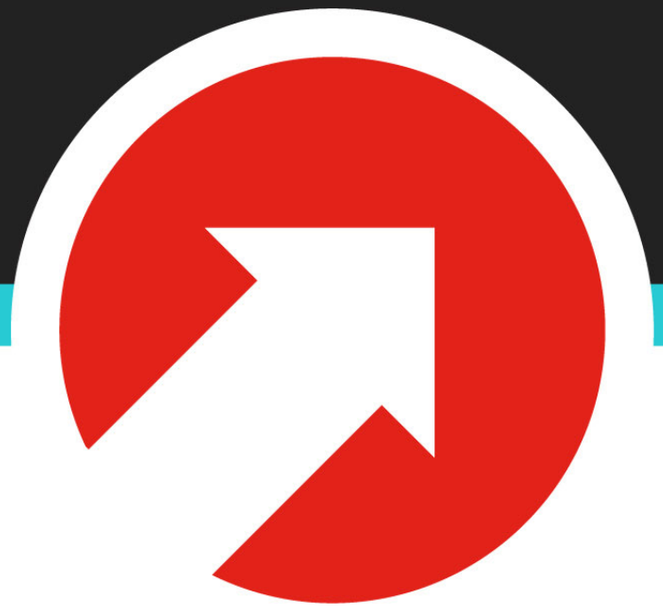

Building Alexa Skills with Node.js

Who Am I?

- My name is **Taylor Lovett**
- Vice President, Engineering at 10up
- Open source community member
- WordPress core contributor

What's All the Hype around "Voice Assistant Technology"?

Voice Assistant Technology

- Voice Assistant Technology refers software that executes commands based on human voice interaction via artificial intelligence and natural language processing.

- Search the internet, communicate, purchase items, interact with your home, play music, interact with external APIs, etc

- Integrated products - TVs (smart TVs), cars (android auto), computers, speakers, refrigerators, etc.

Voice Assistant Technology

- By 2021, there will be 7.5 billion “digital assistants” in the world. (Source AdWeek)

- Alexa (Amazon), Google, Siri (Apple), Cortana (Microsoft), Alice (Yandex)

- Alexa currently has 75% market share (Source Search Engine Land)

adweek.com/digital/infographic-how-consumers-feel-about-voice-technology-and-how-much-theyll-use-it-in-the-near-future/
searchengineland.com/google-home-vs-amazon-alexa-ces-battle-voice-ecosystem-289459

"Apps" and Developer Landscape

- "Skill" for Alexa
- "Action" for Google
- "SiriKit Apps" for Apple (HomePod not yet released)
- "Skill" for Alice

adweek.com/digital/infographic-how-consumers-feel-about-voice-technology-and-how-much-theyll-use-it-in-the-near-future/
searchengineland.com/google-home-vs-amazon-alexa-ces-battle-voice-ecosystem-289459

Alexa Skill Concepts

Alexa Skill Concepts

- Different types of Skills: custom, smart home, flash briefing, video. Custom provides the most flexibility.

- Host Skills on Amazon Lambda or on your own infrastructure. Lambda is “serverless”.

- Write Skills in Node.js

Alexa Skill Concepts

Slot

“Alexa, ask Restaurant Reviews for the best [foodType]”

Invocation Name

Utterance

Maps To

Intent

Alexa Skill Concepts

- **Invocation Name**: What a user must say to “start” your skill. Each skill has exactly one invocation name.

- **Utterance**: A spoken phrase that maps to an intent. A Skill can have many utterances.

- **Slot**: A variable within an Utterance. Variable possibilities must be defined.

- **Intent**: Each time you speak to an Alexa skill, your message is mapped to an Intent which decides what code to run within your skill.

Alexa Request

- Alexa requests are handled via JSON requests.
- When a user talks to Alexa, a JSON request object is formatted and sent to Alexa. Alexa responds with a JSON response object that is parsed and spoken to the user.
- There are three types of Alexa requests: **LaunchRequest, IntentRequest, SessionEndedRequest**

Alexa Request Types

- **LaunchRequest:** This request is triggered when a skill is first “opened”. Usually a welcome message is triggered: “Alexa, open github -> Welcome to GitHub Voice!”

- **IntentRequest:** This request is triggered when a user asks a skill to perform an action. Skills register custom intents e.g. “Get Favorites”. Skills can also use built-in intents.

- **SessionEndedRequest:** This request is triggered when a user “closes” a skill.

