Alexa, Ask My Library: How Do I Build a Custom Skill to Extend Reference Services?

Christopher M. Jimenez
Florida International University, jimenezc@fiu.edu

Follow this and additional works at: https://digitalcommons.fiu.edu/glworks

Part of the Information Literacy Commons, and the Other Computer Engineering Commons

Recommended Citation
https://digitalcommons.fiu.edu/glworks/99

This work is brought to you for free and open access by the FIU Libraries at FIU Digital Commons. It has been accepted for inclusion in Works of the FIU Libraries by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.
Alexa, Ask My Library

How Do I Build a Custom Skill to Extend Reference Services?
Alexa, who is the presenter?

Christopher M. Jimenez
Web Services Librarian
Florida International University
Christopher.Jimenez@fiu.edu
@cjmnz8
github.com/cjmnz8/Echo-Show-Tell
Learning Objectives

1. Describe the major smart speakers in the industry in order to discern the best option for the project.

2. Identify several factors to consider when beginning a voice recognition project in order to incorporate a UX-First design.

3. Develop an Alexa Skill by training the AI model to recognize utterances, querying LibAnswers for information retrieval, and display responses to the end user.
Define...

...a voice-first device.
...the software that runs on a voice-first device.
...the application that the voice-assistant launches on your voice-first device.
Voice-First Device

"An always-on, intelligent piece of hardware where the primary interface is voice, both input and output"
Software that runs on a Voice-First Device

"A Voice Assistant is a digital assistant that uses voice recognition, natural language processing and speech synthesis to provide aid to users through phones and voice recognition software."
Application the voice-assistant launches on your voice-first device

Alexa = Skill
Google = Action
Siri = Hands off to iPhone
<table>
<thead>
<tr>
<th>VF Device</th>
<th>Voice Assistant</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Echo</td>
<td>Alexa</td>
<td>Skill</td>
</tr>
<tr>
<td>Google Home</td>
<td>Google Assistant</td>
<td>Action</td>
</tr>
<tr>
<td>Apple Home Pod</td>
<td>Siri</td>
<td>Handoff</td>
</tr>
</tbody>
</table>
Alexa, Ask my library

What does the voice-first device market look like today?
Infinite Dial: Use of Voice Assistants

- Smartphone: 48%
- Computer: 31%
- Smart Speaker: 20%
- Tablet: 20%
Among non-owners of smart speakers, those who currently use voice assistants are 60% more likely than those who don't to purchase a smart speaker in the next six months.
Which Device?

Infinite Dial
- Amazon: 21%
- Google: 11%
- Apple: 1%

Smart Audio
- Amazon: 78%
- Google: 41%

RKMA (2019)
- Amazon: 63%
- Google: 31%
- Apple: 12%
How Many Devices?

Infinite Dial

Mean #

<table>
<thead>
<tr>
<th>Year</th>
<th>2018</th>
<th>2019</th>
<th>2020</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.7</td>
<td>2.0</td>
<td>2.2</td>
</tr>
</tbody>
</table>
NPR #SmartAudio Report

COVID-19 Related Behaviors
COVID-19 Related Behaviors

Three-quarters of Americans 18+ say their routines have changed due to COVID-19
COVID-19 Related Behaviors

52% of voice-assistant users say they use voice tech several times a day or nearly every day, compared to 46% before the outbreak.
COVID-19 Related Behaviors

36% of U.S. adult smart speaker owners say they are using their device more to listen to music and entertainment since the outbreak, and 52% of 18-34-year-olds say the same.
71% want to buy another smart speaker to entertain children in more rooms of the house. (Up 24% from Spring 2019)
Alexa, Ask my library

What should my skill do?
What should my skill do?

• What is in demand?
• How will a user use this skill?
• Where will the user use this skill?
• How does this experience fit with the library’s current customer service experience?
FIU Libraries’ RefTech Statement

The team at FIU Libraries seeks to harness the popularity and sophistication of voice recognition technology and couple it with Springshare’s robust knowledge base software in order to create a powerful reference tool.

We are doing this by pairing the Information and Research Services Department’s most Frequently Asked Questions with an Alexa Skill using the LibAnswers API, and preloading that skill onto our Alexa Show devices for use in strategic places at our library service points.
1. We chose Amazon Echo Show devices because Amazon was the market leader.
2. We wanted to recycle information and minimize the maintenance burden.
3. We evaluated our most useful FAQ Entries and programmed the skill to answer those questions first.
4. The skill was developed for use inside the library.
Customer Service

Current Service Model:

Face-to-Face support
Phone support
Chat support
FAQ Knowledge Bank / Email support
Use your data

User-Supplied Data
- FAQ Views
- Chat Transcripts
- Query Spy

Self-Reported Data
- Reference Stats
- Anecdotal Evidence
How will a user use this skill?

