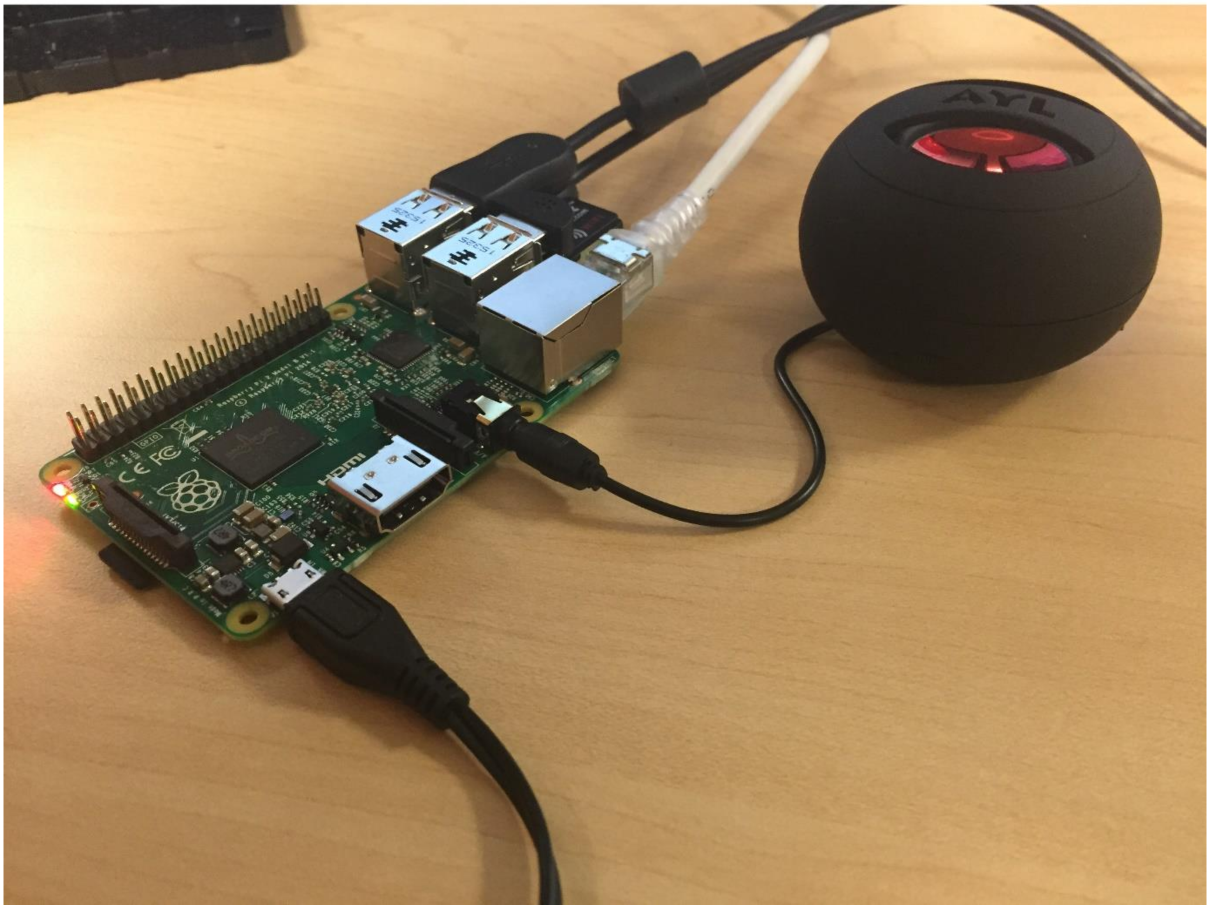
Appendix A

Alexa enabled RaspberryPI

This document is just a copy of the original post found on: https://github.com/alexa/alexa-avs-sample-app/wiki/Raspberry-Pi. The only reason for why this document was made is in case the original web-site will disappear and someone will want to recreate this project, then they could do so by following this tutorial.

Overview

amazon alexa Now with the wake word "Alexa"



This guide provides step-by-step instructions for setting up AVS on a **Raspberry Pi**. It demonstrates how to access and test AVS using our Java sample app (running on a Raspberry

Pi), a Node.js server, and a third-party wake word engine. You will use the Node.js server to obtain a Login with Amazon (LWA) authorization code by visiting a website using your Raspberry Pi's web browser.

For instructions on how to set it up on Mac, Linux or Windows, please see our wiki.

Required hardware

Before you get started, let's review what you'll need.

1. **Raspberry Pi 3** (Recommended) or **Pi 2 Model B** (Supported) - Buy at Amazon - Pi 3 or Pi 2.
2. **Micro-USB power cable** for Raspberry Pi.
3. **Micro SD Card** (Minimum 8 GB) - You need an operating system to get started. NOOBS (New Out of the Box Software) is an easy-to-use operating system install manager for Raspberry Pi. The simplest way to get NOOBS is to buy an SD card with NOOBS pre-installed - Raspberry Pi 8GB Preloaded (NOOBS) Micro SD Card. Alternatively, you can download and install it on your SD card (follow instructions here).
4. **USB 2.0 Mini Microphone** - Raspberry Pi does not have a built-in microphone; to interact with Alexa you'll need an external one to plug in - Buy on Amazon
5. **External Speaker** with 3.5mm audio cable - Buy on Amazon
6. A **USB Keyboard & Mouse**, and an external **HDMI Monitor** - we also recommend having a USB keyboard and mouse as well as an HDMI monitor handy if you're unable to remote(SSH)into your Pi.
7. Internet connection (Ethernet or WiFi)
8. (Optional) WiFi Wireless Adapter for Pi 2 (Buy on Amazon). **Note:** Pi 3 has built-in WiFi.

For extra credit, we'll show you how to remote(SSH) into your device, eliminating the need for a monitor, keyboard and mouse - and how to tail logs for troubleshooting.

Let's get started

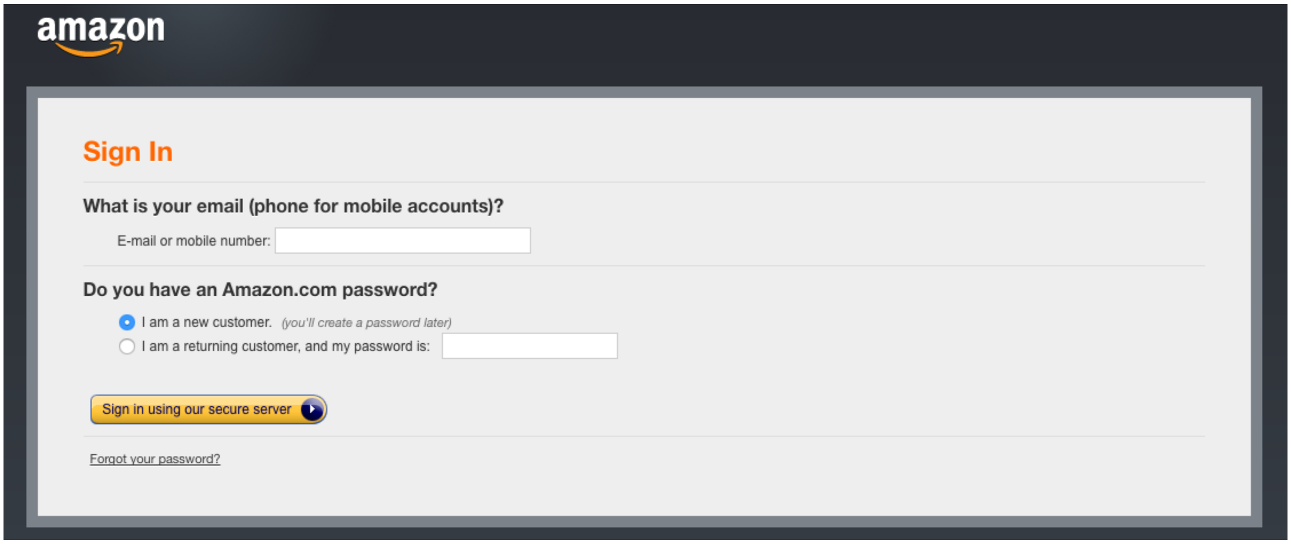
The original Alexa on a Pi project required manual download of libraries/dependencies and updating a series of configuration files that were prone to human error. To make the process faster and easier, we’ve included an install script with the project that will take care of all the heavy lifting. Not only does this reduce setup time to less than an hour on a Raspberry Pi 3, it only requires developers to adjust three variables in a single install script -

