How To Setup

Amazon Alexa Voice Control Service

On the DragonBoard™ 410c

This project demonstrates how to access and test the Alexa Voice Service on a DragonBoard 410c using a Java client and a Node.js server. You will be using the Node.js server to get a Login with Amazon authorization code by visiting a website using your computer's (or DragonBoard 410c in this case) web browser.

This guide is based on the Instructions published by Amazon for RaspberryPi here:
https://github.com/amzn/alexa-avs-raspberry-pi

Contents

1 Install New Software

Before we install any new software lets update our package list first:

Open a new Termial-window: CTRL-ALT-T

```
sudo apt-get update
```

1.1 Install VLC media player

Get VLC media player by typing:

```
sudo apt-get install vlc-nox vlc-data
```

Make sure VLC is installed correctly (This will tell you where VLC is installed.):

```
whereis vlc
```

Set the environment variables for VLC:

```
export LD_LIBRARY_PATH=/usr/lib/vlc
export VLC_PLUGIN_PATH=/usr/lib/vlc/plugins
```

Check if the environment variables were set successfully:

```
echo $LD_LIBRARY_PATH
> /usr/lib/vlc

echo $VLC_PLUGIN_PATH
> /usr/lib/vlc/plugins
```

1.2 Install Node.JS

Node JS should come pre-installed on the latest DragonBoard images.

To verify if NodeJs is available on your system type:

```
node --version
> command not found
```

If this command returns the version information of the Node version installed you can skip this step and proceed to the next step.

Otherwise continue with these instructions to install NodeJS:

```
sudo apt-get install nodejs
```

1.3 Install Java development kit

Java should come pre-installed on the latest DragonBoard images and you should be able to skip this step.

To verify if Java is available on your system type:

```
java -version
javac –version
```

If this command returns the version information of the Java version installed you can skip this step and proceed to the next step.

Otherwise continue with these instructions to install Java:

```
sudo apt-get install openjdk-8-jdk
```

1.4 Install Maven

Download Maven

Download the file **apache-maven-3.3.9-bin.tar.gz** from: https://maven.apache.org/download.cgi
or http://mirror.metrocast.net/apache/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz

Extract the contents of the tarball to the /opt directory:

```
sudo tar zxvf apache-maven-3.3.9-bin.tar.gz -C /opt
```

Tell your shell where to find maven.

Edit your **~/.bashrc** file and type the following inside it:

```
export M2_HOME=/opt/apache-maven-3.3.9
export PATH=$PATH:$M2_HOME/bin
```

(If you like, you can also edit the system profile settings so it is available to all users by creating a new file /etc/profile.d/maven.sh and adding in the export lines above.)

Save the file.

Log out and back into the DragonBoard410c so the profile script takes effect. You can test that it is working with the following command:

```
mvn –version
```

1.5 Install oppenSSL

OpenSSL should come pre-installed on the latest DragonBoard images and you should be able to skip this step.

To verify if OpenSSL is available on your system type:

Verify install

```
whereis openssl
> openssl: /usr/bin/openssl /usr/share/man/man1/openssl.1ssl.gz
```

If this command returns correctly you can skip this step and proceed to the next step.

Otherwise continue with these instructions to install OpenSSL:

Install OpenSSL

```
sudo apt-get install openssl
```

Update CA-certificates:

```
sudo update-ca-certificates
```

2 Getting started with Alexa Voice Service

2.1 Download the sample app code and dependencies on the DragonBoard410c

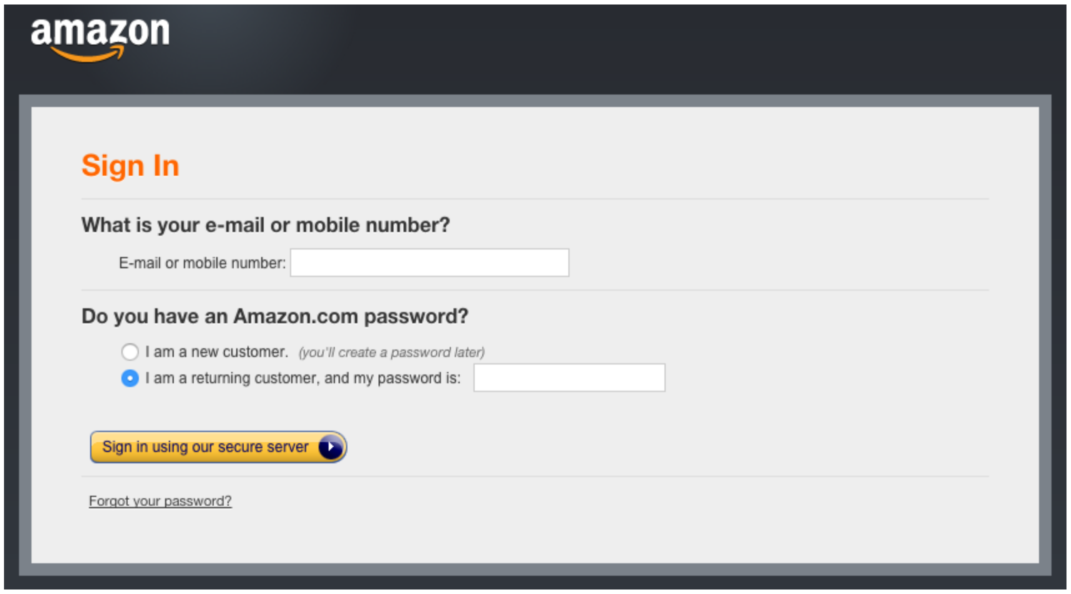
Download the sample apps:

```
mkdir ~/Projects
mkdir ~/Projects/alexa
cd ~/Projects/alexa
git clone https://github.com/amzn/alexa-avs-raspberry-pi.git
```

By downloading this package, you agree to the Alexa Voice Service Agreement.

2.2 Register for a free Amazon Developer Account

Get a free Amazon developer account if you do not already have one: https://developer.amazon.com/login.html



Sign In

What is your e-mail or mobile number?

E-mail or mobile number:

Do you have an Amazon.com password?

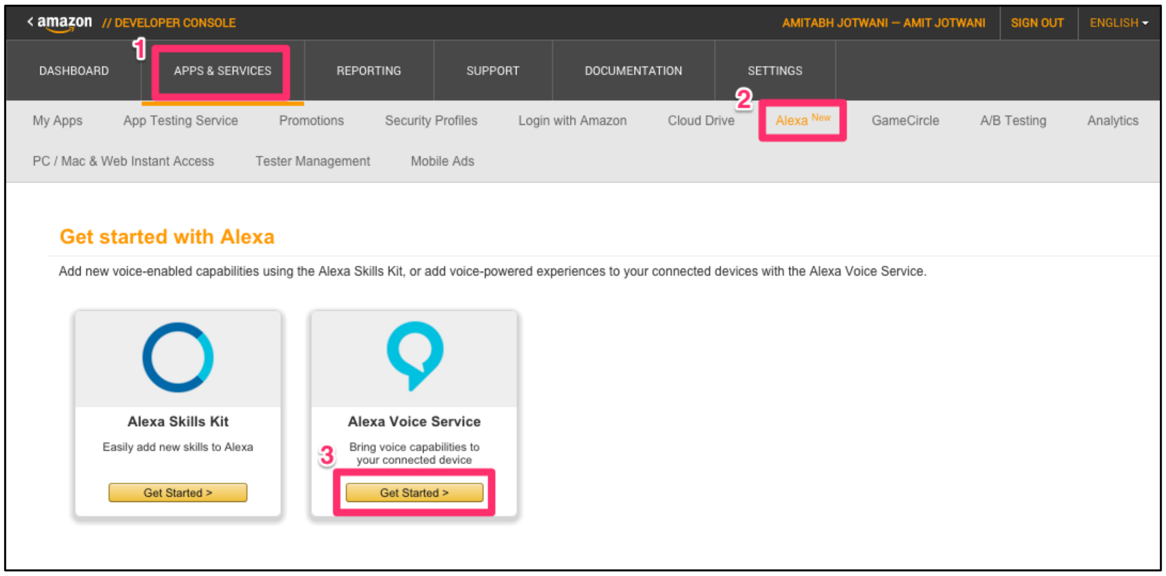
I am a new customer. (you'll create a password later)

