

**Lab: Build an Alexa History Skill**

It's time to create your own Alexa Skill! In this lab, you’ll build a fully functional skill for [Amazon’s Alexa](https://developer.amazon.com/alexa)that provides year-dated facts from AI History (or some other History of your choosing). Through the process, you’ll get to use the [Alexa Skills Kit (ASK)](https://developer.amazon.com/alexa-skills-kit) - a current state of the art API for building voice systems.

But before delving into the lab, it's best for you to go over the [Alexa Fact Skill tutorial](https://github.com/alexa/skill-sample-nodejs-fact) that the Alexa team provides as a template for building Alexa Skills. Do that now, and come back to continue this lab.

**Done with the Alexa Fact Skill tutorial?**

Great! In this lab, we'll build on top of that template to create a slightly more advanced Alexa Skill -- a History skill that, in addition to giving the user a random history fact, is also able to take a specific date from the user and give a fact from that year.

You are free to choose any category you wish for your facts. For example, you could have a skill that provides historical facts about a sport, or a location, or hobby, but each fact you provide must include a **four-digit year**.

If you are not sure what category you want to try, don't worry...an example file has been provided with a few historical facts about artificial intelligence to get you started. It can be hard to come up with a robust list of suitable facts - feel free to crowdsource a list of historical facts with your fellow students!

# Getting Started

Now that you've completed the [Alexa Fact Skill tutorial](https://github.com/alexa/skill-sample-nodejs-fact), we can start talking about the History Skill. We are providing you with [starter code](https://github.com/udacity/NLPND-VUI-Alexa) from GitHub with some unit tests to help guide your progress. The next concept has some tasks that you need to complete to build the history skill. You can run the unit tests as you complete the tasks to see a reflection of your progress.

### Your Amazon accounts

By completing the Fact skill tutorial, you should have already completed setting up your [Amazon Alexa Developer account](https://developer.amazon.com/alexa/) and [Amazon AWS account](https://aws.amazon.com/).

If you do not already have an AWS account, be sure to create one. You will need to provide a credit card to set up the account, but new accounts receive the [AWS Free Tier](https://aws.amazon.com/free/) 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](https://github.com/udacity/NLPND-VUI-Alexa) from GitHub
* Save it in a directory named NLPND-VUI-Alexa. It contains the following directories and files:
  + **src/model/en-US.json** - This is the main Alexa file. It contains the components that Alexa needs to build the "Interaction Model" for your skill. It used to declare "intents", which are ways to handle user requests. The file is also used to give Alexa samples for what a user might say to trigger a certain intent (for example, "tell me a history fact").
  + **src/lambda/index.js** - This is the main Lambda file. It will contain the code that is tasked with finding the right response to give to the user -- initially picking a random fact from a list, but later searching a list for facts for a certain year.
  + **tests/\*.js** - various unit tests to be run locally with mocha; you do not need to change these

After completing the tasks locally, we'll just copy the contents of the first two files in the list above and paste them in specific places in the Alexa and AWS consoles. You can think of the Alexa code as front-end code guiding the user's interaction with Alexa, and, the lambda code as the back-end that gets a programmatic request and returns an appropriate response.

### Environment

##### 1. Install [Node.js](https://nodejs.org/) per instructions on the website for your machine.

##### 2. Install dependencies for the project

* Navigate to the NLPND-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](https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs" \t "_blank) library for Alexa as well as  [mocha](https://mochajs.org/) test framework for Node.js along with [chai](http://chaijs.com/) and [aws-lambda-mock-context](https://www.npmjs.com/package/aws-lambda-mock-context" \t "_blank) 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 two parts: "Part 1", and "Part 2". Initially, all tests will fail. 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](https://atom.io/) and [Visual Studio Code](https://code.visualstudio.com/)
* Udacity has a free course available that will quickly bring you up to speed on JS syntax:

### [Introduction to Javascript](https://www.udacity.com/course/intro-to-javascript--ud803)

[](https://www.udacity.com/course/intro-to-javascript--ud803)  
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](https://www.w3schools.com/js/js_loop_for.asp)
  + [js conditional statements](https://www.w3schools.com/js/js_if_else.asp)
  + [js arrays](https://www.w3schools.com/js/js_arrays.asp)
  + [js String includes() method](https://www.w3schools.com/jsref/jsref_includes.asp)
  + [js JSON](https://www.w3schools.com/js/js_json.asp)
  + [Alexa skill examples](https://github.com/alexa)
  + [Node.js API reference](https://nodejs.org/dist/latest-v6.x/docs/api/)