Step 1: Setting up your Pi

Unless you already have Raspbian Jessie installed on your Pi, please follow our guide - Setting up the Raspberry Pi - that will walk you through downloading and installing Raspbian Jessie, and connecting the hardware (if you're unfamiliar with Raspberry Pi, we **highly recommend** you follow the guide above to get your Pi up and ready before moving further).

Step 2: Register for an Amazon developer account

Unless you already have one, go ahead and create a free developer account at developer.amazon.com. You should review the **AVS Terms and Agreements** here.



Sign In

What is your email (phone for mobile accounts)?

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:

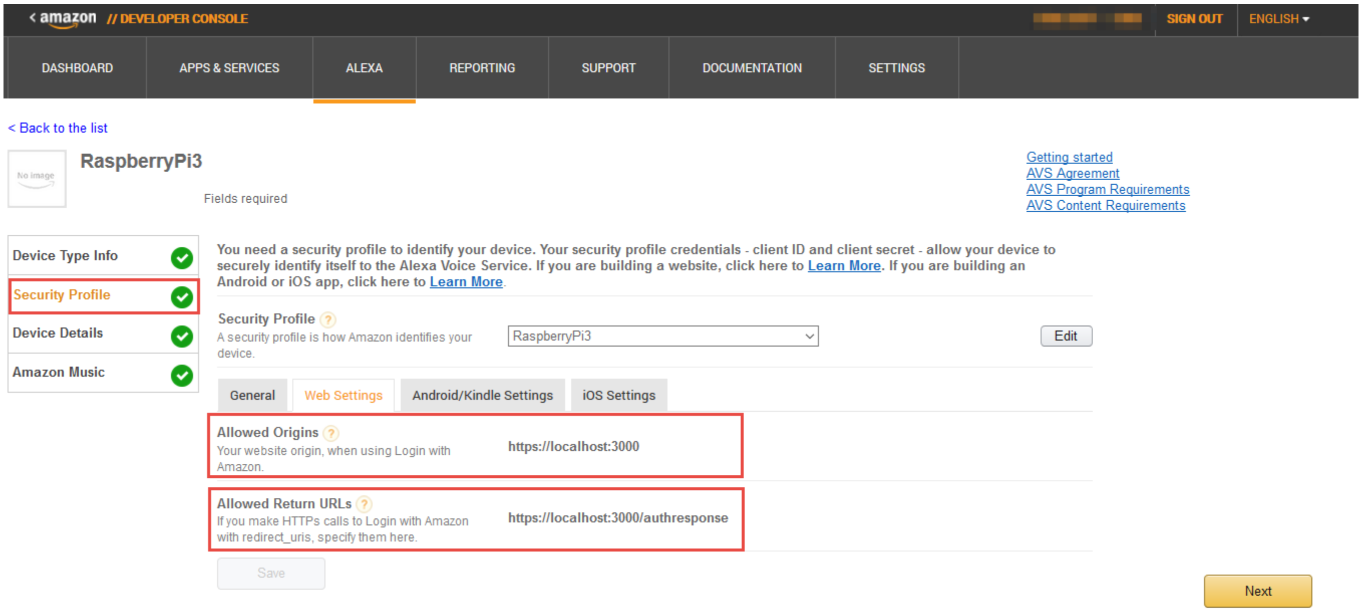
Step 3: Create a device and security profile

Follow the steps here to register your product and create a security profile.

Make note of the following parameters. You'll need these in Step 5 below.

- ProductID (also known as Device Type ID),
- ClientID, and
- ClientSecret

Important: Make sure your **Allowed Origins** and **Allowed Return URLs** are set under **Security Profile > Web Settings** (see Create a device and security profile):



RaspberryPi3

Fields required

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. RaspberryPi3

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

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

- **Allowed Origins:** https://localhost:3000
- **Allowed Return URLs:** https://localhost:3000/authresponse

---

Step 4: Clone the sample app

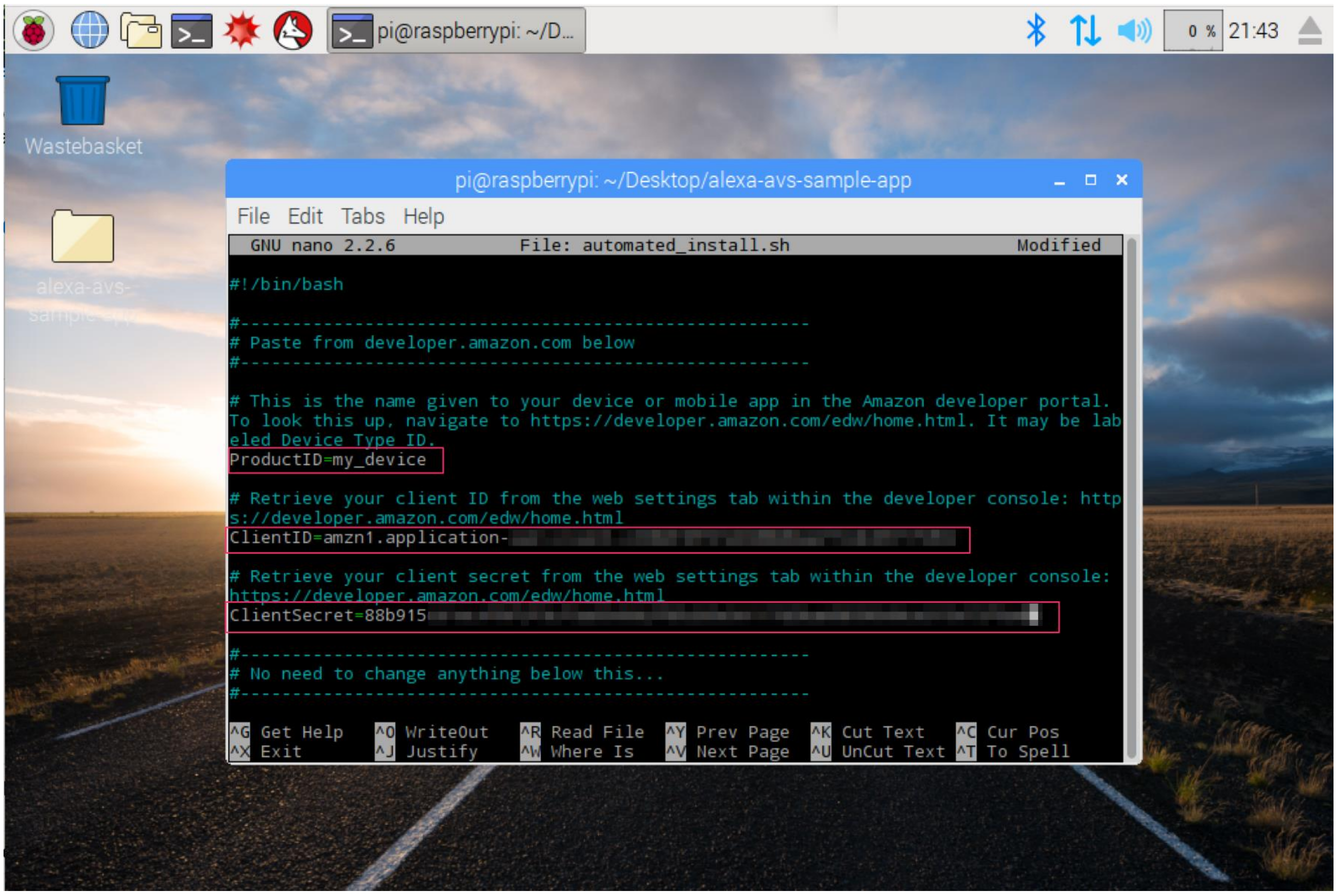
Open **terminal**, and type the following:

```
cd Desktop
git clone https://github.com/alexa/alexa-avs-sample-app.git
```