[✓ I am a returning customer, and my password is:]

2.3 Register your product and create a security profile.

1. Login to Amazon Developer Portal - developer.amazon.com

2. Click on Apps & Services tab -> Alexa -> Alexa Voice Service -> Get Started



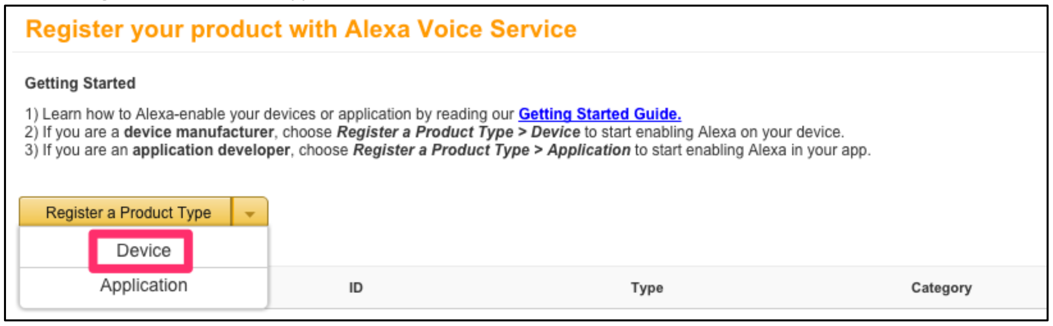
Get started with Alexa

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

Alexa Skills Kit: Easily add new skills to Alexa

Alexa Voice Service: Bring voice capabilities to your connected device

3. In the Register a Product Type menu, select **Device**



Register your product with Alexa Voice Service

Getting Started

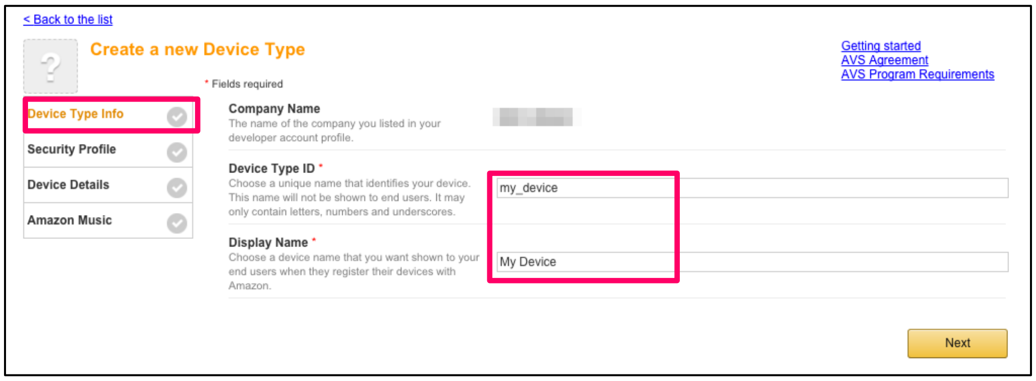
1) Learn how to Alexa-enable your devices or application by reading our Getting Started Guide.
2) If you are a **device manufacturer**, choose ***Register a Product Type* > *Device*** to start enabling Alexa on your device.
3) If you are an **application developer**, choose ***Register a Product Type* > *Application*** to start enabling Alexa in your app.

4. Fill in and save the following values:

Device Type Info:

- Device Type ID: **my_device**
- Display Name: **My Device**

Then click **Next**



Create a new Device Type

* Fields required

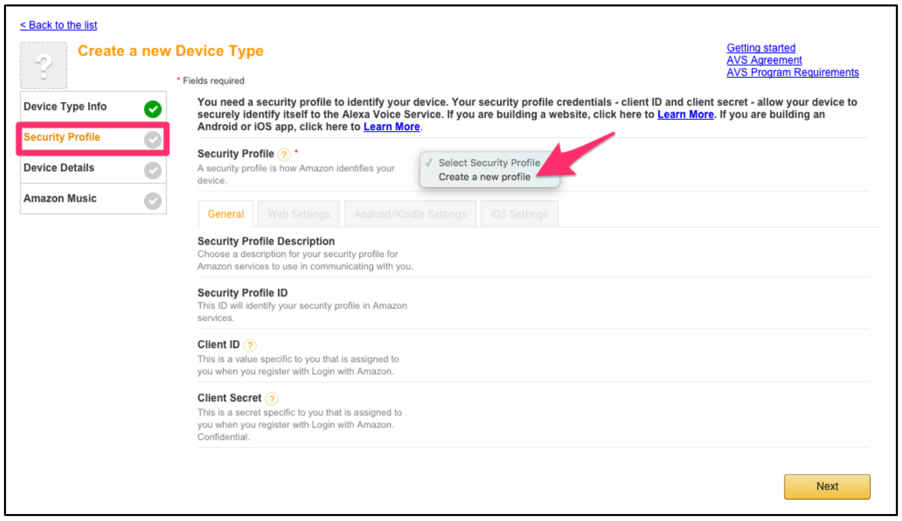
Company Name: [redacted]
The name of the company you listed in your developer account profile.

Device Type ID *: my_device
Choose a unique name that identifies your device. This name will not be shown to end users. It may only contain letters, numbers and underscores.

Display Name *: My Device
Choose a device name that you want shown to your end users when they register their devices with Amazon.

Security Profile

5. Click on the Security Profile dropdown and choose "**Create a new profile**" :



< Back to the list

Create a new Device Type

Getting started
AVS Agreement
AVS Program Requirements

* Fields required

Device Type Info
Security Profile
Device Details
Amazon Music

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to Learn More. If you are building an Android or iOS app, click here to Learn More.

Security Profile *
A security profile is how Amazon identifies your device.
✓ Select Security Profile
Create a new profile

General | Web Settings | Android/Kindle Settings | iOS Settings

Security Profile Description
Choose a description for your security profile for Amazon services to use in communicating with you.

Security Profile ID
This ID will identify your security profile in Amazon services.

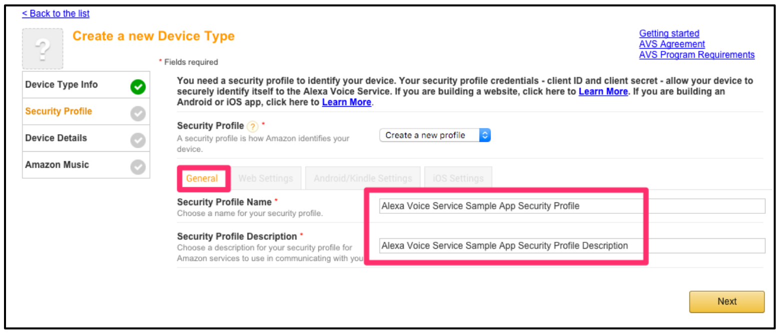
Client ID
This is a value specific to you that is assigned to you when you register with Login with Amazon.

Client Secret
This is a secret specific to you that is assigned to you when you register with Login with Amazon. Confidential.

