

5-20-2020

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](#)

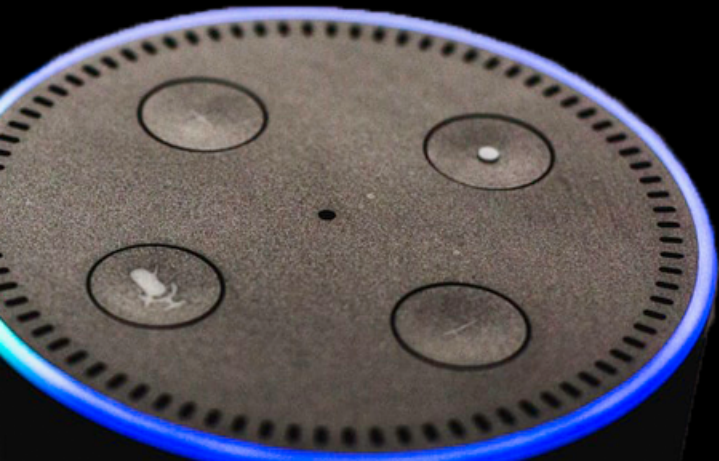
Recommended Citation

Jimenez, Christopher M., "Alexa, Ask My Library: How Do I Build a Custom Skill to Extend Reference Services?" (2020). *Works of the FIU Libraries*. 99.
<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



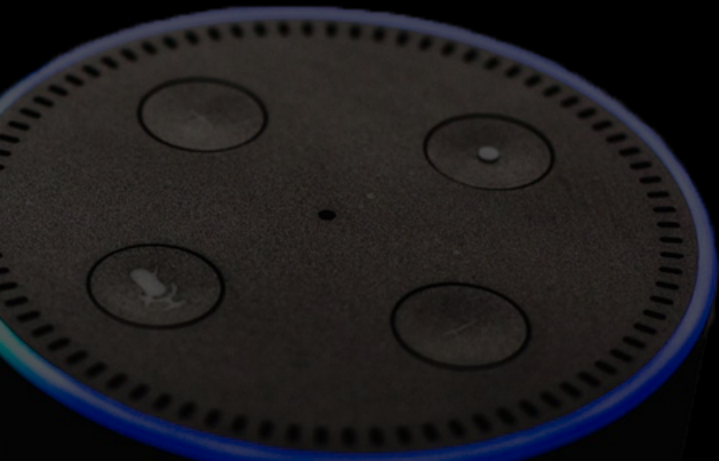
Christopher.Jimenez@fiu.edu



[@cjmnz8](https://twitter.com/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.

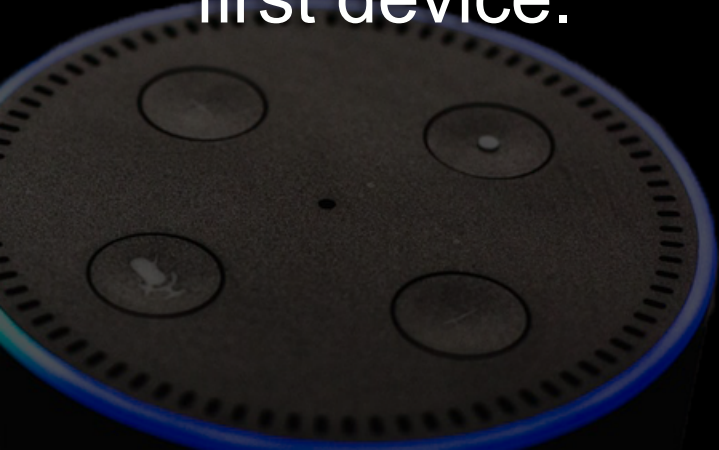


Define...

...a voice-first device.

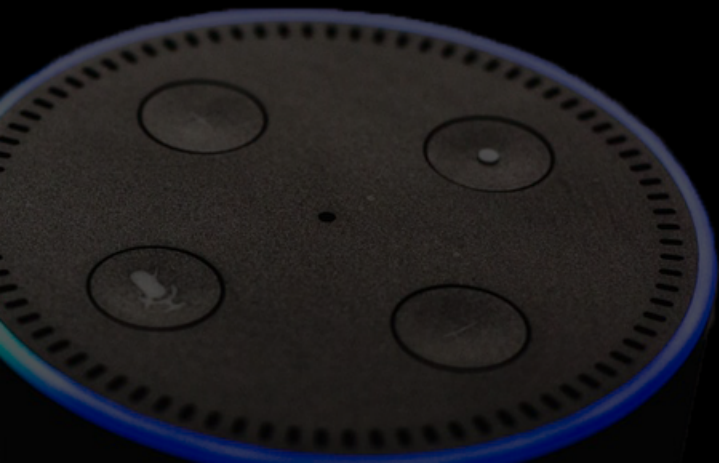
...the software that runs on a voice-first device.

...the application that the voice-assistant launches on your voice-first device.



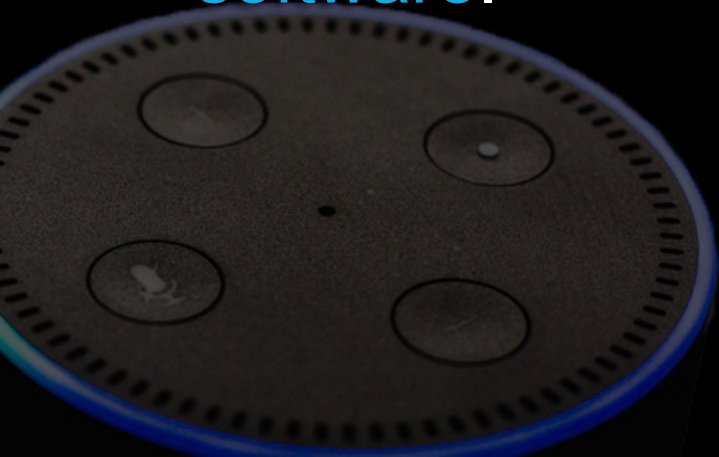
Voice-First Device

"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](#)."

