



Getting Started

Set up your Amazon accounts

We highly recommend you complete the [Alexa Fact Skill tutorial](#), which steps you through setting up your Amazon Developer and Amazon AWS accounts as well as building the project this one is based on. If you've already done that, your accounts are ready to go.

- Navigate to the [Amazon Developer Portal](#) and create an Amazon account if you don't have one already.
- Navigate to [Amazon AWS](#) and create an account if you do not already have one. You will need to provide a credit card to set up the account, but new accounts receive the [AWS Free Tier](#) which should suffice for this project as it includes one million non-expiring AWS Lambda requests per month. Note: Lambda functions for Alexa skills can be hosted in either the US East (N. Virginia) or EU (Ireland) region. These are the only regions the Alexa Skills Kit supports, so you may need to change this setting in the upper right portion of your screen on the console.

Starter Code

Install your starter code locally.

- Download or clone the [starter code](#) from GitHub
- Save it in a directory named `AIND-VUI-Alexa`. It contains the following directories and files:
 - **speechAssets/IntentSchema.json** - intents definition for the interactive model
 - **speechAssets/SampleUtterances_en_US.txt** - utterances for the interactive model
 - **src/index.js** - skill logic and handlers to be run in AWS Lambda
 - **src/facts.js** - a list of facts that the skill will use in responses
 - **tests/*.js** - various unit tests to be run locally with mocha; you do not need to change these



Getting Started

2. Install dependencies for the project

- Navigate to the `AIND-VUI-Alexa/src` directory of the starter code and open a terminal window.
- The dependencies we need are listed in the `package.json` file and include the `alexa-sdk` library for Alexa as well as `mocha` test framework for Node.js along with `chai` and `aws-lambda-mock-context` for local unit testing. Install them all with the following command:

```
$ npm install
```

- There should now be a directory named `node_modules` within the `src` directory. This is how Node.js attaches libraries for your code.

3. Unit testing

- You can now run the provided unit tests from the command line within the `src` directory with the following command. Try it now:

```
$ npm test
```

The test code is in four parts: "Part 1", "Part 2", "Part 3", and "Starter Code". The starter code tests should pass and all others should fail. You may have to scroll up to see the passing tests. As you complete the tasks ahead, you can use these unit tests (and write more yourself) to quickly test code changes prior to deployment to AWS Lambda.

4. JavaScript coding

- At this point, you can open your project with a code editor of your choice. Some free ones that support JavaScript and Node.js include `Atom` and `Visual Studio Code`
- Udacity has a free course available that will quickly bring you up to speed on JS syntax:

[Introduction to Javascript](#)



Getting Started



The JavaScript skills needed for this project primarily require following patterns you find in the existing starter code and the use of general coding constructs such as for loops, conditional statements, and arrays. Here are a few links that might be helpful for quick reference:

- [js for loops](#)
- [js conditional statements](#)
- [js arrays](#)
- [js String includes\(\) method](#)
- [js JSON](#)
- [Alexa skill examples](#)
- [Node.js API reference](#)

5. Run the Starter Code on AWS Lambda

- Deploy the starter code to verify that it works with your accounts in its simple form. This is the same process you went through with the Space Geek Lab. If you need a refresher, step-by-step instructions are provided [here](#).

[NEXT](#)