Step 5: Update the install script with your credentials

Before you run the install script, you need to update the script with the credentials that you got in step 3 - ProductID, ClientID, ClientSecret. Type the following in **terminal**:

```
cd ~/Desktop/alexa-avs-sample-app
nano automated_install.sh
```



```
  GNU nano 2.2.6              File: automated_install.sh                     Modified

#!/bin/bash

#-------------------------------------------------------
# Paste from developer.amazon.com below
#-------------------------------------------------------

# This is the name given to your device or mobile app in the Amazon developer portal.
To look this up, navigate to https://developer.amazon.com/edw/home.html. It may be lab
eled Device Type ID.
ProductID=my_device

# Retrieve your client ID from the web settings tab within the developer console: http
s://developer.amazon.com/edw/home.html
ClientID=amzn1.application-[redacted]

# Retrieve your client secret from the web settings tab within the developer console:
https://developer.amazon.com/edw/home.html
ClientSecret=88b915[redacted]

#-------------------------------------------------------
# No need to change anything below this...
#-------------------------------------------------------

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Paste the values for ProductID, ClientID, and ClientSecret that you got from Step 3 above. The changes should look like this:

- ProductID="RaspberryPi3"
- ClientID="amzn.xxxxx.xxxxxxxxx"
- ClientSecret="4e8cb14xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx6b4f9"

Type **ctrl-X** and then **Y**, and then press Enter to save the changes to the file.

Step 6: Run the install script

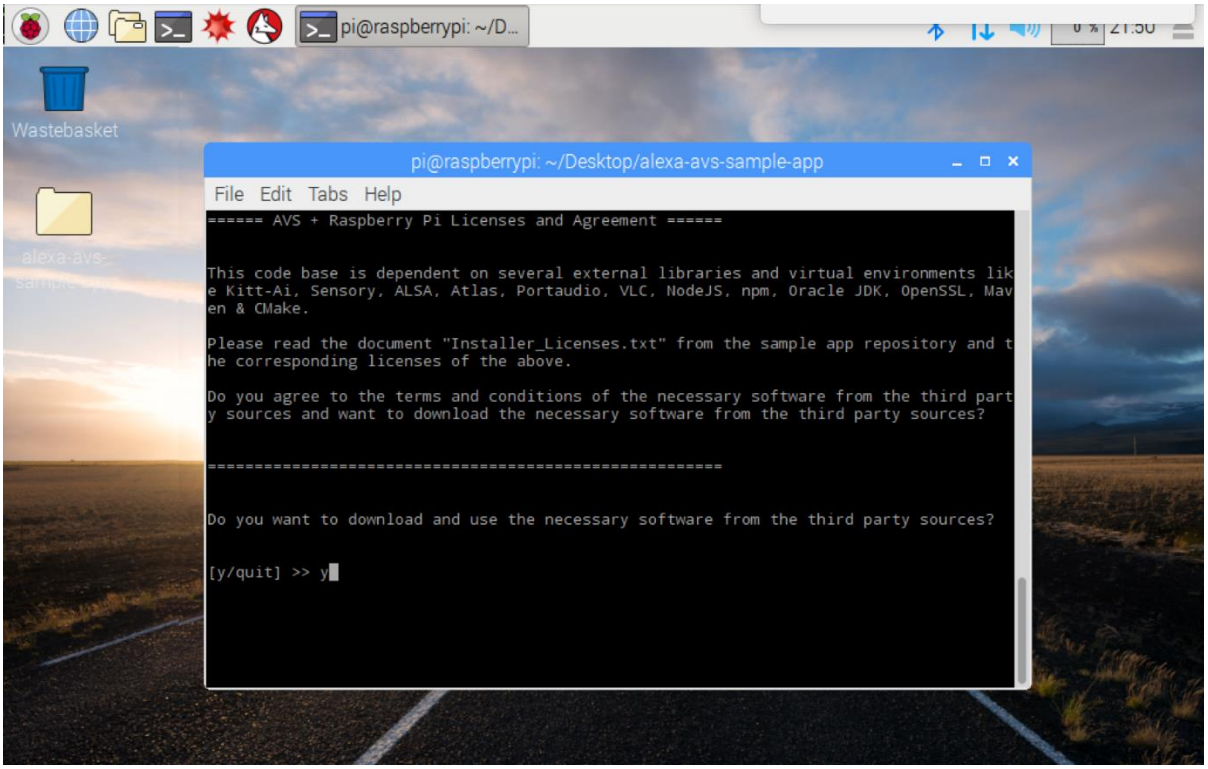
You are now ready to run the install script. This will install all dependencies, including the two wake word engines from Sensory and KITT.AI.

Note: The install script will install all project files in the folder that the script is run from.

To run the script, open **terminal** and navigate to the folder where the project was cloned. Then run the following command:

```
cd ~/Desktop/alexa-avs-sample-app
. automated_install.sh
```

You'll be prompted to answer a few simple questions. These help to ensure that you've completed all necessary prerequisites before continuing.



```
====== AVS + Raspberry Pi Licenses and Agreement ======


This code base is dependent on several external libraries and virtual environments lik
e Kitt-Ai, Sensory, ALSA, Atlas, Portaudio, VLC, NodeJS, npm, Oracle JDK, OpenSSL, Mav
en & CMake.

Please read the document "Installer_Licenses.txt" from the sample app repository and t
he corresponding licenses of the above.

Do you agree to the terms and conditions of the necessary software from the third part
y sources and want to download the necessary software from the third party sources?


=======================================================



Do you want to download and use the necessary software from the third party sources?