# Tasks

The lab consists of two parts:

1. Expand history fact skills set
2. Add a feature by creating a new intent with built-in slot

## Part 1: Expand history facts skills

#### Tasks:

1. Choose a history category you wish to use for your skill. We're going to use Alexa sample skill template from [History Facts](https://github.com/alexa/skill-sample-nodejs-fact) GitHub repo.
2. Expand the sample utterances in the [starter code](https://github.com/udacity/NLPND-VUI-Alexa) src/models/en-US.json file to include at least 15 appropriate samples/utterances for GetNewFactIntent. Examples can be found [here](https://github.com/alexa/skill-sample-nodejs-fact/blob/en-US/models/en-US.json).
3. Expand the facts list at the top of the src/lambda/index.js file such that there are at least 10 distinct facts, where each includes a 4-digit year in its text. These will be spoken by the Amazon Text-To-Speech algorithm (TTS), so keep in mind where you wish pauses to occur. To hear how it sounds, enter your sentence in the developer portal under the “Test” section.
4. Test it! All “Part 1” local unit tests should pass. In the next lesson, you'll try the skill out by uploading your changes to the Interactive model in the Developer Portal and AWS Lambda.

## Part 2: Add a feature by creating a new intent with built-in slot

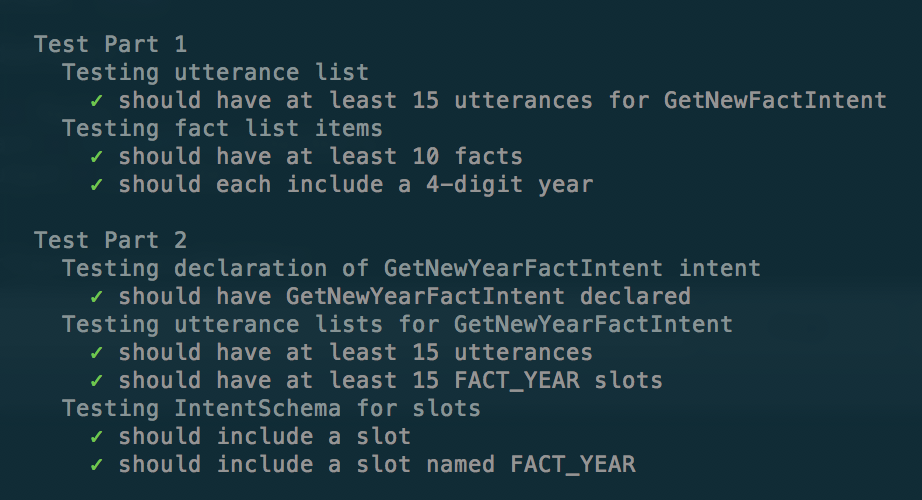
In addition to the GetNewFactIntent intent already included in Part 1, add a new [intent](https://developer.amazon.com/docs/custom-skills/create-the-interaction-model-for-your-skill.html#intents-and-slots) including a [built-in slot](https://developer.amazon.com/docs/custom-skills/slot-type-reference.html) that will provide the user with a fact matching the year requested. You can name this new intent GetNewYearFactIntent and build-in slot named FACT\_YEAR.

Built-in Amazon slots can be used for the fact year. Consider using[*AMAZON.FOUR\_DIGIT\_NUMBER*](https://developer.amazon.com/docs/custom-skills/slot-type-reference.html#four_digit_number)for this purpose. This is not required, however, if you prefer to try a different slot definition.

#### Tasks:

1. Provide at least 15 utterances for the new GetNewYearFactIntent intent.
2. Complete the TODO’s in src/lambda/index.js file to implement a [Lambda function handler](https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-handler.html)for GetNewYearFactIntent that will provide a fact for the specific year requested and provide a random fact if the year is not found in the fact list. You can use*[GetNewFactHandler](https://github.com/udacity/NLPND-VUI-Alexa/blob/master/src/lambda/index.js" \t "_blank)*as a template.
3. Test it! All “Part 2” local unit tests should pass.

From the terminal, run npm test inside the src directory to check your passing tests.