Next

6. **General Tab**

- **Security Profile Name**: Alexa Voice Service Sample App Security Profile
- **Security Profile Description**: Alexa Voice Service Sample App Security Profile Description
- Click **Next**



< Back to the list

Create a new Device Type

Getting started
AVS Agreement
AVS Program Requirements

* Fields required

Device Type Info
Security Profile
Device Details
Amazon Music

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to Learn More. If you are building an Android or iOS app, click here to Learn More.

Security Profile *
A security profile is how Amazon identifies your device.
Create a new profile

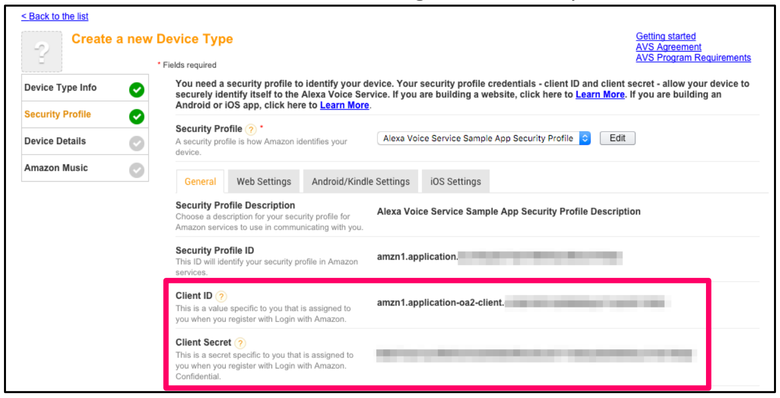
General | Web Settings | Android/Kindle Settings | iOS Settings

Security Profile Name *
Choose a name for your security profile.
Alexa Voice Service Sample App Security Profile

Security Profile Description *
Choose a description for your security profile for Amazon services to use in communicating with you.
Alexa Voice Service Sample App Security Profile Description

Next

The **Client** ID and Client Secret will be generated for you.



< Back to the list

Create a new Device Type

Getting started
AVS Agreement
AVS Program Requirements

* Fields required

Device Type Info
Security Profile
Device Details
Amazon Music

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to Learn More. If you are building an Android or iOS app, click here to Learn More.

Security Profile *
A security profile is how Amazon identifies your device.
Alexa Voice Service Sample App Security Profile | Edit

General | Web Settings | Android/Kindle Settings | iOS Settings

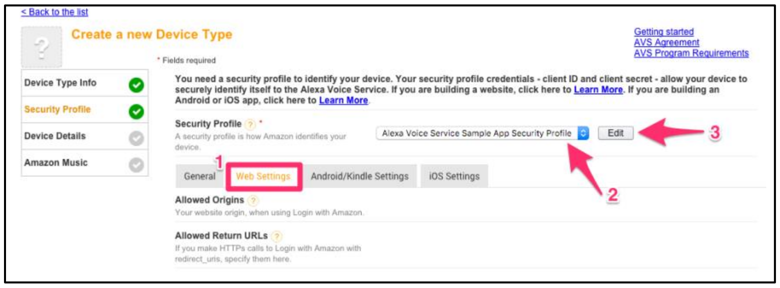
Security Profile Description
Choose a description for your security profile for Amazon services to use in communicating with you.
Alexa Voice Service Sample App Security Profile Description

Security Profile ID
This ID will identify your security profile in Amazon services.
amzn1.application.[redacted]

Client ID
This is a value specific to you that is assigned to you when you register with Login with Amazon.
amzn1.application-oa2-client.[redacted]

Client Secret
This is a secret specific to you that is assigned to you when you register with Login with Amazon. Confidential.
[redacted]

7. Now click on the **Web Settings Tab**
   - Make sure the security profile you just created is selected in the drop-down menu, then click the **"Edit"** button.



< Back to the list

Create a new Device Type

* Fields required

Getting started
AVS Agreement
AVS Program Requirements

Device Type Info ✓
Security Profile ✓
Device Details
Amazon Music

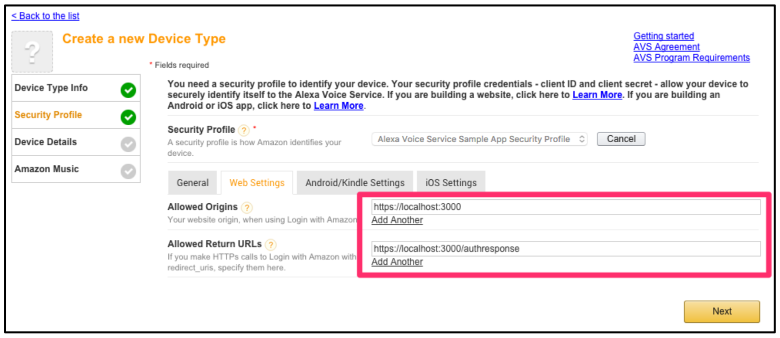
You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to Learn More. If you are building an Android or iOS app, click here to Learn More.

Security Profile ? *
A security profile is how Amazon identifies your device.
Alexa Voice Service Sample App Security Profile — Edit

Allowed Origins ?
Your website origin, when using Login with Amazon.

Allowed Return URLs ?
If you make HTTPs calls to Login with Amazon with redirect_uris, specify them here.

   - **Allowed Origins**: Click "**Add Another**" and then enter **https://localhost:3000** in the text field that appears.
   - **Allowed Return URLs**: Click "Add Another" and then enter **https://localhost:3000/authresponse** in the text field that appears.
   - Click **Next**



< Back to the list

Create a new Device Type

* Fields required

Getting started
AVS Agreement
AVS Program Requirements

Device Type Info ✓
Security Profile ✓
Device Details
Amazon Music

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to Learn More. If you are building an Android or iOS app, click here to Learn More.

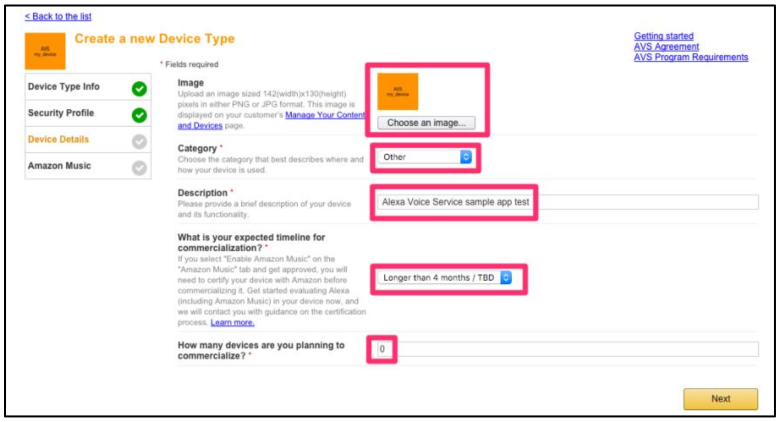
Security Profile ? *
A security profile is how Amazon identifies your device.
Alexa Voice Service Sample App Security Profile — Cancel

Allowed Origins ?
Your website origin, when using Login with Amazon
https://localhost:3000
Add Another

Allowed Return URLs ?
If you make HTTPs calls to Login with Amazon with redirect_uris, specify them here.
https://localhost:3000/authresponse
Add Another

8. **Device Details**
   - [Optional] Image: Save the following test image to your computer, then upload it:

   - Category: **Other**
   - Description: **Alexa Voice Service sample app test**
   - What is your expected timeline for commercialization?: **Longer than 4 months / TBD**
   - How many devices are you planning to commercialize?: **0**
   - Click **Next** *(see screen shot on next page)*


< Back to the list

Create a new Device Type

Getting started
AVS Agreement
AVS Program Requirements

* Fields required

Device Type Info ✓
Security Profile ✓
Device Details
Amazon Music

Image
Upload an image sized 142(width)x130(height) pixels in either PNG or JPG format. This image is displayed on your customer's Manage Your Content and Devices page.
Choose an image...