[y/quit] >> y
```

When the wizard starts, go grab a cup of coffee - it takes about 30 minutes.

Step 7: Run your web service, sample app and wake word engine

Now that installation is complete, you'll need to run three commands in **3 separate terminal windows**:

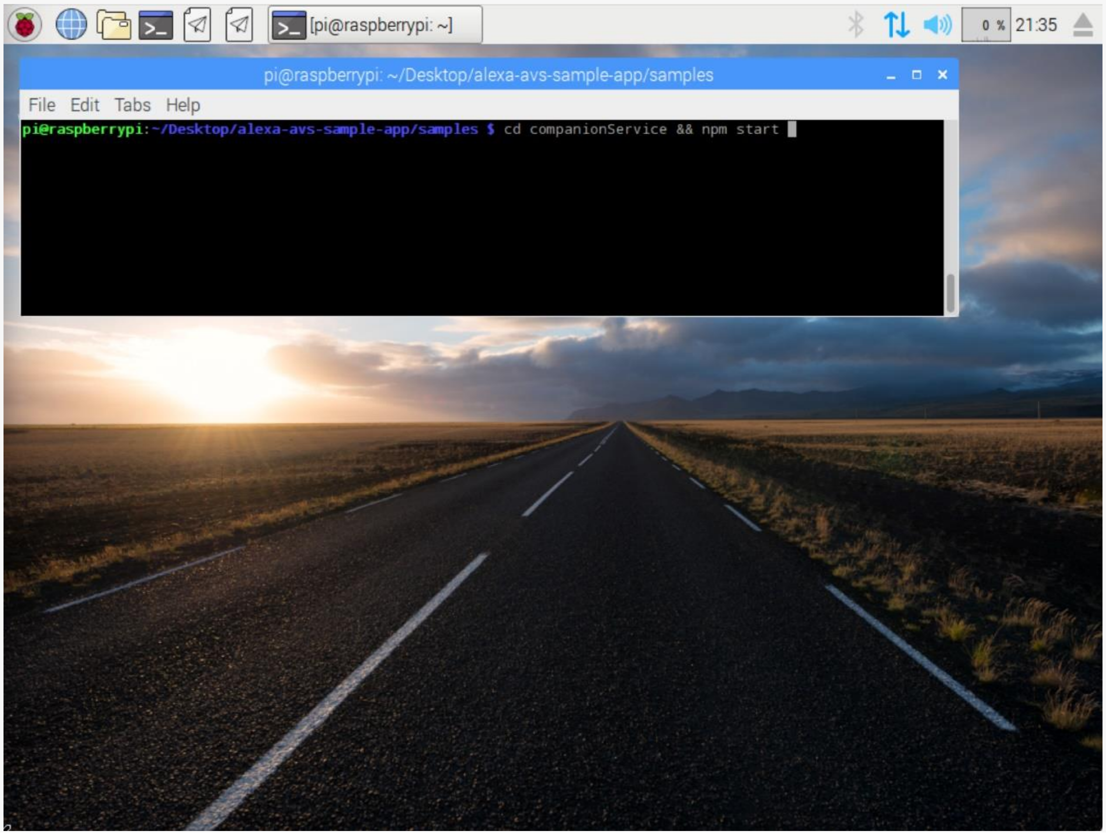
1. **Terminal Window 1**: to run the web service for authorization
2. **Terminal Window 2**: to run the sample app to communicate with AVS
3. **Terminal Window 3**: to run the wake word engine which allows you to start an interaction using the phrase "Alexa".

Note: These commands must be run in order.

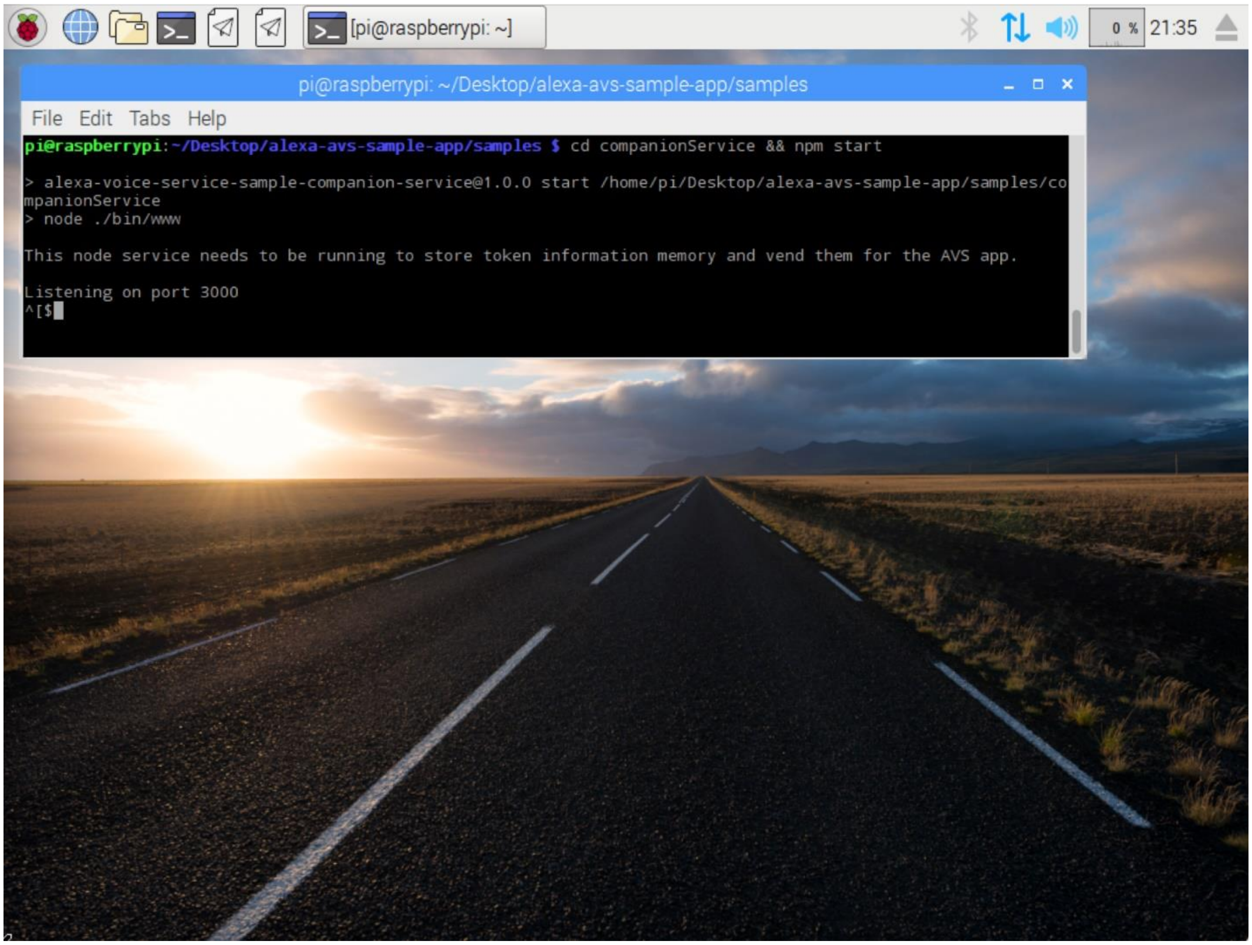
Terminal Window 1

Open a new terminal window and type the following commands to bring up the web service which is used to authorize your sample app with AVS:

```
cd ~/Desktop/alexa-avs-sample-app/samples
cd companionService && npm start
```



```
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd companionService && npm start
```



```
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd companionService && npm start

> alexa-voice-service-sample-companion-service@1.0.0 start /home/pi/Desktop/alexa-avs-sample-app/samples/co
mpanionService
> node ./bin/www

This node service needs to be running to store token information memory and vend them for the AVS app.