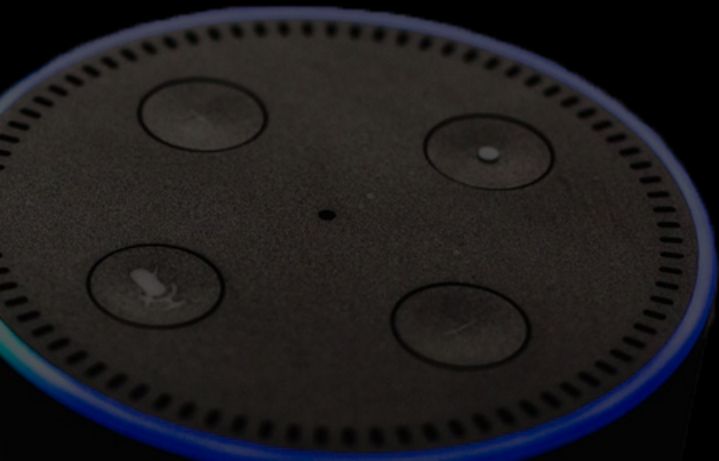


Application the voice-assistant launches on your voice-first device

Alexa = Skill

Google = Action

Siri = Hands off to iPhone



Summary

VF Device

Amazon Echo

Google Home

Apple Home Pod

Voice Assistant

Alexa

Google Assistant

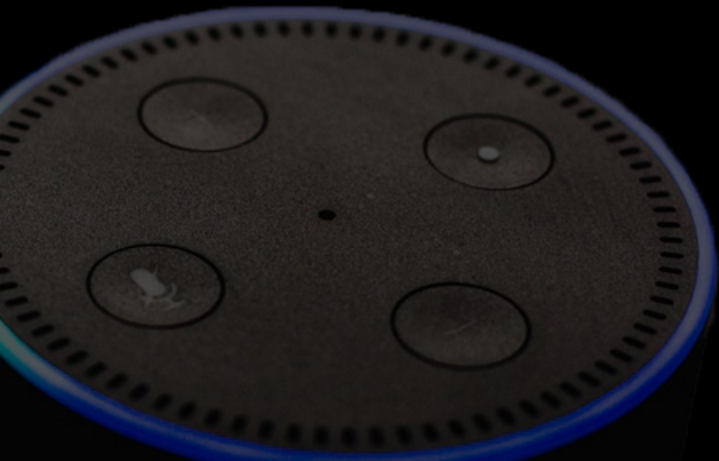
Siri

Application

Skill

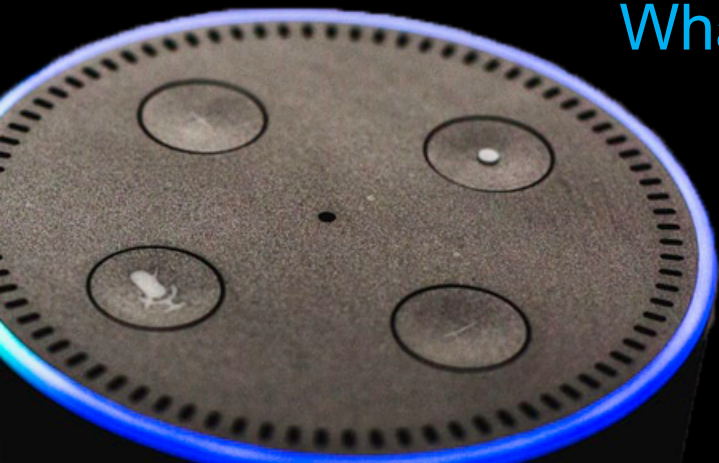
Action

Handoff

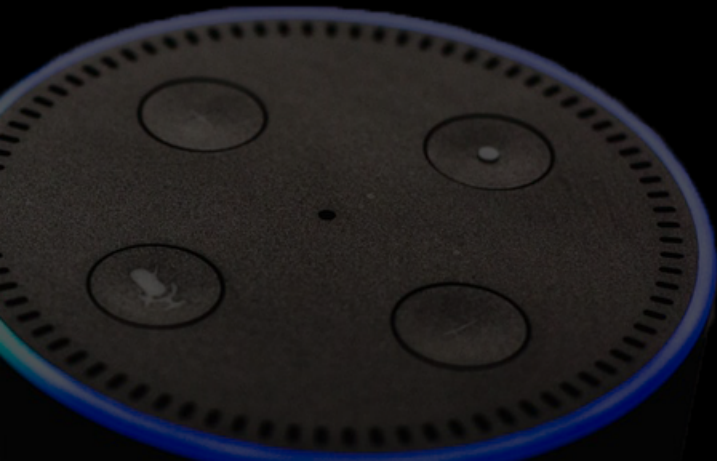


Alexa, Ask my library

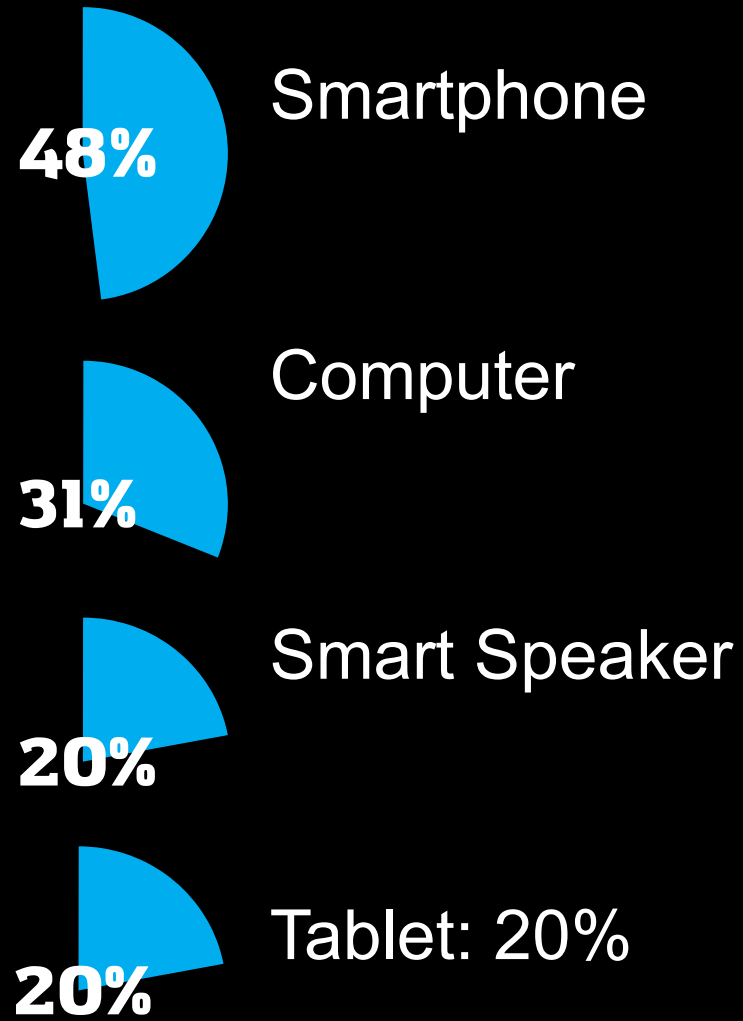