Category *
Choose the category that best describes where and how your device is used.
Other

Description *
Please provide a brief description of your device and its functionality.
Alexa Voice Service sample app test

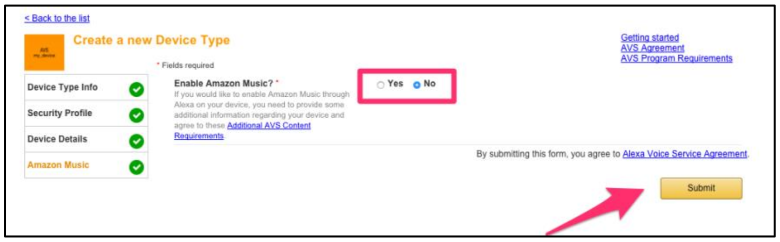
What is your expected timeline for commercialization? *
If you select "Enable Amazon Music" on the "Amazon Music" tab and get approved, you will need to certify your device with Amazon before commercializing it. Get started evaluating Alexa (including Amazon Music) in your device now, and we will contact you with guidance on the certification process. Learn more.
Longer than 4 months / TBD

How many devices are you planning to commercialize? *
0

Next

9. **Amazon Music**

- Enable Amazon Music?: **No** (You may optionally select Yes and fill in the required fields if you want to experiment with Amazon Music. However, Amazon Music is not required for basic use of the Alexa Voice Service.)
- Click the **Submit** button


< Back to the list

Create a new Device Type

Getting started
AVS Agreement
AVS Program Requirements

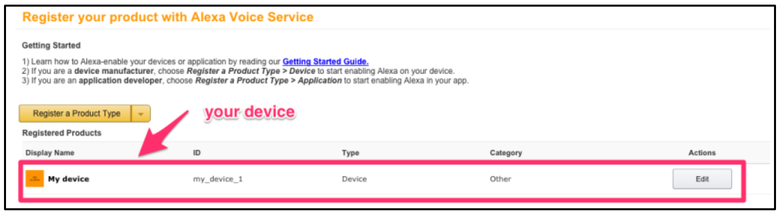
* Fields required

Device Type Info ✓
Security Profile ✓
Device Details ✓
Amazon Music ✓

Enable Amazon Music? *
If you would like to enable Amazon Music through Alexa on your device, you need to provide some additional information regarding your device and agree to these Additional AVS Content Requirements
Yes [✓ No]

By submitting this form, you agree to Alexa Voice Service Agreement.

Submit


Register your product with Alexa Voice Service

Getting Started

1) Learn how to Alexa-enable your devices or application by reading our Getting Started Guide.
2) If you are a **device manufacturer**, choose ***Register a Product Type > Device*** to start enabling Alexa on your device.
3) If you are an **application developer**, choose ***Register a Product Type > Application*** to start enabling Alexa in your app.

Register a Product Type

Registered Products

Display Name	ID	Type	Category	Actions
My device	my_device_1	Device	Other	Edit

2.4 Generate self-signed certificates.

Change directories to <REFERENCE_IMPLEMENTATION>/samples/javaclient. In this instance, the <REFERENCE_IMPLEMENTATION> is "/home/linaro/Projects/alexa/alexa-avs-raspberry-pi/":

```
cd <REFERENCE_IMPLEMENTATION>/samples/javaclient - //your sample apps
location
```

Step 1: Edit the text file **ssl.cnf**, which is an SSL configuration file. Fill in appropriate values in place of the placeholder text that starts with **YOUR_**.

```
[req_distinguished_name]
commonName               = $ENV::COMMON_NAME                 # CN=
countryName              = US # C=
stateOrProvinceName      = CA # ST=
localityName             = San Diego # L=
organizationName         = Qualcomm Technologies, Inc. # O=
organizationalUnitName   = Qualcomm Developer Network
```

Note that **countryName** must be two characters. If it is not two characters, certificate creation will fail. Here's what the **ssl.cnf** file would look like, replacing country, state, locality with your respective info.

Save the file.

Step 2: Make the certificate generation script executable by typing:

```
chmod +x generate.sh
```

Step 3: Run the certificate generation script:

```
./generate.sh
```

Step 4: You will be prompted for some information:

- When prompted for a product ID, enter **my_device**
- When prompted for a serial number, enter **123456**
- When prompted for a password, enter any password and remember what you entered
  - Password: **talktome** (you can even leave it blank)

Step 5: Edit the configuration file for the Node.js server

The configuration file is located at: <REFERENCE_IMPLEMENTATION>/samples/companionService/config.js.

Make the following changes:

- Set **sslKey** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/server/node.key
- Set **sslCert** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/server/node.crt
- Set **sslCaCert** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/ca/ca.crt

IMPORTANT: **Do not** use **~** to denote the home directory. Use the absolute path instead.
So in this case, use /home/linaro/Projects/alexa/alexa-avs-raspberry-pi/samples/…

Save the file.

Step 6: Edit the configuration file for the Java client

The configuration file is located at:

<REFERENCE_IMPLEMENTATION>/samples/javaclient/config.json.

Make the following changes:

- Set **companionApp.sslKeyStore** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/server/jetty.pkcs12
- Set **companionApp.sslKeyStorePassphrase** to the passphrase entered in the certificate generation script in step 4 above.
- Set **companionService.sslClientKeyStore** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/client/client.pkcs12

- Set **companionService.sslClientKeyStorePassphrase** to the passphrase entered in the certificate generation script in step 4 above.
- Set **companionService.sslCaCert** to <REFERENCE_IMPLEMENTATION>/samples/javaclient/certs/ca/ca.crt

2.5 Install the Companion Service dependencies

Change directories to <REFERENCE_IMPLEMENTATION>/samples/companionService

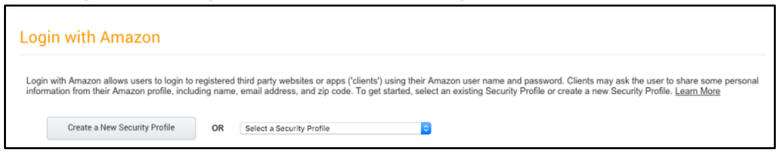
```
cd <REFERENCE_IMPLEMENTATION>/samples/companionService
```

Install the dependencies by typing:

```
npm install
```

2.6 Create Security Profile

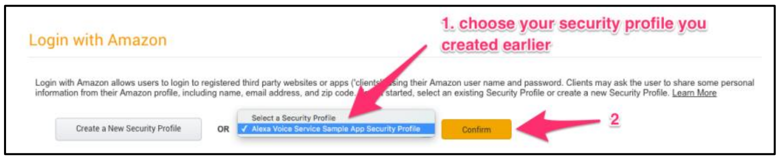
1. Open a web browser, and visit https://developer.amazon.com/lwa/sp/overview.html.



Login with Amazon

Login with Amazon allows users to login to registered third party websites or apps ('clients') using their Amazon user name and password. Clients may ask the user to share some personal information from their Amazon profile, including name, email address, and zip code. To get started, select an existing Security Profile or create a new Security Profile. Learn More

2. Near the top of the page, select the security profile you created earlier from the drop down menu and click **Confirm**.