Listening on port 3000
^[$
```

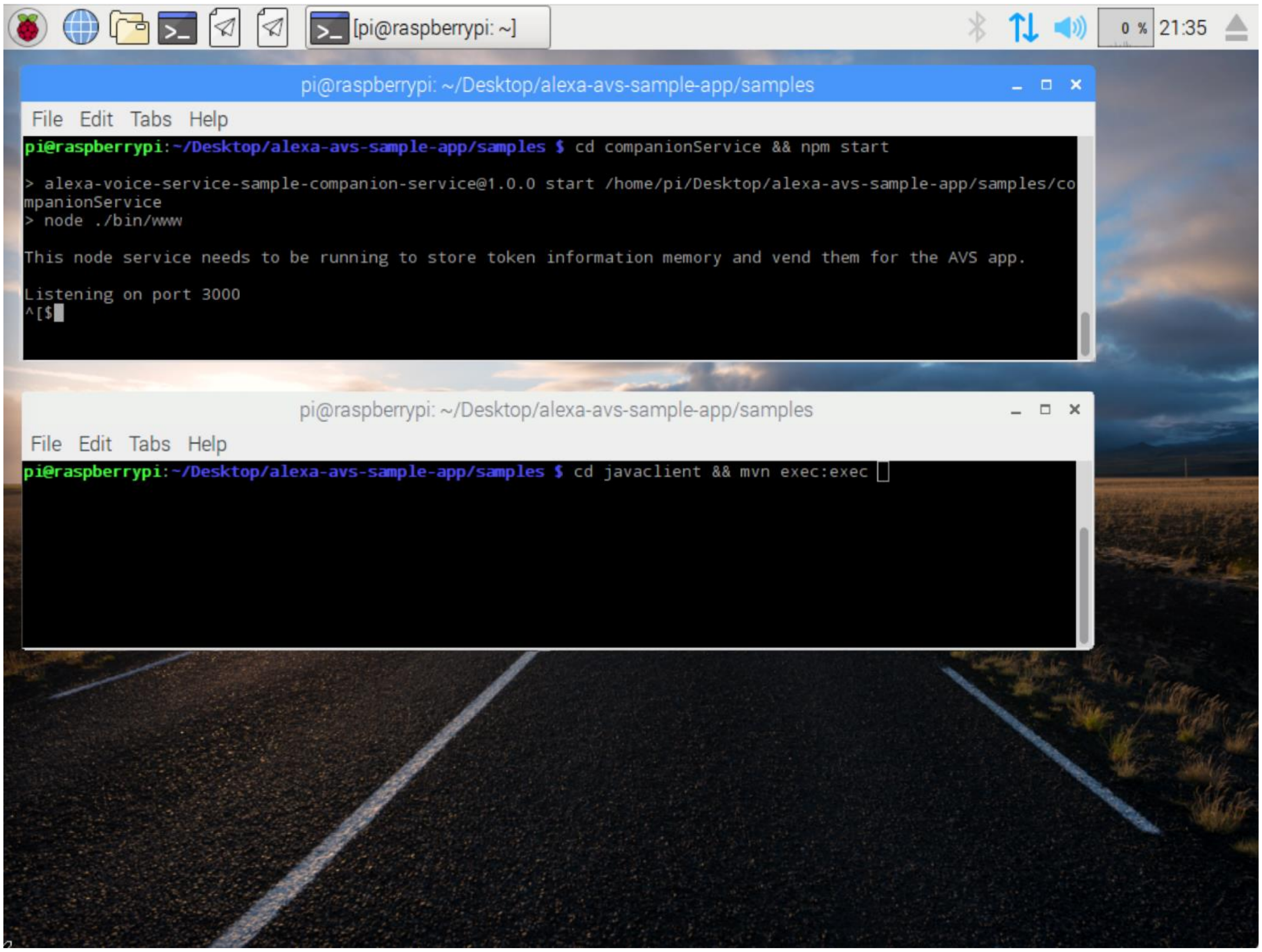
The server is now running on port 3000 and you are ready to start the client.

See API Overview > Authorization to learn more about authorization.

Terminal Window 2

Open a new terminal window and type the following commands to run the sample app, which communicates with AVS:

```
cd ~/Desktop/alexa-avs-sample-app/samples
cd javaclient && mvn exec:exec
```


[pi@raspberrypi: ~]
0 %
21:35
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File Edit Tabs Help

```
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd companionService && npm start

> alexa-voice-service-sample-companion-service@1.0.0 start /home/pi/Desktop/alexa-avs-sample-app/samples/co
mpanionService
> node ./bin/www

This node service needs to be running to store token information memory and vend them for the AVS app.