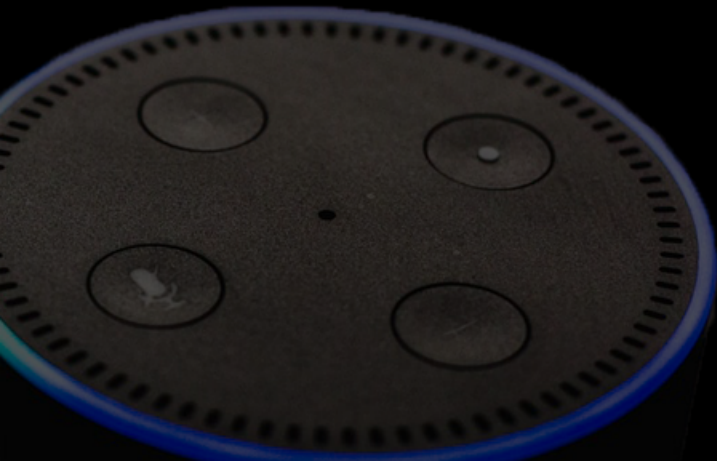
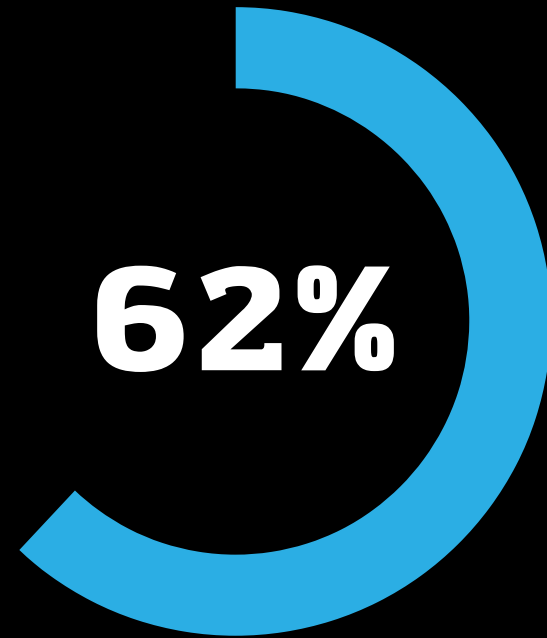
What does the voice-first device market look like today?



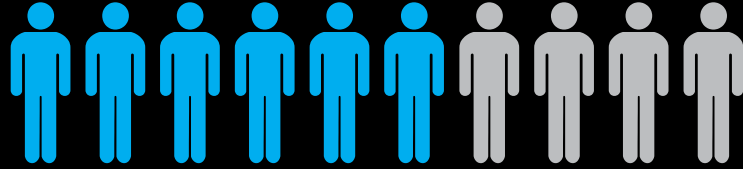
Market Research



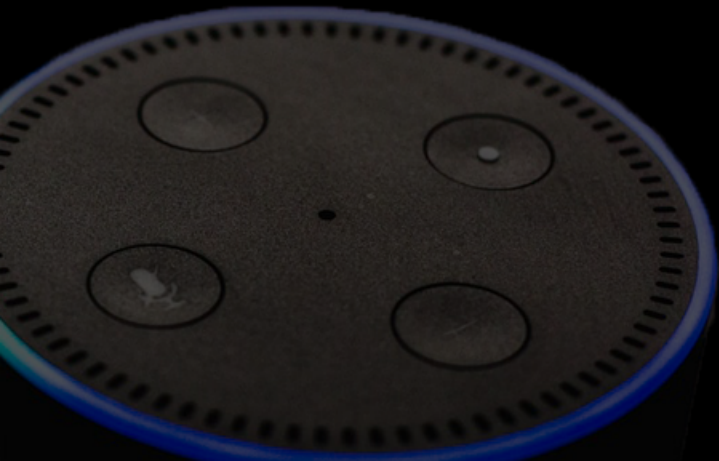
Infinite Dial: Use of Voice Assistants



NPR: Conversion Rate

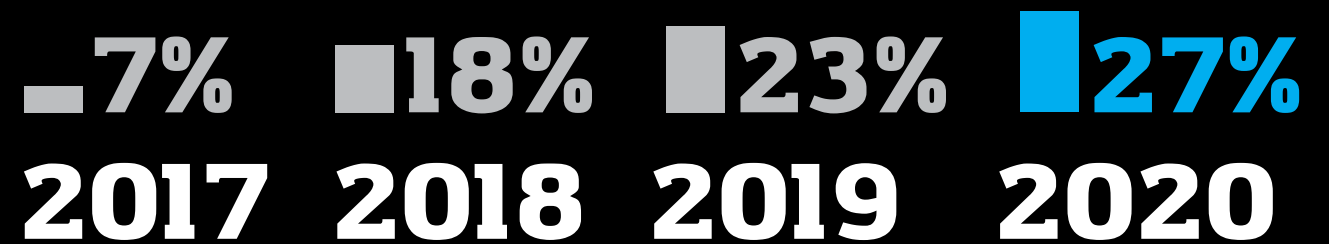


Among non-owners of smart speakers, those who currently use voice assistants are **60%** more likely than those who don't to purchase a smart speaker in the next six months