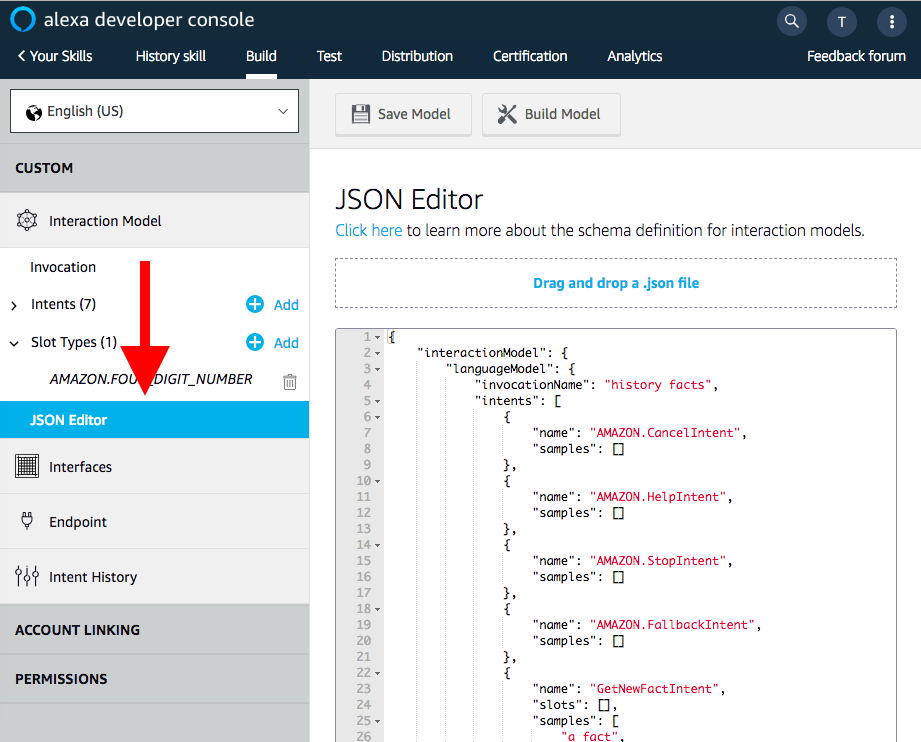
[[](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)

# Deploying your Skill

The Fact Skill [tutorial](https://github.com/alexa/skill-sample-nodejs-fact) steps you through setting up an Alexa Skill. The starter code can be deployed in the same way. The following step-by-step instructions assume you already have completed the steps in the **Getting Started** concept, and are ready to deploy your starter code. Open both [Alexa Developer](https://developer.amazon.com/alexa/)portal and the [Amazon AWS](https://aws.amazon.com/) console to complete the tasks.

#### Tasks inside Alexa Developer portal:

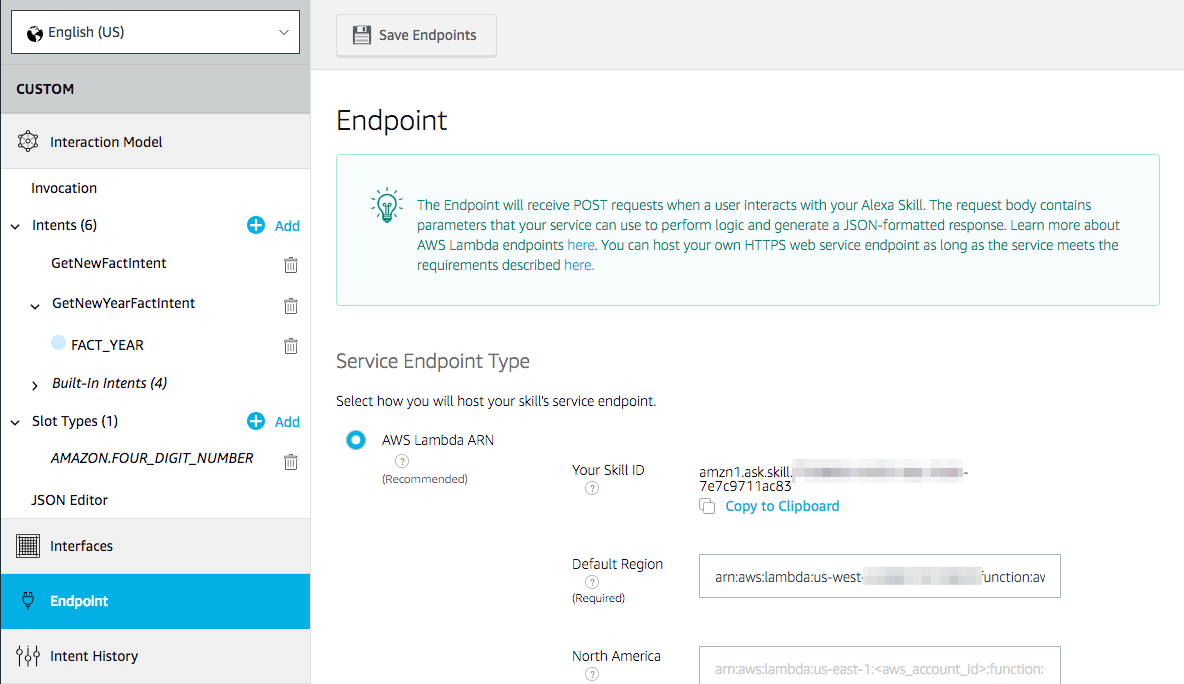
1. On Alexa Developer portal, click Create Skill, enter Skill name, and choose Custommodel and Fact Skill template.
2. Navigate to the Build tab inside the skill Interaction Model.
3. From the Interaction Model menu on the left, select JSON Editor.
4. Paste or drag-and-drop the contents of your src/model/en-US.json file in the text box on that page.
5. Click Save Model at the top of the page.