Listening on port 3000
^[$
```

pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File Edit Tabs Help

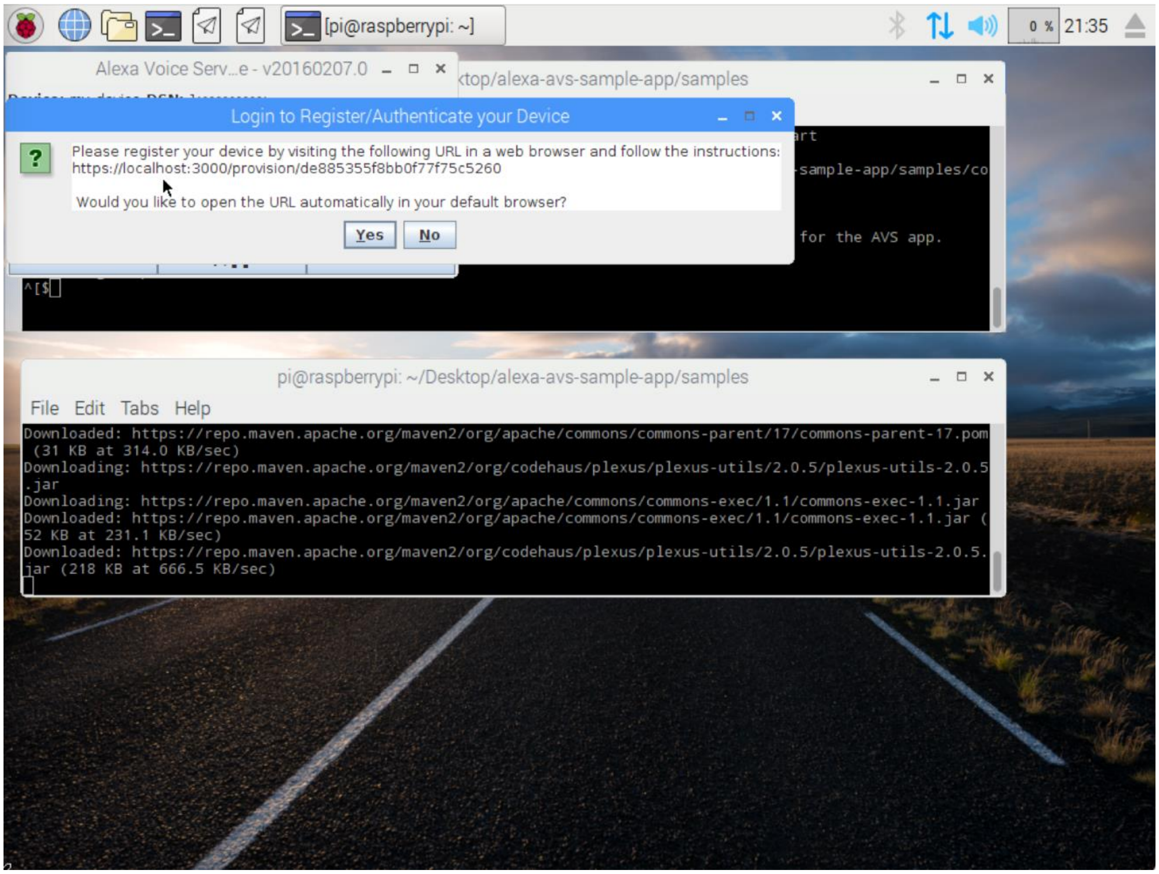
```
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd javaclient && mvn exec:exec
```

See API Overview > Interfaces to learn more about the messages sent to/from AVS.

Let's walk through the next few steps relevant to Window 2.

1. When you run the client, a window should pop up with a message that says -

   Please register your device by visiting the following URL in a web browser and following the instructions: https://localhost:3000/provision/d340f629bd685deeff28a917. Would you like to open the URL automatically in your default browser?


[pi@raspberrypi: ~]
21:35
Alexa Voice Serv...e - v20160207.0
ktop/alexa-avs-sample-app/samples
Login to Register/Authenticate your Device
Yes
No
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File
Edit
Tabs
Help

Please register your device by visiting the following URL in a web browser and follow the instructions:
https://localhost:3000/provision/de885355f8bb0f77f75c5260

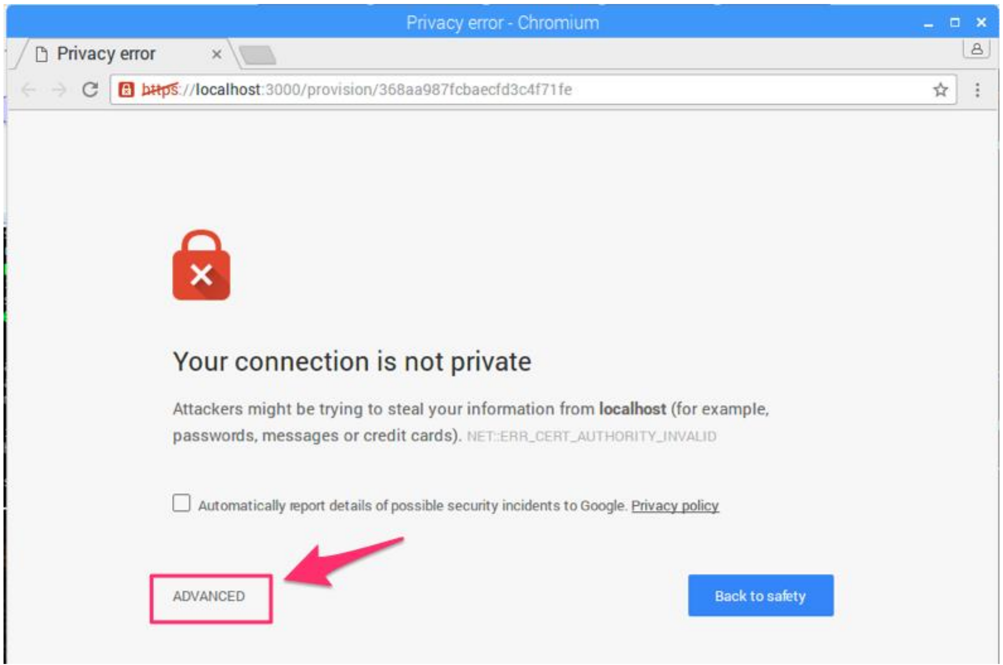
Would you like to open the URL automatically in your default browser?