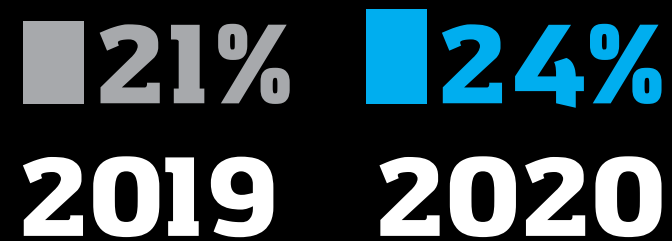


Market Penetration

Infinite Dial



Smart Audio



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

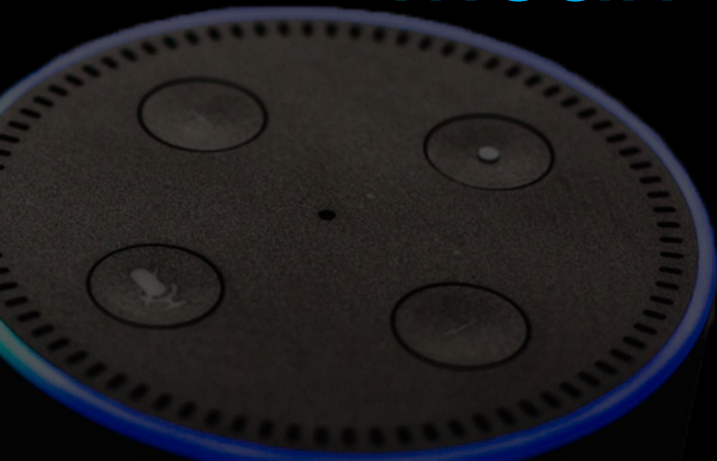


Mean #

1.7

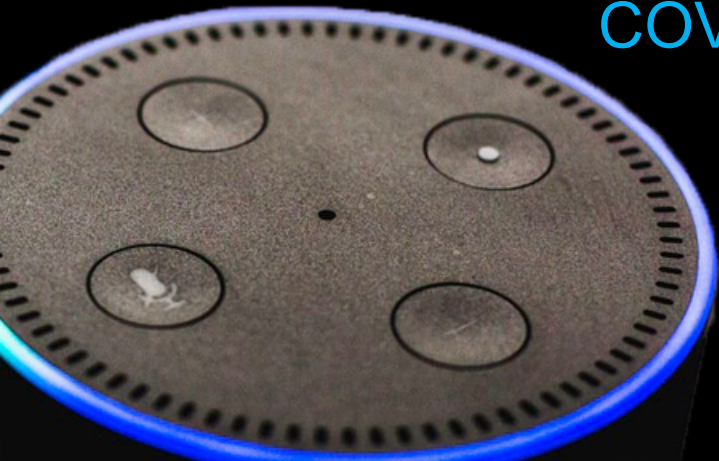
2.0

2.2



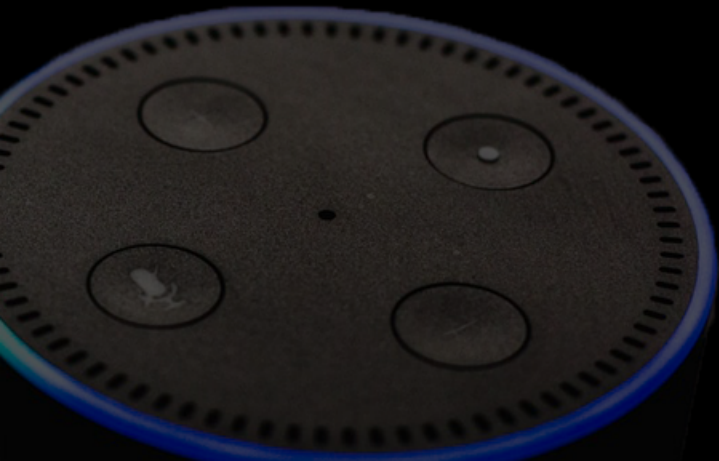
NPR #SmartAudio Report

COVID-19 Related Behaviors



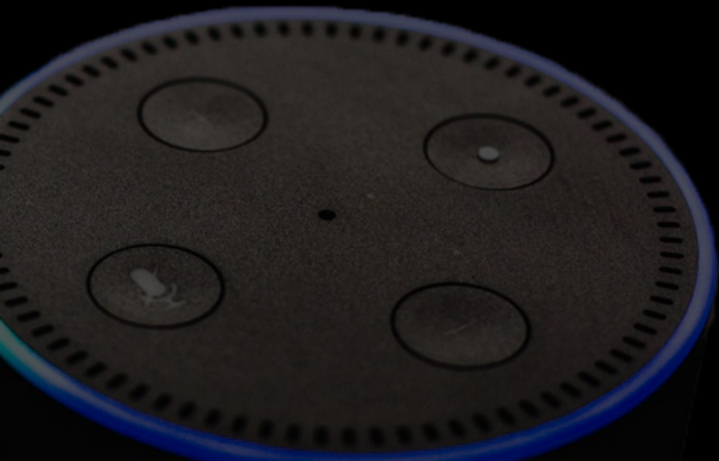
COVID-19 Related Behaviors

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



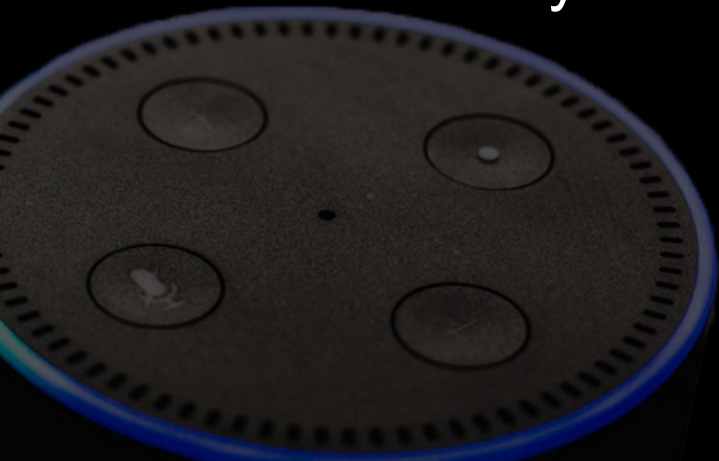
COVID-19 Related Behaviors

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



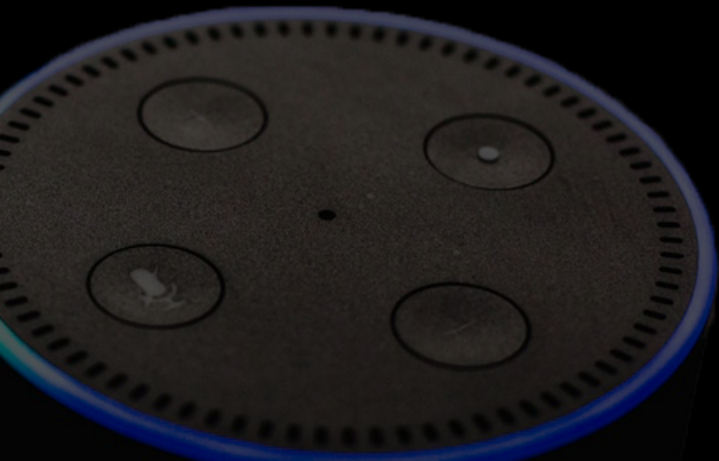
COVID-19 Related Behaviors

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.



