



Tasks

The starter code provides a simple fact skill similar to Space Geek, except a few AI history facts have been provided in an external file, `facts.js`, instead of the space facts. Each fact includes a 4-digit year in its text, which we will use in the project for a new feature.

The project consists of three parts:

1. Customize the fact skill
2. Add a feature using an additional intent and a slot
3. Add conversational elements

Part 1: Customize the fact skill

1. Choose a history category you wish to use for your skill. You can continue to use the AI History Facts already started for you if you wish.
2. Expand the utterances in the `speechAssets/SampleUtterances_en_US.txt` file to include at least 15 appropriate utterances for `GetNewFactIntent`. Examples can be found [here](#).
3. Expand the facts list in `facts.js` 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:

English (U.S.) Add a New Language

Skill Information

Interaction Model

Configuration

Test

Publishing Information

Privacy & Compliance

Skills Beta Testing NEW

Status: Not yet eligible

Please complete the Interaction Model tab to start testing this skill.

Enabled This skill is enabled for testing on your account.

Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

You will be able to see your skill in the Skills tab in Alexa App and you can enable the skill and start testing.

After completing your testing please submit the skill for certification. If it passes Amazon's testing and certification process, it will become available to Alexa end users

The skill is available in "Skills > Your Skills" page of the Alexa App when you select "Yes" above. You can then enable the skill and test its functionality by asking Alexa, ask arty

Voice Simulator

Hear how Alexa will speak a response entered in plain text or SSML. [Learn more about supported SSML tags.](#)

For example: Here is a word spelled out: `<say-as interpret-as="spell-out">hello</say-as>`.

hello world

Listen



Part 2: Add a feature

In addition to the `GetNewFactIntent` intent already included, add an `intent` including a built-in `slot` named `FACT_YEAR` that will provide the user with a fact matching the year requested. Name this intent `GetNewYearFactIntent`. Built-in Amazon slots can be used for the year. Consider using `AMAZON.FOUR_DIGIT_NUMBER` for this purpose. This is not required, however, if you prefer to try a different slot definition.

1. Provide at least 15 utterances for the new intent.
2. Complete the TODO's in `index.js` to implement a handler 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.
3. Test it. All "Starter Code", "Part 1", and "Part 2" local unit tests should pass. Try the skill out by uploading your changes to the Interactive model in the Developer Portal and AWS Lambda.

Part 3: Add conversational elements

As discussed in the videos, conversations with a VUI will seem more natural if the session window can be left open to continue request/response interactions. Additionally, adding randomization to the Alexa response text provides a more natural feel to the conversation.

1. Change your "tell" statements to "ask" statements as directed by the TODO's in `index.js` and include reprompt messages as necessary.
2. Change the `GET_FACT_MESSAGE` snippet to an array of at least 5 similar phrases. Randomize this portion of the Alexa response.
3. Test it. All local tests should now pass. Deploy your changes to the Interactive model in the Developer Portal and AWS Lambda.
4. Provide a screen capture (PNG) from the Service Simulator in the Developer portal of your skill working. The screen capture should include the part of the Lambda Request that shows the `GetNewYearFactIntent` and `slot` with the year requested. The Lambda Response side only needs to show that a fact was provided. Note that in order to request a slot with the Simulator, you will need to phonetically request a year. For example, if the year is 2012, the input will need to be "two thousand and twelve" rather than "2012" in the simulator. Save the screen capture for submission with the name `skill_simulator.png`



Tasks

a fact for two thousand and twelve

Ask AI History Facts Reset

Lambda Request

```
14  "type": "IntentRequest",
15  "requestId": "EdwRequestId.6d99c11c-24a5-48c1",
16  "locale": "en-US",
17  "timestamp": "2017-05-05T16:09:01Z",
18  "intent": {
19    "name": "GetNewYearFactIntent",
20    "slots": {
21      "FACT_YEAR": {
22        "name": "FACT_YEAR",
23        "value": "2012"
24      }
25    }
26  },
27 },
28 "version": "1.0"
29 }
```

Lambda Response

```
4  "outputSpeech": {
5    "type": "SSML",
6    "ssml": "<say> Here's your fact for the
7  },
8  "card": {
9    "content": "In 2012, a Google Brain clust
10   "title": "American AI History Facts",
11   "type": "Simple"
12 },
13 "reprompt": {
14   "outputSpeech": {
15     "type": "SSML",
16     "ssml": "<say> You can say tesll me a
17   }
18 }
```

Listen

Optional Additional Testing

In addition to testing with unit tests and the Service Simulator, you may find it useful to try your skill with one or more of the following:

- [Echosim.io](#): web browser simulator
- [Alexa app](#): view card
- Amazon Echo, Echo Dot, Echo Tap devices: if the device is on the same account as the development code, you can "open" the skill there.

NEXT