```
art
-sample-app/samples/co
 for the AVS app.
^[$
```

```
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/17/commons-parent-17.pom
 (31 KB at 314.0 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5
.jar
Downloading: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar (
52 KB at 231.1 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5.
jar (218 KB at 666.5 KB/sec)
```

Click on **"Yes"** to open the URL in your default browser.

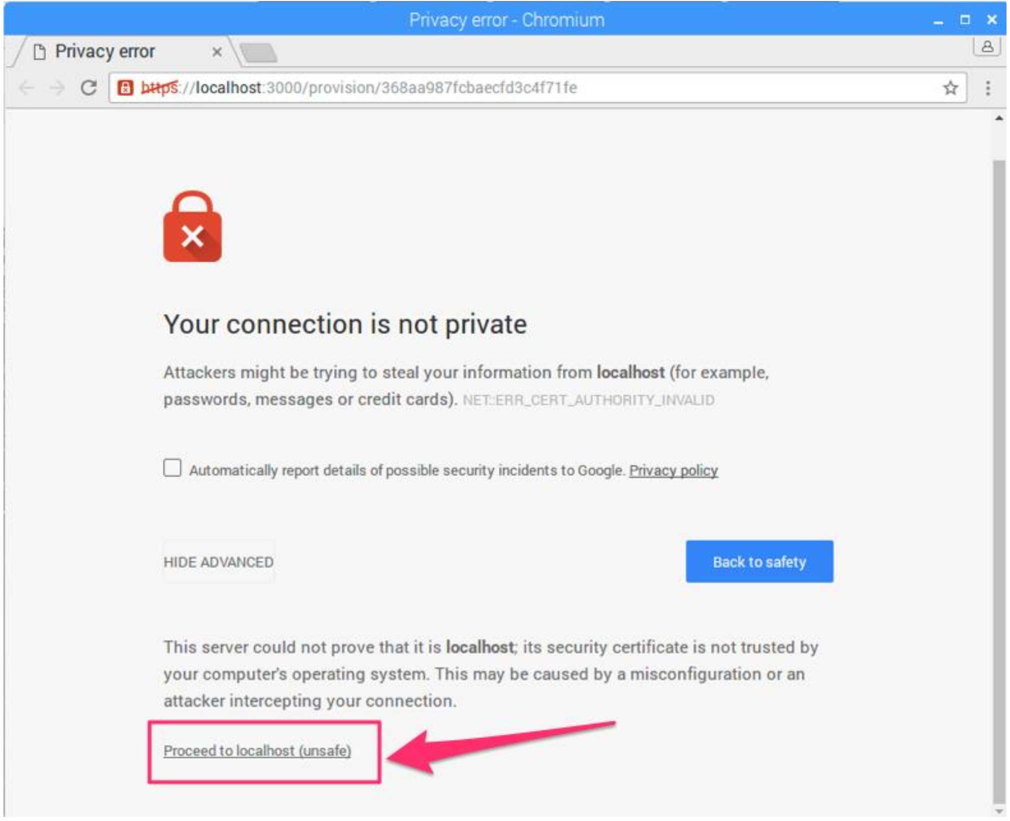
2. If you're running Raspbian Jessie with Pixel desktop (and with Chromium browser), you may get a warning from the browser. You can get around it by clicking on **Advanced -> Proceed to localhost(unsafe)**.


Privacy error - Chromium
Privacy error
~~https~~://localhost:3000/provision/368aa987fcbaecfd3c4f71fe
ADVANCED
Back to safety

Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

□ Automatically report details of possible security incidents to Google. Privacy policy


Privacy error - Chromium
Privacy error
~~https~~://localhost:3000/provision/368aa987fcbaecfd3c4f71fe
HIDE ADVANCED
Back to safety
Proceed to localhost (unsafe)

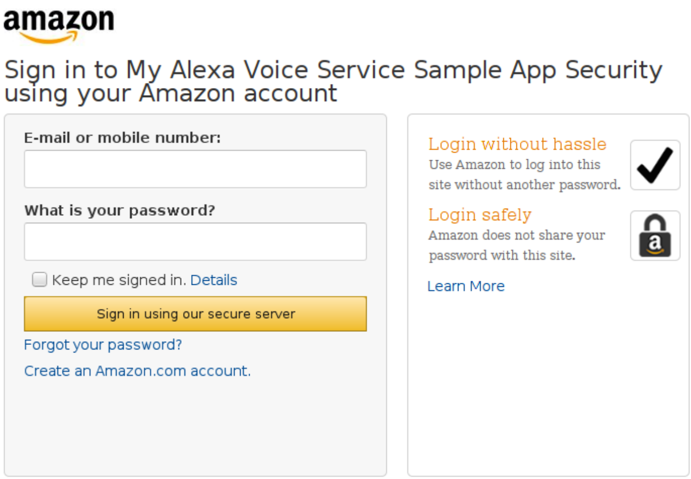
Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

□ Automatically report details of possible security incidents to Google. Privacy policy

This server could not prove that it is **localhost**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

3. You'll 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

Forgot your password?

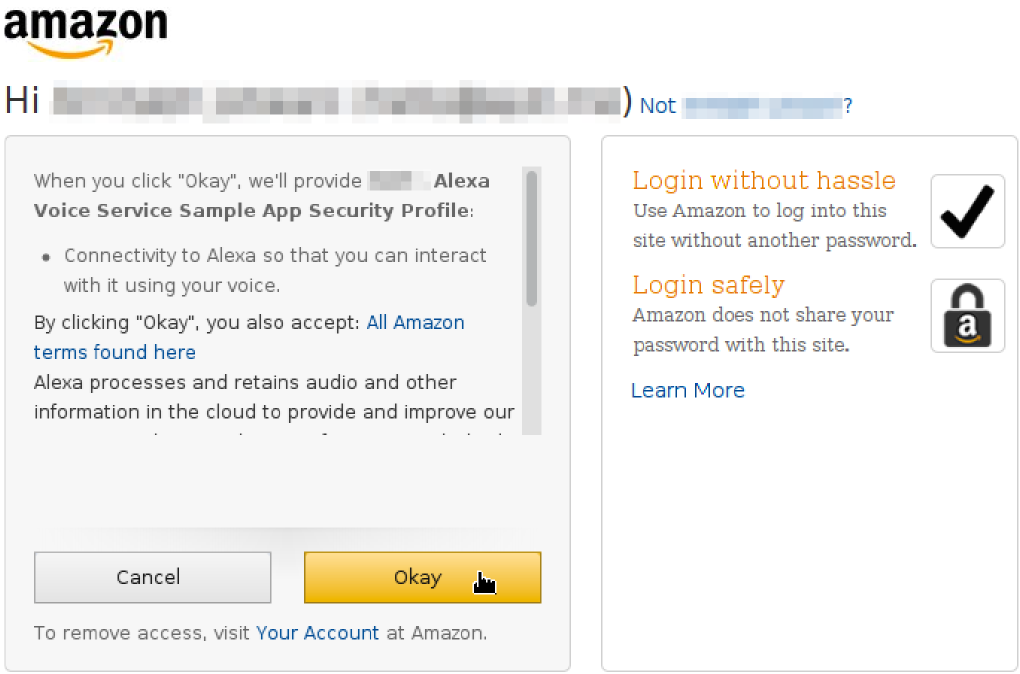
Create an Amazon.com account.

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

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

Learn More

4. You'll be taken to a Dev Authorization page, confirming that you'd like your device to access the Security Profile created earlier.



amazon

Hi [redacted]) Not [redacted]?

When you click "Okay", we'll provide [redacted] **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

To remove access, visit Your Account at Amazon.

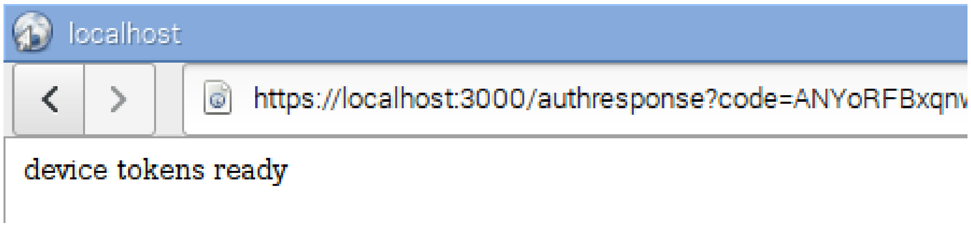
Login without hassle
Use Amazon to log into this site without another password.

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

Learn More

Click **Okay**.

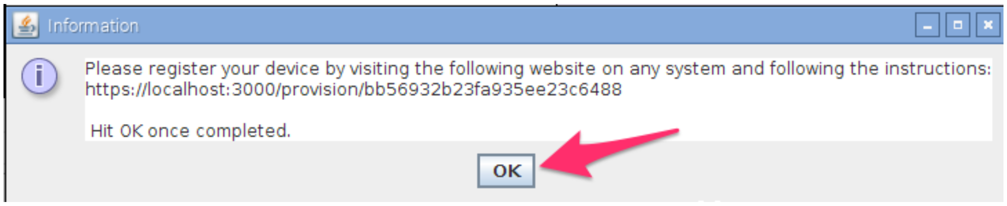
5. You will now be redirected to a URL beginning with https://localhost:3000/authresponsefollowed by a query string. The body of the web page will say device tokens ready.



```
https://localhost:3000/authresponse?code=ANYoRFBxqnw
```

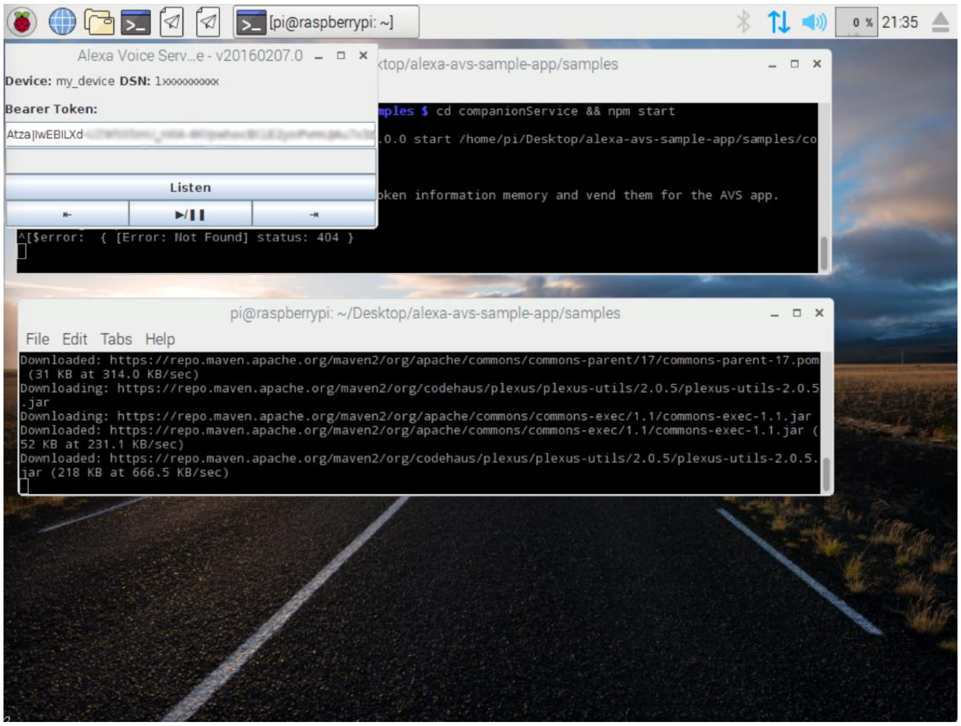
device tokens ready

6. 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.



Device: my_device DSN: 1xxxxxxxxxx

Bearer Token:

Atza|IwEBILXd[redacted]

```
ktop/alexa-avs-sample-app/samples

mples $ cd companionService && npm start
.0.0 start /home/pi/Desktop/alexa-avs-sample-app/samples/co

oken information memory and vend them for the AVS app.