[[](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)

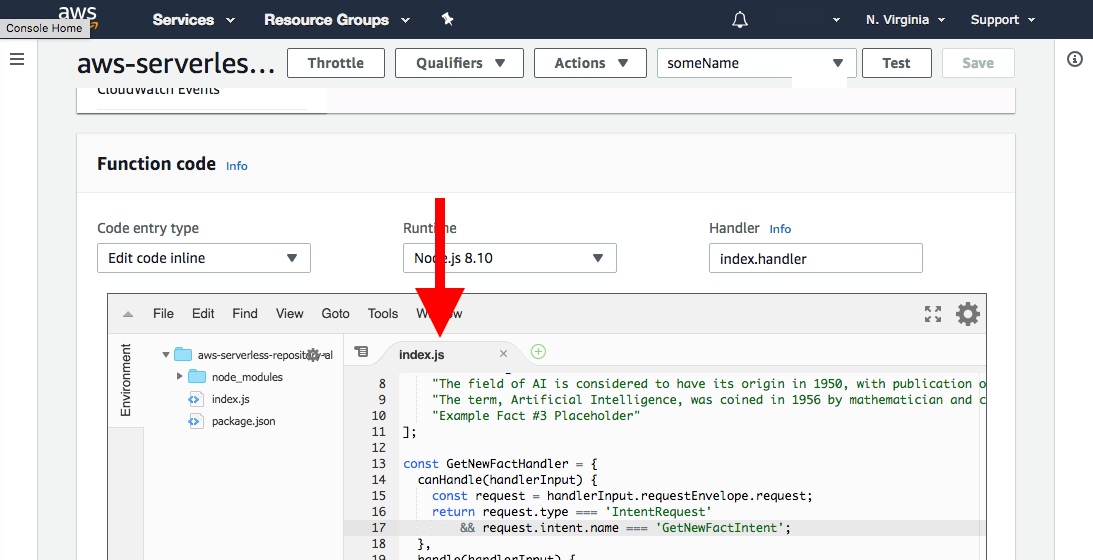
#### Tasks inside AWS console:

1. Navigate to AWS console and choose the [AWS Lambda](https://aws.amazon.com/lambda/) service.
2. Create and set up the Lambda function using Serverless Application Repository and NodeJS template by following [step 2](https://github.com/alexa/skill-sample-nodejs-fact/blob/en-US/instructions/2-lambda-function.md) of Alexa sample skills set instructions.
3. Time to connect AWS Lambda service and Alexa Developer VUI console! Copy AWS Lambda ARNinto VUI Endpoint. See [step 3](https://github.com/alexa/skill-sample-nodejs-fact/blob/en-US/instructions/3-connect-vui-to-code.md) detail instructions.
4. Copy the contents of your src/lambda/index.js into the index.js file in the editor under the **Function Code** header.