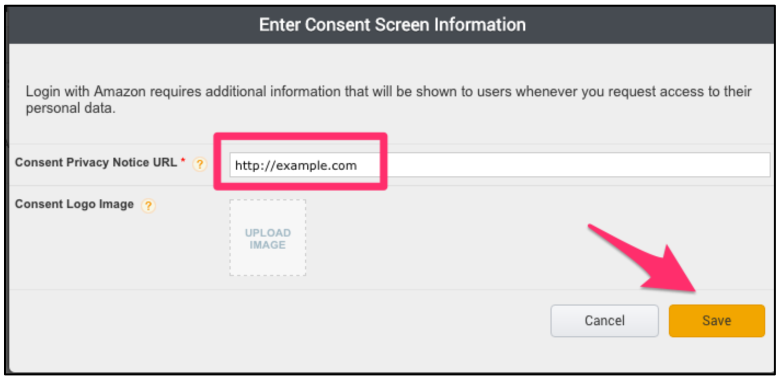


1. choose your security profile you created earlier
Login with Amazon
Create a New Security Profile
OR
Select a Security Profile
✓ Alexa Voice Service Sample App Security Profile
Confirm
2

3. Enter a privacy policy URL beginning with http:// or https://. For this example, you can enter a fake URL such as http://example.com.

4. [Optional] You may upload an image as well. The image will be shown on the Login with Amazon consent page to give your users context.

5. Click Save.

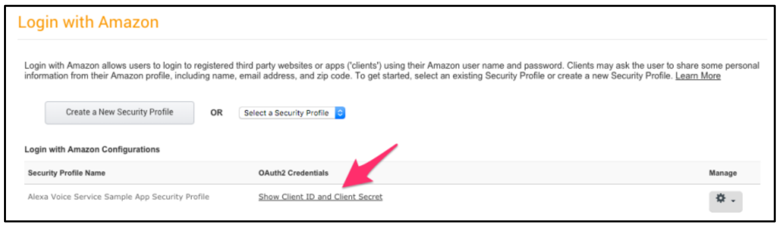


Enter Consent Screen Information

Login with Amazon requires additional information that will be shown to users whenever you request access to their personal data.

Consent Privacy Notice URL *: http://example.com

6. Next to the **Alexa Voice Service Sample App Security Profile**, click **Show Client ID and Client Secret**. This will display your client ID and client secret. Save these values. You'll need these.



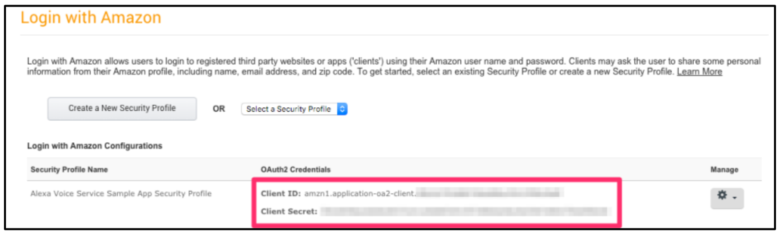
Login with Amazon

Login with Amazon allows users to login to registered third party websites or apps ('clients') using their Amazon user name and password. Clients may ask the user to share some personal information from their Amazon profile, including name, email address, and zip code. To get started, select an existing Security Profile or create a new Security Profile. Learn More

Create a New Security Profile OR Select a Security Profile

Login with Amazon Configurations

Security Profile Name	OAuth2 Credentials	Manage
Alexa Voice Service Sample App Security Profile	Show Client ID and Client Secret	



Login with Amazon

Login with Amazon allows users to login to registered third party websites or apps ('clients') using their Amazon user name and password. Clients may ask the user to share some personal information from their Amazon profile, including name, email address, and zip code. To get started, select an existing Security Profile or create a new Security Profile. Learn More

Create a New Security Profile OR Select a Security Profile

Login with Amazon Configurations

Security Profile Name	OAuth2 Credentials	Manage
Alexa Voice Service Sample App Security Profile	Client ID: amzn1.application-oa2-client.[redacted] Client Secret: [redacted]	

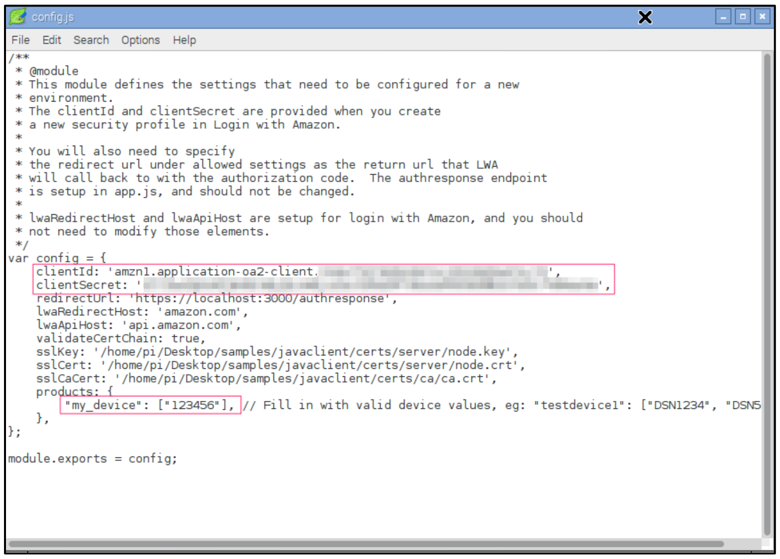
2.7 Updating the config files

Step 1: Update **config.js** Navigate to the following file and open it in your favorite text editor.

<REFERENCE_IMPLEMENTATION>/samples/companionService/config.js

Edit the following values in this file:

- **clientId**: Paste in the client ID that you noted in the previous step as a string.
- **clientSecret**: Paste in the client secret that you noted in the previous step as a string.
- **products**: The product's object consists of a key that should be the same as the product type ID that you set up in the developer portal and a value that is an array of unique product identifiers. If you followed the instructions above, the product type ID should be **my_device**. The unique product identifier can be any alphanumeric string, such as 123456. Example products JSON is: **products: {"my_device": ["123456"]}**



```
/**
 * @module
 * This module defines the settings that need to be configured for a new
 * environment.
 * The clientId and clientSecret are provided when you create
 * a new security profile in Login with Amazon.
 *
 * You will also need to specify
 * the redirect url under allowed settings as the return url that LWA
 * will call back to with the authorization code.  The authresponse endpoint
 * is setup in app.js, and should not be changed.
 *
 * lwaRedirectHost and lwaApiHost are setup for login with Amazon, and you should
 * not need to modify those elements.
 */
var config = {
    clientId: 'amzn1.application-oa2-client.[redacted]',
    clientSecret: '[redacted]',
    redirectUrl: 'https://localhost:3000/authresponse',
    lwaRedirectHost: 'amazon.com',
    lwaApiHost: 'api.amazon.com',
    validateCertChain: true,
    sslKey: '/home/pi/Desktop/samples/javaclient/certs/server/node.key',
    sslCert: '/home/pi/Desktop/samples/javaclient/certs/server/node.crt',
    sslCaCert: '/home/pi/Desktop/samples/javaclient/certs/ca/ca.crt',
    products: {
        "my_device": ["123456"], // Fill in with valid device values, eg: "testdevice1": ["DSN1234", "DSN5
    },
};

module.exports = config;
```

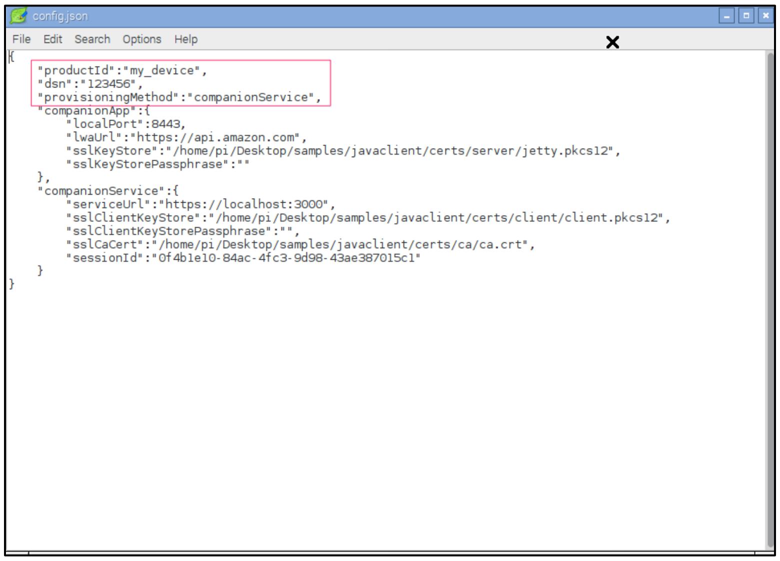
Save the file.