^[$error:  { [Error: Not Found] status: 404 }
```

```
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples

Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/17/commons-parent-17.pom
 (31 KB at 314.0 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5
.jar
Downloading: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar (
52 KB at 231.1 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5.
jar (218 KB at 666.5 KB/sec)
```

Terminal Window 3

Note: *Skip this step to run the same app without a wake word engine*.

This project supports two third-party wake word engines: Sensory's TrulyHandsFree and KITT.AI's Snowboy. The -e parameter is used to select the agent and supports two values for {{engine}}: kitt_ai and sensory.

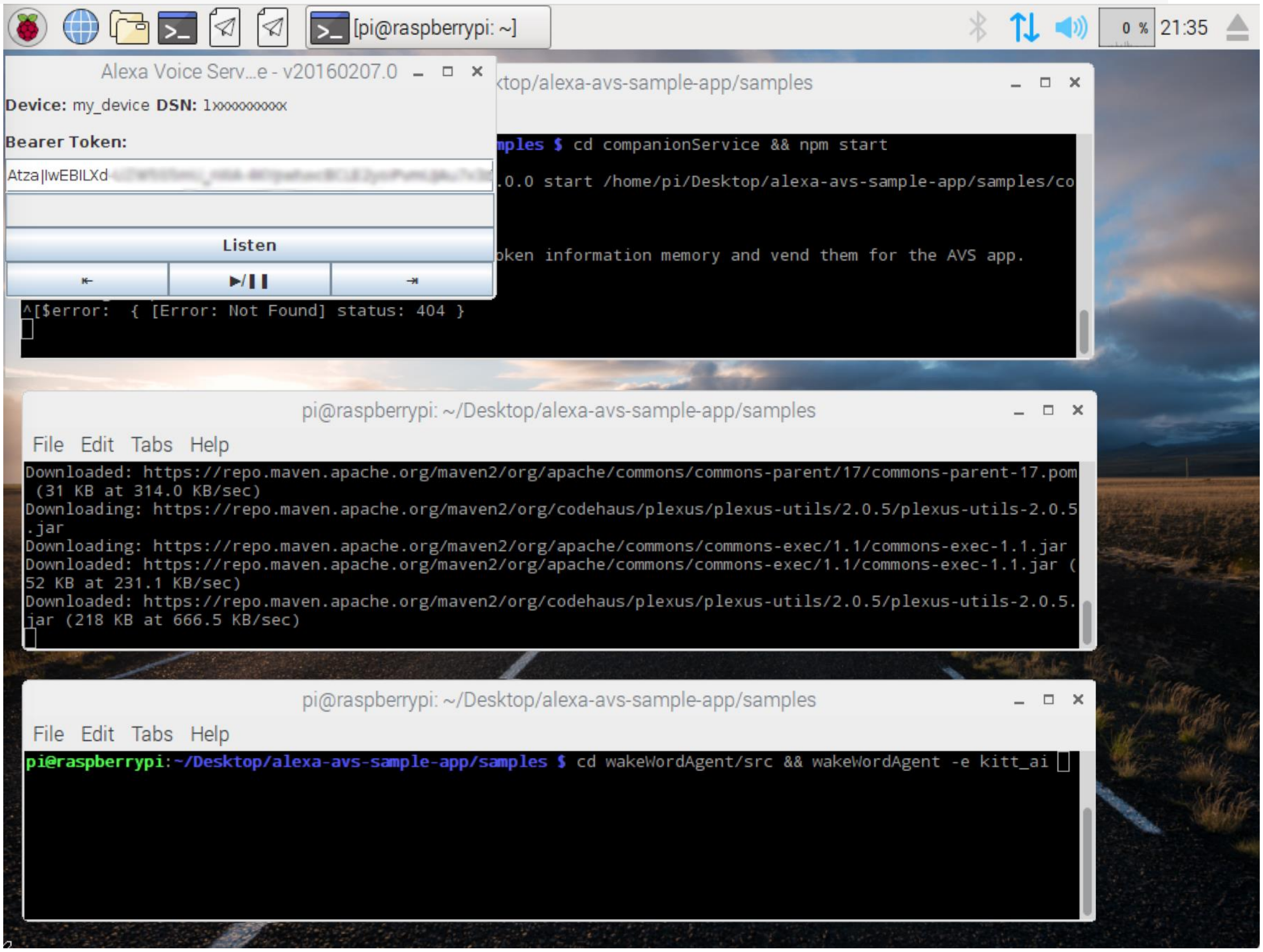
Open a new terminal window and use the following commands to bring up a wake word engine from Sensory or KITT.AI. The wake word engine will allow you to initiate interactions using the phrase "Alexa".

To use the **Sensory** wake word engine, type -

```
cd ~/Desktop/alexa-avs-sample-app/samples
cd wakeWordAgent/src && ./wakeWordAgent -e sensory
```

or, type this to use **KITT.AI**'s wake word engine -

```
cd ~/Desktop/alexa-avs-sample-app/samples
cd wakeWordAgent/src && ./wakeWordAgent -e kitt_ai
```



Device: my_device **DSN:** 1xxxxxxxxxx

Bearer Token:

Atza|IwEBILXd[redacted]

```
ktop/alexa-avs-sample-app/samples
mples $ cd companionService && npm start
.0.0 start /home/pi/Desktop/alexa-avs-sample-app/samples/co
oken information memory and vend them for the AVS app.
^[$error:  { [Error: Not Found] status: 404 }
```

pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples

```
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/17/commons-parent-17.pom
 (31 KB at 314.0 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5
.jar
Downloading: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-exec/1.1/commons-exec-1.1.jar (
52 KB at 231.1 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.5/plexus-utils-2.0.5.
jar (218 KB at 666.5 KB/sec)
```

pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples

```
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd wakeWordAgent/src && wakeWordAgent -e kitt_ai
```

Now you have a working hands-free AVS prototype!

Use the following resources to learn more about available wake word engines:

- Sensory

- KITT.AI

Step 8: Talk to Alexa

You can now talk to Alexa by simply using the wake word "Alexa". Try the following -

Say "Alexa", then wait for the beep. Now say "what's the time?"

Say "Alexa", then wait for the beep. Now say "what's the weather in Seattle?"

If you prefer, you can also click on the "Listen" button, instead of using the wake word. Click the "Listen" button and wait for the audio cue before beginning to speak. It may take a second or two before you hear the audio cue.