Location: Select same country as ARN location
  - Paste ARN from Lambda Function in endpoint field
  - For chat only skills, no need to allow users to create or link accounts
  - Click Next

English (U.S.) Add New Language

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

**Endpoint**

Service Endpoint Type:  AWS Lambda ARN (Amazon Resource Name) Recommended  HTTPS

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

Europe

**Account Linking**

Do you allow users to create an account or link to an existing account with you?  Yes  No  
[Learn more](#)

**Buttons:** Save, Submit for Certification, Next

11. Complete Publishing Information and Privacy & Compliance when ready to publish

a. Complete Publishing Information

- i. Specify Category, Testing Instructions and Country Availability
- ii. Specify Skill Description, Example Phrases, Keywords, Upload Icons
- iii. Click next

b. Complete Privacy & Compliance

- i. Specify Privacy & Compliance Settings
- ii. Free Privacy Policy Generator: [www.iubenda.com](http://www.iubenda.com)
- iii. Free Terms of Use Generator: [www.termsfeed.com](http://www.termsfeed.com)
- iv. Click Submit for Certification; note skills can take up to 21 days for review

Skill Information	
Interaction Model	
Configuration	
Test	
Publishing Information	
Privacy & Compliance	

\*\*\* CONGRATULATIONS \*\*\*

You've successfully built an Alexa skill with API.ai and Lambda!

Thanks for checking out this tutorial and be sure to share with others.

Questions? Feedback?

Have you built a skill with this tutorial?

Let me know: [bfisher@ebfour.com](mailto:bfisher@ebfour.com)