Step 2: Update **config.json** Navigate to the following file, and open it in a text editor.

<REFERENCE_IMPLEMENTATION>/samples/javaclient/config.json

Edit the following values in this file:

- **productId**: Enter **my_device** as a string.
- **dsn**: Enter the alphanumeric string that you used for the unique product identifier in the products object in the server's config.js. For example: **123456**.
- **provisioningMethod**: Enter **companionService**.



```
{
    "productId":"my_device",
    "dsn":"123456",
    "provisioningMethod":"companionService",
    "companionApp":{
        "localPort":8443,
        "lwaUrl":"https://api.amazon.com",
        "sslKeyStore":"/home/pi/Desktop/samples/javaclient/certs/server/jetty.pkcs12",
        "sslKeyStorePassphrase":""
    },
    "companionService":{
        "serviceUrl":"https://localhost:3000",
        "sslClientKeyStore":"/home/pi/Desktop/samples/javaclient/certs/client/client.pkcs12",
        "sslClientKeyStorePassphrase":"",
        "sslCaCert":"/home/pi/Desktop/samples/javaclient/certs/ca/ca.crt",
        "sessionId":"0f4b1e10-84ac-4fc3-9d98-43ae387015c1"
    }
}
```

Save the file.

Step 3: Preparing the pom.xml file

Navigate to the following file and open it in a text editor.

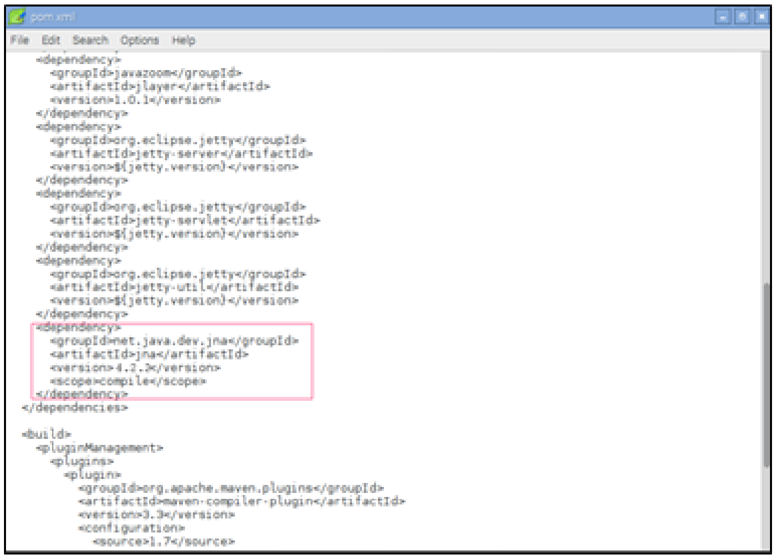
<REFERENCE_IMPLEMENTATION>/samples/javaclient/pom.xml

- In the **<dependency>** section, change the VLCJ version number to 3.10.1:

```
<dependency>
  <groupId>uk.co.caprica</groupId>
  <artifactId>vlcj</artifactId>
  <version>3.10.1</version>
```

- Add the following in the **<dependency>** section:

```
<dependency>
  <groupId>net.java.dev.jna</groupId>
  <artifactId>jna</artifactId>
  <version>4.2.2</version>
  <scope>compile</scope>
</dependency>
```



```
  <dependency>
    <groupId>javazoom</groupId>
    <artifactId>jlayer</artifactId>
    <version>1.0.1</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jetty</groupId>
    <artifactId>jetty-server</artifactId>
    <version>${jetty.version}</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jetty</groupId>
    <artifactId>jetty-servlet</artifactId>
    <version>${jetty.version}</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jetty</groupId>
    <artifactId>jetty-util</artifactId>
    <version>${jetty.version}</version>
  </dependency>
  <dependency>
    <groupId>net.java.dev.jna</groupId>
    <artifactId>jna</artifactId>
    <version>4.2.3</version>
    <scope>compile</scope>
  </dependency>
</dependencies>

<build>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.3</version>
        <configuration>
          <source>1.7</source>
```

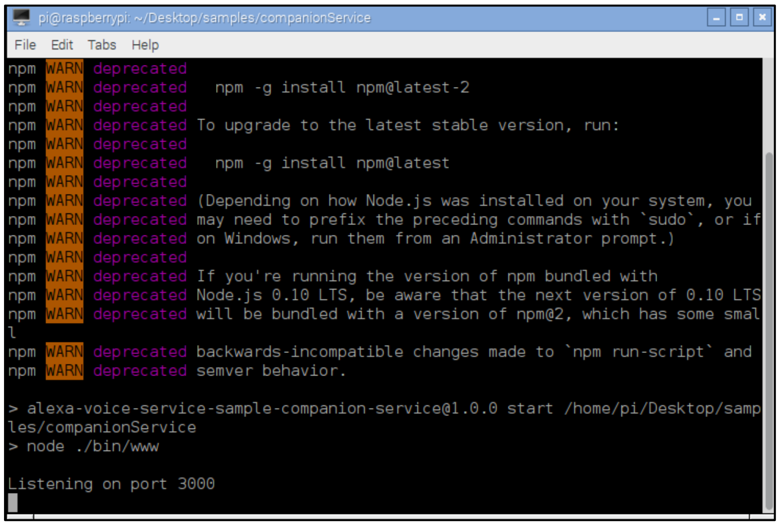
Save the file.

2.8 Run the server

Finally we are ready to start the companion service app

In your terminal window or from the command prompt, type:

```
cd <REFERENCE_IMPLEMENTATION>/samples/companionService
npm start
```



```
npm WARN deprecated
npm WARN deprecated   npm -g install npm@latest-2
npm WARN deprecated
npm WARN deprecated To upgrade to the latest stable version, run:
npm WARN deprecated
npm WARN deprecated   npm -g install npm@latest
npm WARN deprecated
npm WARN deprecated (Depending on how Node.js was installed on your system, you
npm WARN deprecated may need to prefix the preceding commands with `sudo`, or if
npm WARN deprecated on Windows, run them from an Administrator prompt.)
npm WARN deprecated
npm WARN deprecated If you're running the version of npm bundled with
npm WARN deprecated Node.js 0.10 LTS, be aware that the next version of 0.10 LTS
npm WARN deprecated will be bundled with a version of npm@2, which has some smal
l
npm WARN deprecated backwards-incompatible changes made to `npm run-script` and
npm WARN deprecated semver behavior.

> alexa-voice-service-sample-companion-service@1.0.0 start /home/pi/Desktop/samp
les/companionService
> node ./bin/www

Listening on port 3000
```

The server is now running on port 3000 and you are ready to start the client.