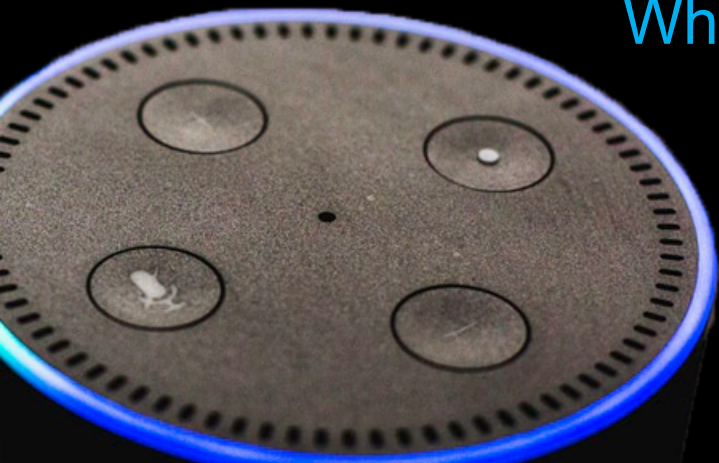
COVID-19 Related Behaviors

71% want to buy another smart speaker to entertain children in more rooms of the house. (Up 24% from Spring 2019)



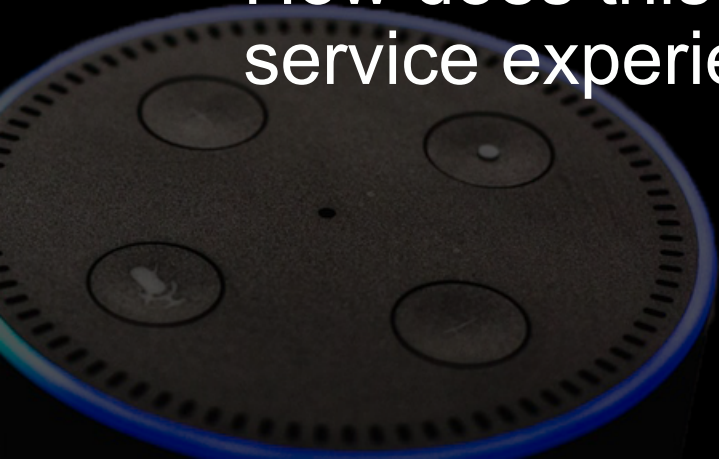
Alexa, Ask my library

What should my skill do?



What should my skill do?

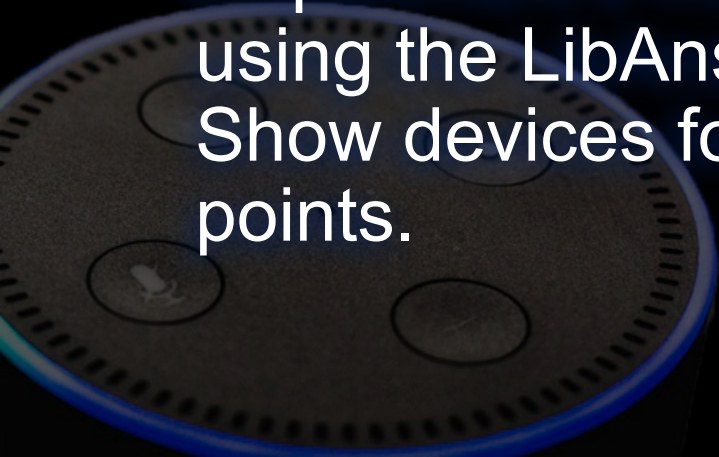
- What is in demand?
- How will a user use this skill?
- Where will the user use this skill?
- How does this experience fit with the library's current customer service experience?



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.



FIU Libraries' RefTech Statement

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



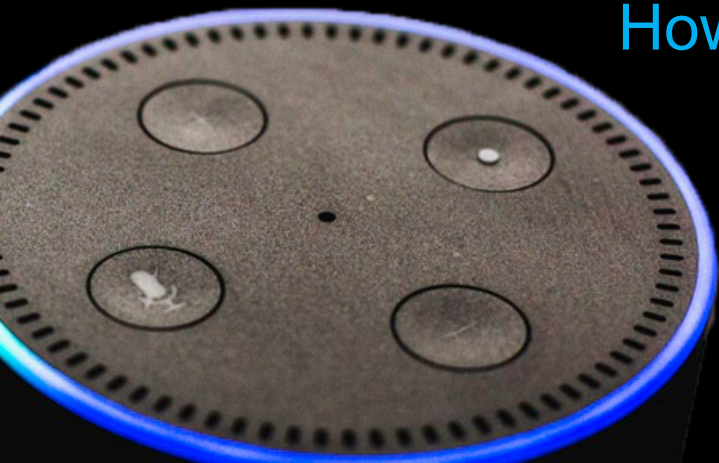
How will a user use this skill?

- 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?