• What is the scope of your project?
• What do you hope to accomplish?
• What are the overarching principles that govern what is included in your project and what is excluded from your project?
Alexa, Ask my library

How do I code this skill myself?
Two Parts to Skill Development

Front End

• Train the AI
• Use reference interview strategies

Back End

• Programming that fulfills the request made by a user
Front End: Getting Started

  • User Interface: walks you through training your skill to tie utterances to intents.

• What you will do in the Developer Portal
  • Name your skill (2 or 3 word invocation name)
  • Train your skill to recognize questions
Front End: Step-by-Step

1. Name your skill
2. Create intents
3. Describe utterances
4. Define & assign slots (Optional: Interaction Model)
5. Evaluate the model
Terminology

Intents, Utterances, & Slots
Intent

“An *intent* represents an action that fulfills a user’s spoken request. Intents can optionally have arguments called *slots*. Intents are specified in a JSON structure called the *intent schema*.”

- What is the user actually trying to get the device to do?
- What is the end goal?
Utterance

“A set of likely spoken phrases mapped to the intents. This should include as many representative phrases as possible.”

- What are a few examples of how users voice their questions?
- How do people express themselves?
Slots

“A representative list of possible values for a slot. Custom slot types are used for lists of items that are not covered by one of Amazon's built-in slot types.”

- Synonyms that might be used to replace key terms in a question.
- Example: Library
Let’s Build!

Intents, Utterances, & Slots

+ Evaluate the Model
Create an Intent

How to get started

Alexa Skills Kit Developer Tutorial

Resources

Update your live skill instantly

Qualify for live updates to your skill if changes are made to slot values and/or sample utterances. Learn more about live updates to your skill.

Catalog Management

Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on an Alexa Skill.

Skill builder checklist

1. Invocation Name
   Enter an invocation name for your skill

2. Intents, Samples, and Slots
   Add at least one intent and one sample utterance

3. Build Model
   Successfully build your interaction model

4. Endpoint
   Set a web service endpoint to handle skill requests
Add Utterances

Updates to sample utterances qualify for instant live updates. Learn more about live updates to your skill.

**Intents** / newIntent

Sample Utterances (0)

What might a user say to invoke this intent?
Define Slots
Assign Slots

Slot Types / library

Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on Alexa Skills Kit Command Line Interface (ASK CLI). Learn more about Catalog management.

Slot Values (2)

Enter a new value for this slot type

VALUE

ID (OPTIONAL)

SYNONYMS (OPTIONAL)
Evaluate the model
Back End: Getting Started

• Developer Portal: https://developer.amazon.com/
• Alexa Skill Code Generator: https://s3.amazonaws.com/webappvui/skillcode/v2/index.html
• Lambda Console: https://console.aws.amazon.com/lambda
Back End: Step-by-Step

1. Export JSON / Generate js code
2. Create serverless repository on the Lambda Console
3. Associate serverless repository to your skill using the Amazon Resource Name (ARN)
4. Paste the generated code (Working Alexa Skill)
5. Customize the code
Export JSON / Generate js code
Create Serverless Repository

AWS Lambda

Resources for US East (N. Virginia)

<table>
<thead>
<tr>
<th>Lambda function(s)</th>
<th>Code storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>517.1 kB (0% of 75.0 GB)</td>
</tr>
</tbody>
</table>

Full account concurrency

1000

Unreserved account concurrency

1000

Account-level metrics

The charts below show metrics across all your Lambda functions in this AWS Region.

Error count and success rate (%)

<table>
<thead>
<tr>
<th>Count</th>
<th>Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

No data available.

Try adjusting the dashboard time range.

Throttles

<table>
<thead>
<tr>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

No data available.

Try adjusting the dashboard time range.

Invocations

<table>
<thead>
<tr>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

No data available.

Try adjusting the dashboard time range.
Associate the Repository to Your Skill

Deployment status for serverlessrepo-alexa-new-skill

Your application has been deployed
Review the application's README for what to do next.

Permissions

The section outlines the SAM policy templates and additional capabilities used by this application.

**SAM policy templates**

This application does not use any SAM template policies. To learn more about SAM policy templates, please see our documentation.

**Capabilities**

This application does not require any additional capabilities. To learn more, see our documentation.

Resources

View CloudFormation Stack

<table>
<thead>
<tr>
<th>Resource</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>alexaskillkitnodesfactskill</td>
<td>CREATE_COMPLETE</td>
</tr>
<tr>
<td>alexaskillkitnodesfactskillAlexaTriggerPermission</td>
<td>CREATE_COMPLETE</td>
</tr>
<tr>
<td>alexaskillkitnodesfactskillRole</td>
<td>CREATE_COMPLETE</td>
</tr>
</tbody>
</table>