2.9 Start the client

Open a new terminal window/tab and change into the java-client directory:

```
cd <REFERENCE_IMPLEMENTATION>/samples/javaclient
```

2.9.1 Build the app

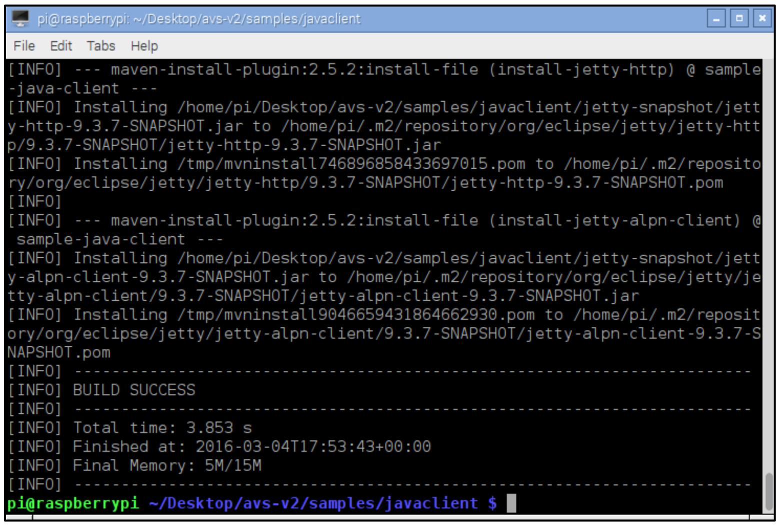
Before you build the app, let's validate to make sure the project is correct and that all necessary information is available. You do that by running:

```
mvn validate
```

Download dependencies and build the app by typing: (you might want to close any unnecessary apps or you might run into an error due to insufficient memory)

```
mvn install
```

When the installation is completed, you will see a "Build Success" message in the terminal:



```
[INFO] --- maven-install-plugin:2.5.2:install-file (install-jetty-http) @ sample
-java-client ---
[INFO] Installing /home/pi/Desktop/avs-v2/samples/javaclient/jetty-snapshot/jett
y-http-9.3.7-SNAPSHOT.jar to /home/pi/.m2/repository/org/eclipse/jetty/jetty-htt
p/9.3.7-SNAPSHOT/jetty-http-9.3.7-SNAPSHOT.jar
[INFO] Installing /tmp/mvninstall746896858433697015.pom to /home/pi/.m2/reposito
ry/org/eclipse/jetty/jetty-http/9.3.7-SNAPSHOT/jetty-http-9.3.7-SNAPSHOT.pom
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install-file (install-jetty-alpn-client) @
 sample-java-client ---
[INFO] Installing /home/pi/Desktop/avs-v2/samples/javaclient/jetty-snapshot/jett
y-alpn-client-9.3.7-SNAPSHOT.jar to /home/pi/.m2/repository/org/eclipse/jetty/je
tty-alpn-client/9.3.7-SNAPSHOT/jetty-alpn-client-9.3.7-SNAPSHOT.jar
[INFO] Installing /tmp/mvninstall9046659431864662930.pom to /home/pi/.m2/reposit
ory/org/eclipse/jetty/jetty-alpn-client/9.3.7-SNAPSHOT/jetty-alpn-client-9.3.7-S
NAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.853 s
[INFO] Finished at: 2016-03-04T17:53:43+00:00
[INFO] Final Memory: 5M/15M
[INFO] ------------------------------------------------------------------------
pi@raspberrypi ~/Desktop/avs-v2/samples/javaclient $
```

2.9.2 Run the client app:

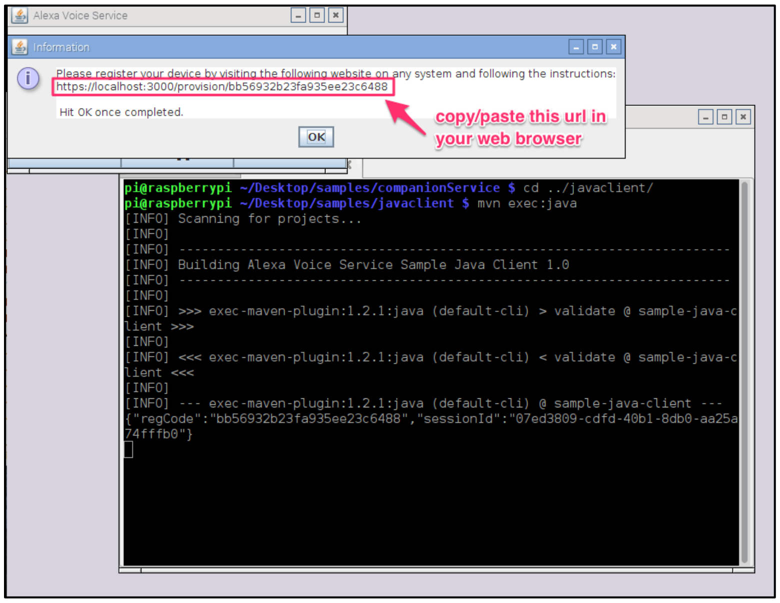
You are now ready to run the client app by typing:

```
mvn exec:exec
```

2.10 Obtain Authorization from Login with Amazon

1. When you run the client, a window should pop up with a message that says something similar to:

Please register your device by visiting the following website on any system and following the instructions: https://localhost:3000/provision/d340f629bd685deeff28a917 Hit OK once completed.



Alexa Voice Service

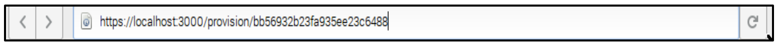
Information

Please register your device by visiting the following website on any system and following the instructions:
https://localhost:3000/provision/bb56932b23fa935ee23c6488

Hit OK once completed.

```
pi@raspberrypi ~/Desktop/samples/companionService $ cd ../javaclient/
pi@raspberrypi ~/Desktop/samples/javaclient $ mvn exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Alexa Voice Service Sample Java Client 1.0
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> exec-maven-plugin:1.2.1:java (default-cli) > validate @ sample-java-c
lient >>>
[INFO]
[INFO] <<< exec-maven-plugin:1.2.1:java (default-cli) < validate @ sample-java-c
lient <<<
[INFO]
[INFO] --- exec-maven-plugin:1.2.1:java (default-cli) @ sample-java-client ---
{"regCode":"bb56932b23fa935ee23c6488","sessionId":"07ed3809-cdfd-40b1-8db0-aa25a
74fffb0"}
```

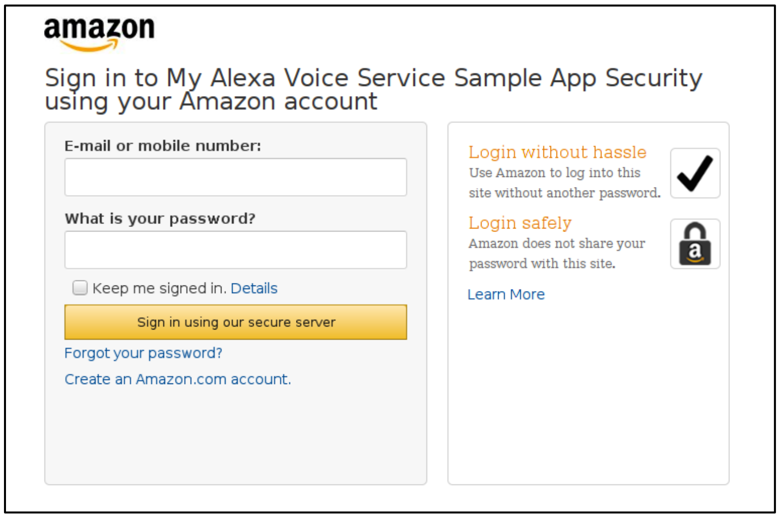
Copy the URL from the popup window and **paste** it into a **web browser**. In this example, the URL to copy and paste is https://localhost:3000/provision/d340f629bd685deeff28a917.



https://localhost:3000/provision/bb56932b23fa935ee23c6488

NOTE: Due to the use of a self-signed certificate, you will see a warning about an insecure website. This is expected. It is safe to ignore the warnings during testing.