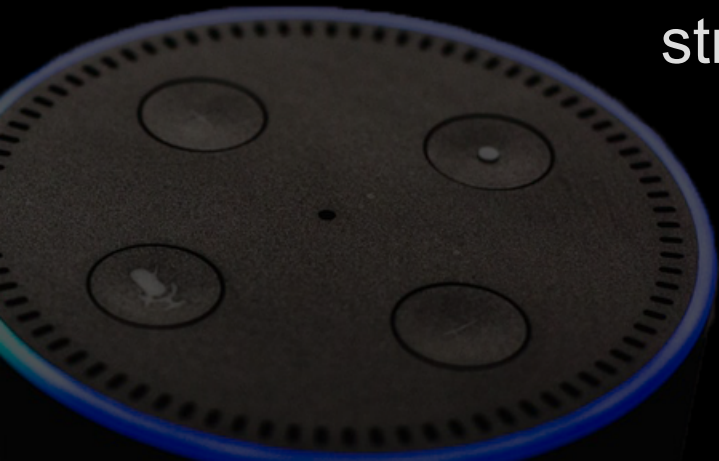
Two Parts to Skill Development

Front End

- Train the AI
- Use reference interview strategies

Back End

- Programming that fulfills the request made by a user



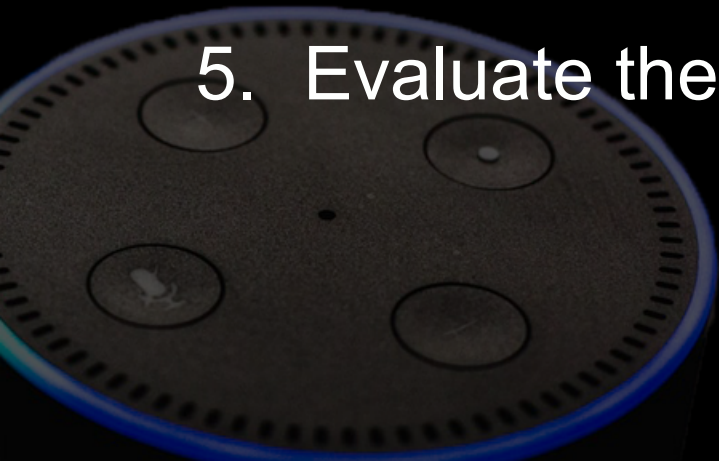
Front End: Getting Started

- Developer Portal (<https://developer.amazon.com/>)
 - 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



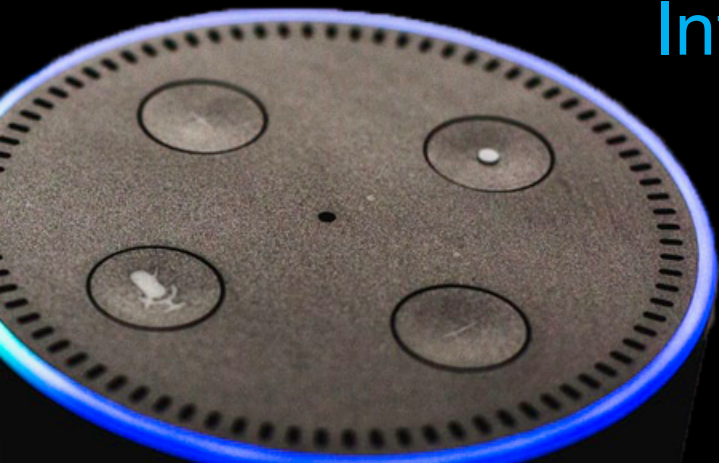
Front End: Step-by-Step

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*.”

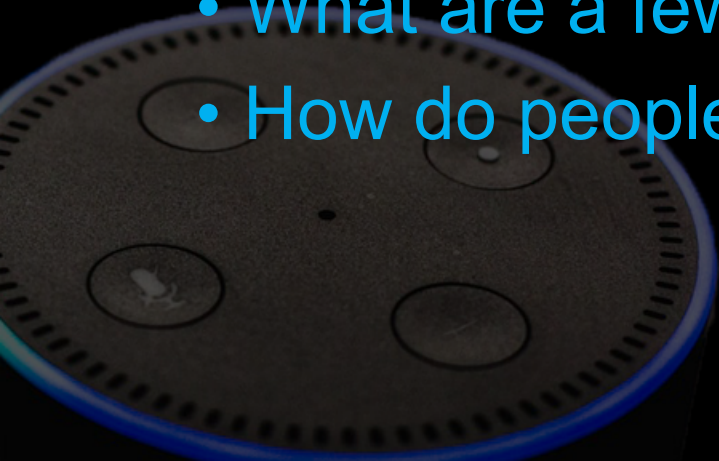
- What is the user actually trying to get the device to do?
- What is the end goal?



Utterance

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

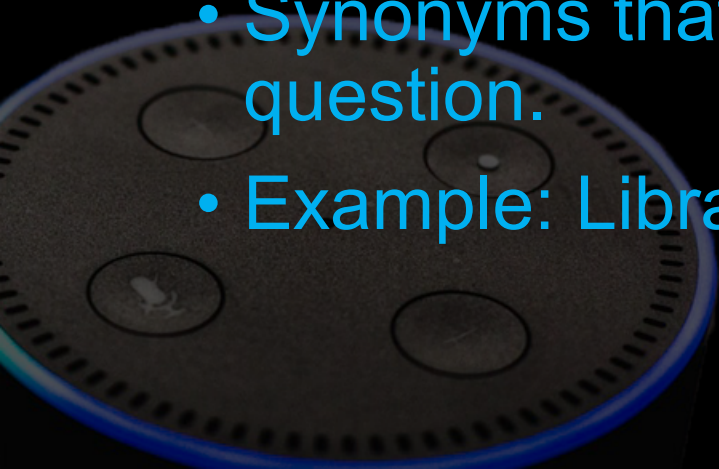
- What are a few examples of how users voice their questions?
- How do people express themselves?



Slots

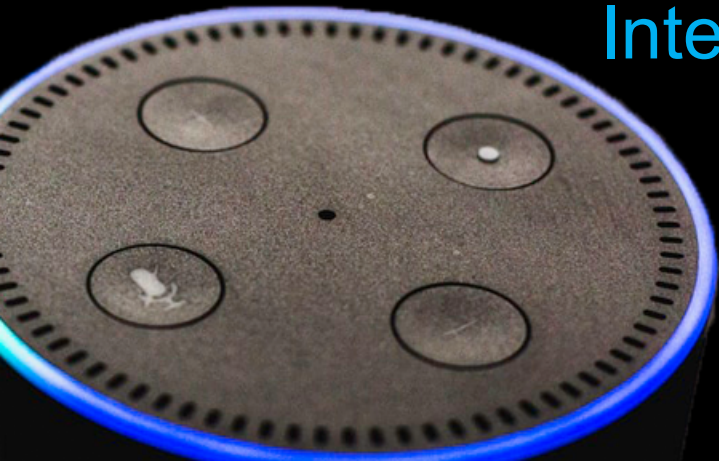
“A representative list of possible values for a slot. Custom slot types are used for lists of items that are not covered by one of Amazon's built-in slot types.”