Built In Intents

- Built in intents allow you to handle common requests without a bunch of boilerplate code:

- **AMAZON.StopIntent**
- **AMAZON.HelpIntent**
- **AMAZON.CancelIntent**

https://developer.amazon.com/docs/custom-skills/standard-built-in-intents.html

Let's Look at a Skill

Example Skill

- I’ve created an Alexa skill for us to look at. It’s called **“Repo Voice”**

- Repo Voice allows you to ask Alexa for updates on GitHub repositories.

- https://github.com/tlovett1/repo-voice

Example Skill

Alexa ask Repo Voice for updates on my favorites

Alexa tell Repo Voice to add favorite node

Alexa tell Repo Voice to remove favorite node

Alexa ask Repo Voice what are my favorites

Alexa ask Repo Voice for updates on node

[Repo Voice Demo with Alexa]

How should a new skill be started?

ASK CLI

- **ask-cli** is a simple npm package provided by Amazon to scaffold and deploy your Alexa Skill.

> npm install -g ask-cli
> ask init
> ask new

Scaffolded Skill

.ask/ <- Deployment code for Lambda
lambda/custom/ <- Code to run in Lambda
models/ <- Interactions model(s)
skill.json <- Basic information on skill

Skill Manifest

- The skill manifest (JSON) defines properties for your skills that Alexa will interpret e.g. the name of your skill.

[Show Skill Manifest]

https://github.com/tlovett1/repo-voice/blob/master/skill.json

Interaction Model

- The interaction model (JSON) describes how users will interact with your skill.

- Most importantly, it defines intents that your skill will be handling.

[Show Interaction Model JSON]

https://github.com/tlovett1/repo-voice/blob/master/models/en-US.json

Alexa SDK

- The Node.js Alexa SDK (npm package) is the easiest way to create an Alexa skill.

- It provides an API that saves us from having to send raw JSON to our Lambda.

- It lets us easily map handlers to intents.

https://www.npmjs.com/package/alexa-sdk

Intents

- Each intent in our model is mapped to a handler in our code.
- This includes built-in intents

GetFavorites Intent

- This code will be invoked whenever a user triggers an utterance that maps to the GetFavorites intent.

- Remember, all this is defined in our interaction model.

```
'GetFavorites': function() {
  if (!this.attributes.favorites) {
    this.response.speak("You currently have no favorites. Tell alexa add favorite to add one.");
    this.emit(':responseReady');
    return;
  }

  var speech = 'Here are your favorites: ';

  Object.keys(this.attributes.favorites).forEach(function(repoKey) {
    speech += ', ' + repoKey;
  });

  this.response.speak(speech);
  this.emit(':responseReady');
},
```

External API Calls

- Alexa supports async external API calls.
- Using promises we can issue and wait for HTTP requests.
- Repo Voice makes a number of external calls to the GitHub API.
- Alexa SDK will wait for the :responseReady event to be broadcasted before responding.

```
'RepoUpdates': function() {
  var self = this;

  var repoKey = this.event.request.intent.slots.repo_name.value;
  var repo = repos[repoKey];

  if (!repo) {
    this.response.speak("Sorry, I haven't heard of that repo.");
    this.emit(':responseReady');
    return;
  }

  var speech = 'Here are the updates on ' + repoKey + '. ';

  var updates = api.getRepoUpdates(repoKey);

  updates.then(function(results) {
    results.forEach(function(result) {
      repo[result.type] = result.result
    });

    speech += util.formatUpdateSpeech(repo);

    self.response.speak(speech);
    self.emit(':responseReady');
  });
},
```

Data Persistence

- The Alexa SDK makes it very easy to persist data to DynamoDB

- DynamoDB is a NoSQL database system provided by AWS.

https://aws.amazon.com/dynamodb/

Data Persistence

- Simply tell Alexa what your DynamoDB table name should be and it will set it up for you when first invoked.

```
exports.handler = function(event, context) {
  console.log('Received event: ', JSON.stringify(event, null, 2));

  const alexa = Alexa.handler(event, context);
  alexa.dynamoDBTableName = 'repo-voice';
  alexa.registerHandlers(handlers);
  alexa.execute();
};
```

How can we test our skill?

Testing

- Use Amazon Alexa Developer dashboard.
- Deploy Skill in test mode to Alexa. (This is actually very easy).
- Use Alexa Skill Test: https://github.com/tlovett1/alexa-skill-test

Service Simulator

Use Service Simulator to test your HTTPS endpoint:

arn:aws:lambda:us-east-1:967770413081:function:ask-custom-tennis-facts-alexa-skill-default

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking. Text mode does not support launch intents and single interaction phrases.

Text | JSON

Enter Utterance

how many grand slams has roger federer won

Ask tennis-facts-alexa-skill | Reset

Service Request

1

Service Response

1

Listen

Alexa Skill Test

- Isomorphic React/Redux application.
- Live refresh as you update your skill code
- Allows you to debug skills locally.
- Enables you to view debug output.