Readme file

alexa-skills-kit-nodeis-factskill
Paste your generated code
Custom Code: LibAnswers API

https://ask.springshare.com/libanswers/faq/1669

- GET faqs/:id
- https://libanswers.../api/1.0/faqs/:id

HOST: 'https://libanswers.../api'
Stable content in the HTTPGET function

PARAM: '/1.0/faqs/' + answerId + '?group_id=000&iid=000'
Variable content in the INTENT function
function httpGet(param) {
  return new Promise(((resolve, reject) => {
    var options = {
      host: 'libanswers.fiu.edu',
      path: param,
      method: 'GET',
      rejectUnauthorized: false
    };

    const request = https.request(options, (response) => {
      response.setEncoding('utf8');
      let returnData = '';

      response.on('data', (chunk) => {
        returnData += chunk;
      });

      response.on('end', () => {
        resolve(JSON.parse(returnData));
      });

      response.on('error', (error) => {
        reject(error);
      });

      request.end();
    }));
  }));
Customize Node.js: Intent Overview

```javascript
//SKIPPING CUSTOM INTENT HANDLERS
//EXAMPLE: IF A USER ASKS HOW TO PRINT, THIS HANDLER WILL RUN.

* const hangprint_handler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    return request.requestType === 'intentRequest' & request.intent.name === 'hangprint';
  },
  async handle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    const responseBuilder = handlerInput.responseBuilder;
    let sessionAttributes = handlerInput.attributesManager.getSessionAttributes();
    //mensagem that gives "say" a value
    let say = 'Hello from ' + invocationName + ' your F, I, U Libraries voice assistant.';
    //Retrieves the answer from your instance of libanswers
    let answerId = ''; enter the answer ID from your system */
    //Inspect your URL structure to make sure "param" will build the URL correctly. It is your "HOST" + "API VERSION" + "ANSWER ID" + "GROUP ID"
    //Mostly "param" with your GROUP ID & LIEANSWERS system ID {{{6
    let param = '/1.0/ads' + answerId + '/group_id=0000110000000000';
    let slotStatus = ' '; //Retrieved from libanswers
    let resolvedSlot = ''; //Sends the request to your HTTPS function and waits for the parsed JSON data
    const response = await responseBuilder
      .addToSpeech('Hello from your F, I, U Libraries voice assistant.
      ');
    console.log(response);
    say = slotStatus;
    //Use slot notation to feed data into your skill
    return responseBuilder
      .addRequestCard
      .addSimpleCard
      .withTitle('Your skill answer')
      .withSubtitle('Hello from your F, I, U Libraries voice assistant.
      ');
    //The standard card displays text on echo show screens. I used a 'stripTags()' function to remove HTML tags before displaying on the screen.
    .withStandardCard(stripHtml(response.faq[0].question),
      stripHtml(response.faq[0].answer),
      welcomeCarding.smallerImageUrl, welcomeCarding.largeImageUrl
    )
    .withShouldEndSession(true)
    .getResponse();
};
```
Customize Node.js: param

```javascript
// RETRIEVES THE ANSWER FROM YOUR INSTANCE OF LIBANSWERS
let answerId = /* ENTER THE ANSWER ID FROM YOUR SYSTEM */;

// INSPECT YOUR URL STRUCTURE TO MAKE SURE "PARAM" WILL BUILD THE URL CORRECTLY. IT IS YOUR "HOST" + "API VERSION" + "ANSWER ID" + "GROUP ID"
// MODIFY "PARAM" WITH YOUR GROUP ID & LIBANSWERS SYSTEM ID (iid)
let param = '/1.0/faqs/' + answerId + '?group_id=000&id=000';
let slotStatus = '';
let resolvedSlot;

// SENDS THE REQUEST TO YOUR HTTPGET FUNCTION AND WAITS FOR THE PARSED JSON DATA
const response = await httpGet(param);
console.log(response);
say += slotStatus;
```
Customize Node.js: responseBuilder

```javascript
// USE DOT NOTATION TO FEED DATA INTO YOUR SKILL
return responseBuilder
  .speak(response.faqs[0].answer)

// THE STANDARD CARD DISPLAYS TEXT ON ECHO SHOW SCREENS. I USED A "stripHtml" FUNCTION TO REMOVE HTML TAGS BEFORE DISPLAYING ON THE SCREEN.
  .withStandardCard(stripHtml(response.faqs[0].question),
    '\n' + stripHtml(response.faqs[0].answer),
    welcomeCardImg.smallImageUrl, welcomeCardImg.largeImageUrl
  )
  .withShouldEndSession(true)
  .getResponse();
```
Alexa, Ask my library

What did we learn today?
Alexa, Ask my library

**Voice First Market:** This is a major emerging market.

**UX-First Design:** Consider how voice-first tech might fit into your customer service model.

**Coding:** One way to program a skill to communicate with LibAnswers.
Alexa, show me the references.


Alexa, are there any questions?

Christopher M. Jimenez
Web Services Librarian
Florida International University

Christopher.Jimenez@fiu.edu
@cjmnz8
github.com/cjmnz8/Echo-Show-Tell