- 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

The screenshot displays the Alexa Developer Console interface. The browser address bar shows the URL: `developer.amazon.com/alexa/console/ask/build/custom/amzn1.ask.skill.ec86bc12-fd26-4d2e-8ebd-041bee75766a/devel...`. The page title is "alexa developer console". The navigation menu includes "Your Skills", "My New Skill", "Build", "Code", "Test", "Distribution", "Certification", and "Analytics".

The left sidebar shows the "CUSTOM" section with a dropdown menu for "English (US)". Below it, there are sections for "Interaction Model", "Utterance Conflicts (0)", "Invocation", "Intents (5)", and "Slot Types (0)". The "Intents (5)" section is expanded, showing "HelloWorldIntent" and "Built-In Intents (4)" which includes "AMAZON.CancelIntent", "AMAZON.HelpIntent", "AMAZON.StopIntent", and "AMAZON.NavigateHomeIntent".

The main content area is titled "How to get started" and features a video player for "Alexa Skills Kit Developer Tutorial...". Below the video, there is a "Resources" section with links for "Update your live skill instantly" and "Catalog Management".

On the right side, there is a "Skill builder checklist" section. It contains four items, each marked as "REQUIRED" and with a green checkmark:

- 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

Below the checklist, there is a note: "Complete these steps to be able to test your skill using the simulator in the test tab, or with your echo device."

Add Utterances

The screenshot displays the Alexa Developer Console interface. At the top, the browser address bar shows the URL: `developer.amazon.com/alexa/console/ask/build/custom/amzn1.ask.skill.ec86bc12-fd26-4d2e-8ebd-041bee75766a/devel...`. The console header includes navigation tabs: `< Your Skills`, `My New Skill`, `Build`, `Code`, `Test`, `Distribution`, `Certification`, and `Analytics`. A search icon and `FR` are also present.

The left sidebar is titled `CUSTOM` and contains the following items:

- `Interaction Model`
- `Utterance Conflicts (0)`
- `Invocation`
- `Intents (6)` (with a `+ Add` button)
 - `HelloWorldIntent`
 - `newIntent` (highlighted)
 - `Built-In Intents (4)`
 - `AMAZON.CancelIntent`
 - `AMAZON.HelpIntent`
 - `AMAZON.StopIntent`
 - `AMAZON.NavigateHomeIntent`
- `Slot Types (0)` (with a `+ Add` button)

The main content area features several buttons: `Save Model`, `View Model Versions`, `Build Model`, `Update live skill`, and `Evaluate Model`. A lightbulb icon with the text `Updates to sample utterances qualify for instant live updates. Learn more about live updates to your skill.` is displayed.

The current view is `Intents / newIntent`. Below this, it shows `Sample Utterances (0)` with `Bulk Edit` and `Export` options. A text input field contains the placeholder text `What might a user say to invoke this intent?` and a `+` button to add new utterances. A speech bubble icon is visible below the input field.

Define Slots

The screenshot shows the Amazon Developer console interface for defining a custom skill. The left sidebar contains a navigation menu with the following items:

- English (US)
- CUSTOM
- Interaction Model
- Utterance Conflicts (0)
- Invocation
- Intents (6) + Add
 - HelloWorldIntent
 - newIntent**
 - Built-In Intents (4)
 - AMAZON.CancelIntent
 - AMAZON.HelpIntent
 - AMAZON.StopIntent
 - AMAZON.NavigateHomeIntent
 - Slot Types (0) + Add
 - JSON Editor

The main content area displays the 'Intents / newIntent' page. At the top, there are buttons for 'Save Model', 'View Model Versions', 'Build Model', 'Update live skill', and 'Evaluate Model'. A light blue box contains a lightbulb icon and the text: 'Updates to sample utterances qualify for instant live updates. Learn more about live updates to your skill.'

The 'Intents / newIntent' section shows 'Sample Utterances (4)' with a 'Bulk Edit' and 'Export' option. A text input field asks 'What might a user say to invoke this intent?'. Below this, four sample utterances are listed, each with a trash icon for deletion:

- when does the library close today
- does the library open this weekend
- when does the library open
- is the library open

At the bottom right, there is a pagination indicator: '< 1 - 4 of 4 >'.

Assign Slots

The screenshot displays the Amazon Developer console interface for configuring a custom skill. The top navigation bar includes 'Your Skills', 'My New Skill', 'Build', 'Code', 'Test', 'Distribution', 'Certification', and 'Analytics'. The left sidebar shows a tree view of the skill's configuration, with 'Slot Types (1)' selected. The main content area is titled 'Slot Types / library' and contains a table for defining slot values. A search bar and 'Bulk Edit' and 'Export' buttons are visible above the table. The table has columns for 'VALUE', 'ID (OPTIONAL)', and 'SYNONYMS (OPTIONAL)'. A text input field is provided for entering a new value for the selected slot type.

English (US)

Save Model View Model Versions Build Model Update live skill Evaluate Model

CUSTOM

Interaction Model

Utterance Conflicts (0)

Invocation

Intents (6)

- HelloWorldIntent
- newIntent
- library
- Built-In Intents (4)
 - AMAZON.CancelIntent
 - AMAZON.HelpIntent
 - AMAZON.StopIntent
 - AMAZON.NavigateHomeIntent

Slot Types (1)

Updates to slot values qualify for instant live updates. [Learn more about live updates to your skill.](#)