Copy AWS Lambda ARN into VUI Endpoint to connect Lambda service and VUI console:

[[](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)

Copy*index.js*script into AWS Lambda Function Code:

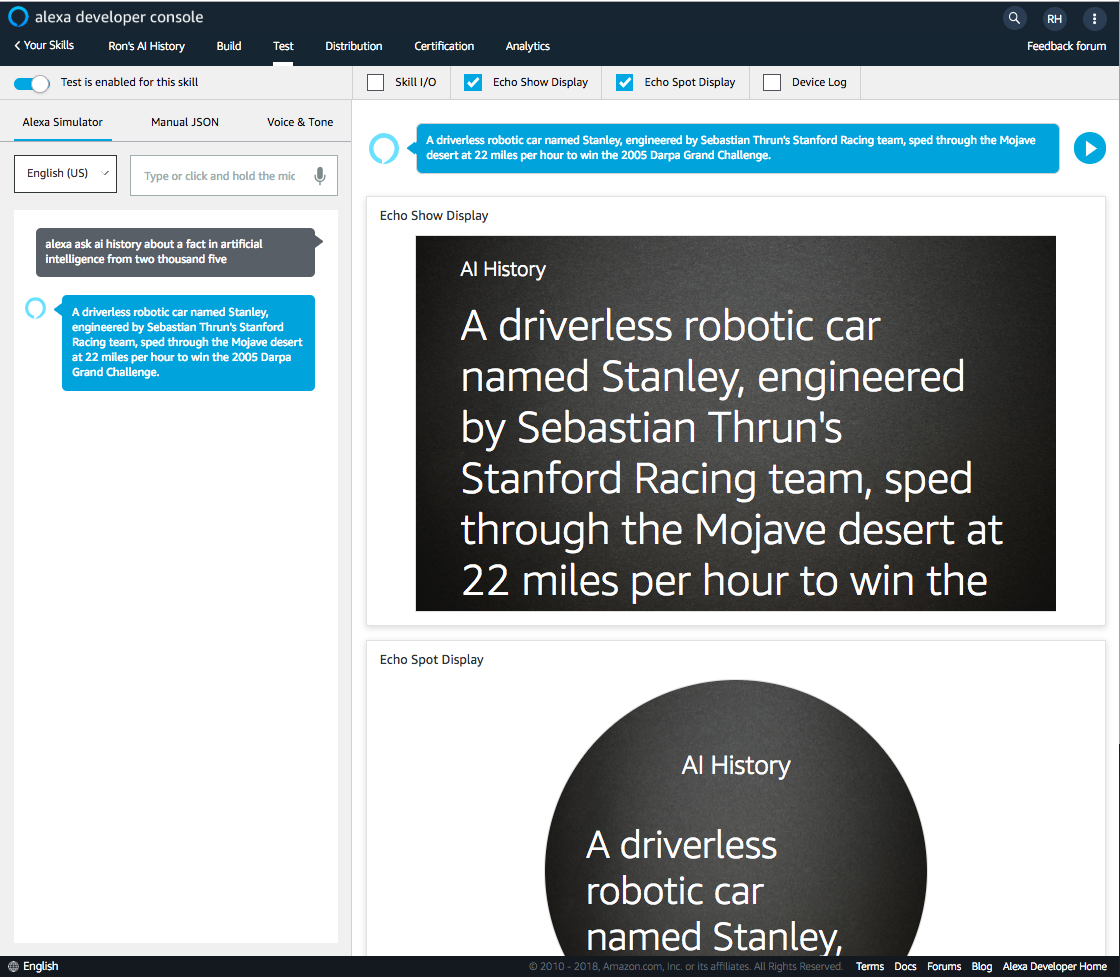
[[](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)

#### Test your Alexa skill deployment

Once your Lambda function is deployed and ready, and you have successfully built the skill in the Alexa Developer Console, you can proceed to the Test tab in the Alexa Developer Console to test your skill.

Notes on testing:

* You might have to enable testing your skill by toggling the button on the top left
* You can use the Alexa simulator to utterances by voice or by typing text into the text box.
* With more advanced utterances (especially those including slots), you might have to run more explicit tests from the Manual JSON tab.
* You can monitor the logs in [AWS Cloudwatch](https://console.aws.amazon.com/cloudwatch) to check any error messages logged by the Lambda function. You can also see the output of any log messages your code creates (using console.log('')).
* More at [Test Your Skill](https://developer.amazon.com/docs/devconsole/test-your-skill.html)

[[](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)](https://classroom.udacity.com/nanodegrees/nd892/parts/94230570-1456-442c-a152-ca4e171acbfc)