2. You will be taken to a Login with Amazon web page. Enter your Amazon credentials.



amazon

Sign in to My Alexa Voice Service Sample App Security using your Amazon account

E-mail or mobile number:

What is your password?

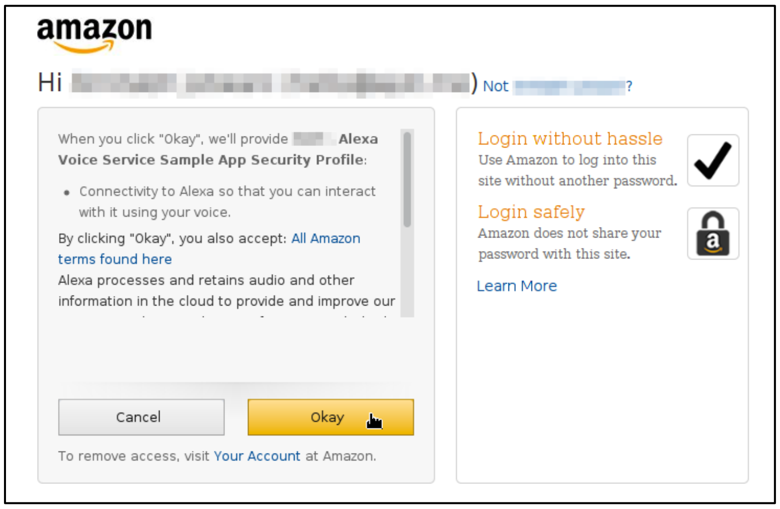
Keep me signed in. Details

Sign in using our secure server
Forgot your password?
Create an Amazon.com account.

Login without hassle
Use Amazon to log into this site without another password.

Login safely
Amazon does not share your password with this site.

3. You will be taken to a Dev Authorization page, confirming that you'd like your device to access the Security Profile created earlier. Click **Okay.**


Hi) Not ?

When you click "Okay", we'll provide Alexa Voice Service Sample App Security Profile:

- Connectivity to Alexa so that you can interact with it using your voice.

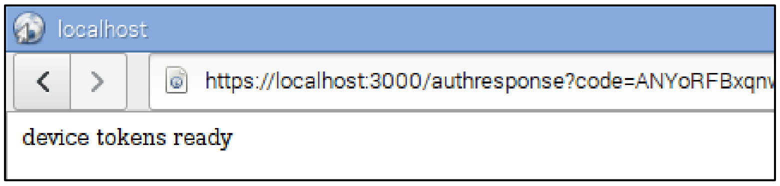
By clicking "Okay", you also accept: All Amazon terms found here

Alexa processes and retains audio and other information in the cloud to provide and improve our

amazon
Cancel
Okay
Login without hassle
Use Amazon to log into this site without another password.
Login safely
Amazon does not share your password with this site.
Learn More

To remove access, visit Your Account at Amazon.

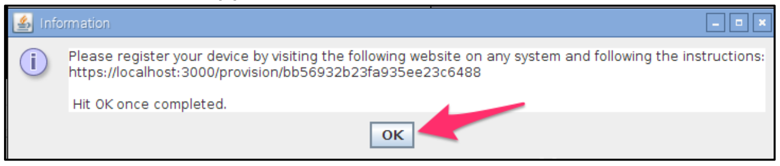
4. You will now be redirected to a URL beginning with https://localhost:3000/authresponse followed by a query string. The body of the web page will say **device tokens ready**.



https://localhost:3000/authresponse?code=ANYoRFBxqnv

device tokens ready

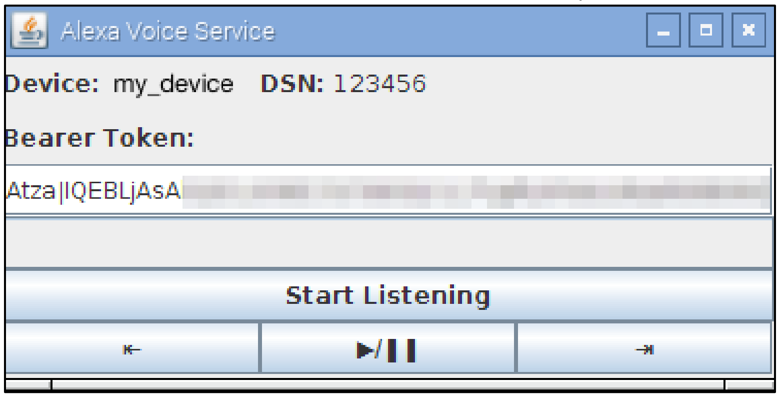
5. **Return to the Java application** and click the OK button. The client is now ready to accept Alexa requests.



Please register your device by visiting the following website on any system and following the instructions:
https://localhost:3000/provision/bb56932b23fa935ee23c6488

Hit OK once completed.

6. Wait a few seconds for the Token to appear in the Alexa Voice Service window.
Click the **Start Listening** button and wait for the **audio cue** before beginning to speak. It may take a second or two for the connection to be made before you hear the audio cue.

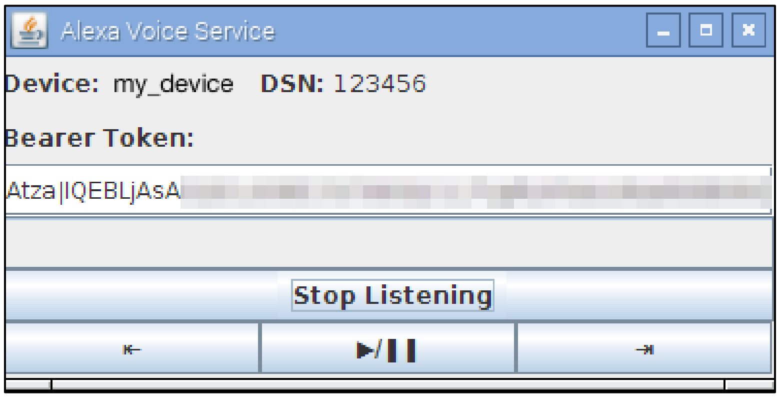


Device: my_device DSN: 123456

Bearer Token:

Atza|IQEBLjAsA

7. Press the **Stop Listening** button when you are done speaking.



Device: my_device DSN: 123456

Bearer Token:

Atza|IQEBLjAsA[redacted]

3 Let’s talk to Alexa

Ask for Weather: *Click the Start Listening button*. **You**: What's the weather in Seattle? *Click the Stop Listening button*.**Alexa**: Current weather report for Seattle

Some other fun questions you can ask Alexa

Once you hear the audio cue after clicking “Start Listening” button, here are a few things you can try saying -

- **Request Music Playback**: Play Bruce Springsteen
- **General Knowledge**: What's the mass of the sun in grams?
- **Geek**: What are the three laws of robotics?
- **Fun**: Can you rap?
- **Set a Timer**: Set the timer for 2 minutes.
- **Set Alarm**: Set the alarm for 7:30 a.m.

More on Music Playback The "previous", "play/pause", and "next" buttons at the bottom of the Java client UI are to demonstrate the music button events. Music button events allow you to initiate changes in the playback stream without having to speak to Alexa. For example, you can press the "play/pause" button to pause and restart a track of music.

To demonstrate the "play/pause" button, you can speak the following command: Play DC101 on iHeartRadio, then press the "play/pause" button. The music will pause in response to the button click. Press the "play/pause" button again to restart the music.