Alexa Skill Test

> npm install -g alexa-skill-test
> cd my-skill/lambda/custom
> alexa-skill-test --interaction-model=../../models/en-US.json

Alexa Skill Test tennis-facts-alexa-skill

Alexa Skill Test lets you easily mock requests to send to your Alexa skill locally. This page will automatically refresh when you update your skill.

Request Type:

IntentRequest

Intent Name:

NumberOfGrandSlamsIntent

Slots:

player | Roger Federer

Send Request

Request


```
{
    "version": "1.0",
    "session": {
        "new": false,
        "sessionId": "amzn1.echo-api.session.abeee1a7-aee0-41e6-8192-e6faaed9f5ef",
        "application": {
            "applicationId": "amzn1.echo-sdk-ams.app.000000-d0ed-0000-ad00-000000d00ebe"
        },
        "attributes": {},
        "user": {
            "userId": "amzn1.account.AM3B227HF3FAM1B261HK7FFM3A2"
        }
    },
    "context": {
        "System": {
            "application": {
                "applicationId": "amzn1.echo-sdk-ams.app.000000-d0ed-0000-ad00-000000d00ebe"
            },
            "user": {
                "userId": "amzn1.account.AM3B227HF3FAM1B261HK7FFM3A2"
            },
            "device": {
                "supportedInterfaces": {
                    "AudioPlayer": {}
```


Response


```
{}
```

Example Bug

- Let's say we've made an error in our skill code.

- Alexa Skill Test will say "An error has occurred" in the response.

Request


```
{
    "version": "1.0",
    "session": {
        "new": false,
        "sessionId": "amzn1.echo-api.session.abeee1a7-aee0-41e6-8192-e6faaed
        "application": {
            "applicationId": "amzn1.echo-sdk-ams.app.000000-d0ed-0000-ad00-0
        },
        "attributes": {},
        "user": {
            "userId": "amzn1.account.AM3B227HF3FAM1B261HK7FFM3A2"
        }
    },
    "context": {
        "System": {
            "application": {
                "applicationId": "amzn1.echo-sdk-ams.app.000000-d0ed-0000-ad
            },
            "user": {
                "userId": "amzn1.account.AM3B227HF3FAM1B261HK7FFM3A2"
            },
            "device": {
                "supportedInterfaces": {
                    "AudioPlayer": {}
                }
            }
```


Response


```
"An error occurred"
```

Debugging an Error

- We can simply look at our terminal to see whatever errors Alexa has returned.

- With Alexa Skill Test, this is very easy. Debugging a live Alexa Skill is much more difficult requiring Lambda logging.

```
skill Lambda event: +0ms
skill { version: '1.0',
skill   session:
skill    { new: 'false',
skill      sessionId: 'amzn1.echo-api.session.abeee1a7-aee0-41e6-8192-e6faaed9f5ef',
skill      application: { applicationId: 'amzn1.echo-sdk-ams.app.000000-d0ed-0000-ad00-000000d00ebe' },
skill      user: { userId: 'amzn1.account.AM3B227HF3FAM1B261HK7FFM3A2' } },
skill   context:
skill    { System: { application: [Object], user: [Object] },
skill      AudioPlayer: { offsetInMilliseconds: '0', playerActivity: 'IDLE' } },
skill   request:
skill    { type: 'IntentRequest',
skill      requestId: 'amzn1.echo-api.request.6919844a-733e-4e89-893a-fdcb77e2ef0d',
skill      timestamp: '2015-05-13T12:34:56Z',
skill      locale: 'en-US',
skill      intent: { name: 'RepoUpdates', slots: [Object] } } } +0ms
info: START RequestId: a4051806-72fd-7764-84ca-06448f140a4d
Warning: Application ID is not set
Uncaught exception: ReferenceError: repoKesfy is not defined
ReferenceError: repoKesfy is not defined
    at Object.RepoUpdates (/Users/tlovett1/projects/alexa-skill-test/github/lambda/custom/index.js:35:38)
```

Deployment

- With ASK CLI it's very easy:

> ask deploy

Publishing

- After deploying your skill, you can submit it for review to get published in the Alexa Skill directory.

- Amazon offers rewards to developers who submit skills and free Lambda hosting.

- You'll need to submit more information about your skill via the dashboard (e.g. images).

10up is hiring!

@tlovett12

taylor.lovett@10up.com



Questions?

@tlovett12

10up.com

taylor.lovett@10up.com

taylorlovett.com

github.com/tlovett1