Slot Types / library

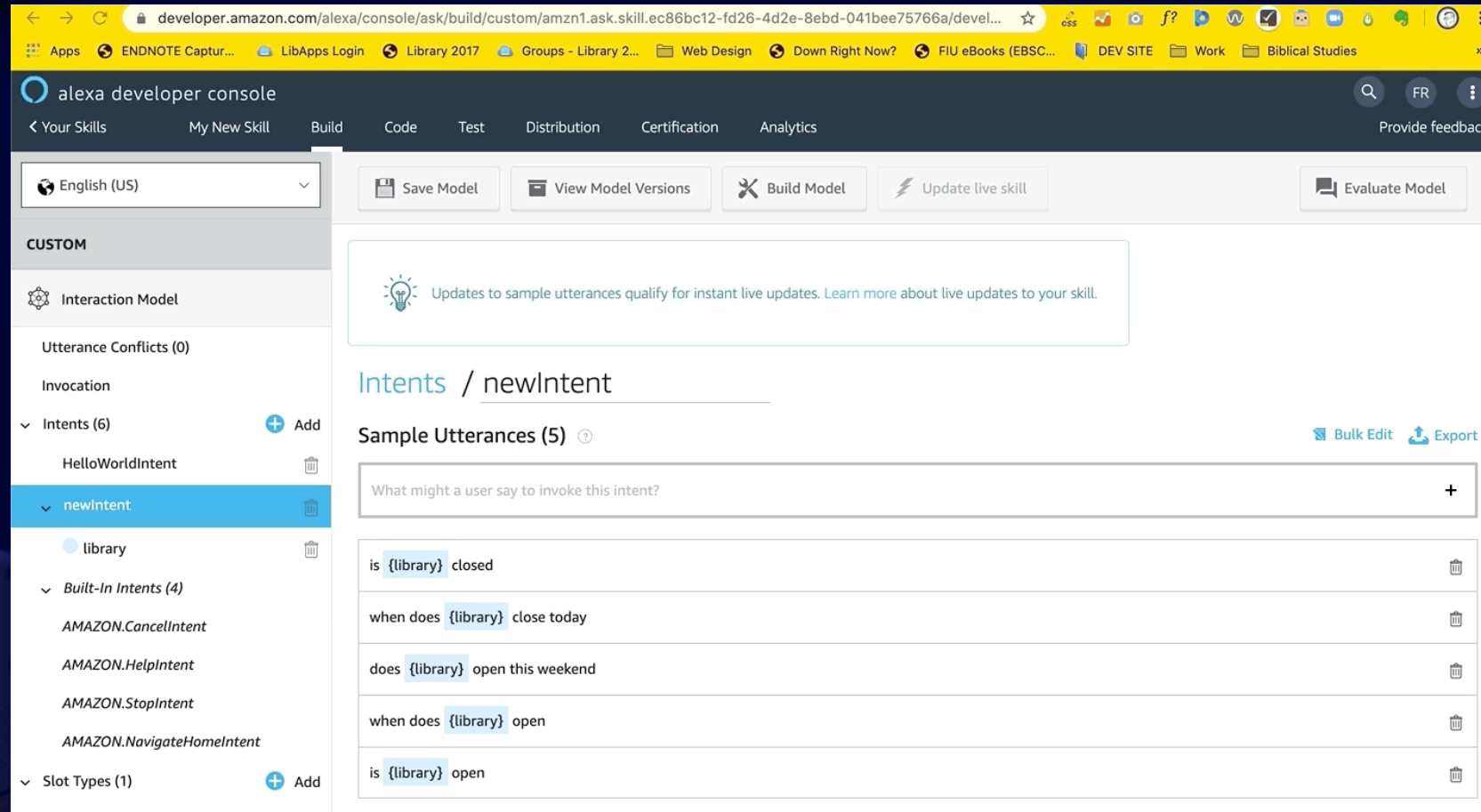
Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on Alexa Skill Management API (SMAPI) and the Alexa Skills Kit Command Line Interface (ASK CLI). [Learn more about Catalog management.](#)

Slot Values (2) [Bulk Edit](#) [Export](#)

Enter a new value for this slot type

VALUE	ID (OPTIONAL)	SYNONYMS (OPTIONAL)

Evaluate the model



The screenshot displays the Amazon Alexa Developer Console interface for a custom skill. The browser address bar shows the URL: `developer.amazon.com/alexa/console/ask/build/custom/amzn1.ask.skill.ec86bc12-fd26-4d2e-8ebd-041bee75766a/devel...`. The console header includes the 'alexa developer console' logo and navigation tabs: 'Your Skills', 'My New Skill', 'Build', 'Code', 'Test', 'Distribution', 'Certification', and 'Analytics'. A search bar and 'Provide feedback' link are also present.

The main content area features a top navigation bar with buttons for 'Save Model', 'View Model Versions', 'Build Model', 'Update live skill', and 'Evaluate Model'. Below this, a light blue box contains a lightbulb icon and the text: 'Updates to sample utterances qualify for instant live updates. [Learn more about live updates to your skill.](#)'

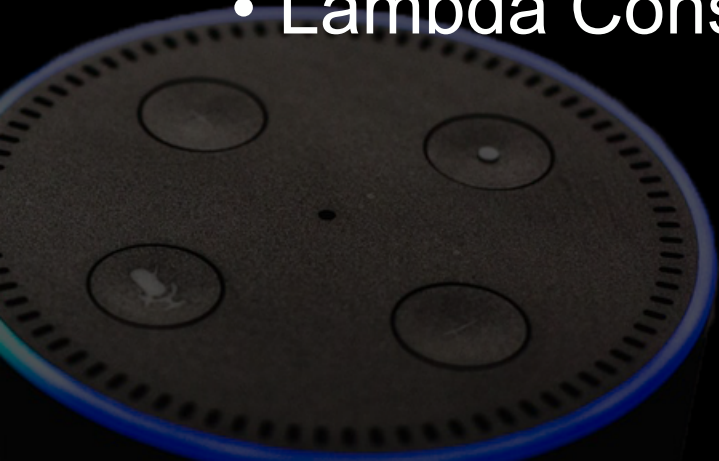
The central section is titled 'Intents / newIntent'. Below the title, there are 'Bulk Edit' and 'Export' options. A text input field prompts the user: 'What might a user say to invoke this intent?'. Below the input field, a list of five sample utterances is displayed, each with a trash icon for deletion:

- is {library} closed
- when does {library} close today
- does {library} open this weekend
- when does {library} open
- is {library} open

The left sidebar shows a navigation menu for the 'CUSTOM' skill. The 'Intents (6)' section is expanded, showing 'HelloWorldIntent', 'newIntent' (selected), and 'library'. Other sections include 'Utterance Conflicts (0)', 'Invocation', 'Built-In Intents (4)' (listing 'AMAZON.CancelIntent', 'AMAZON.HelpIntent', 'AMAZON.StopIntent', and 'AMAZON.NavigateHomeIntent'), and 'Slot Types (1)'.

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

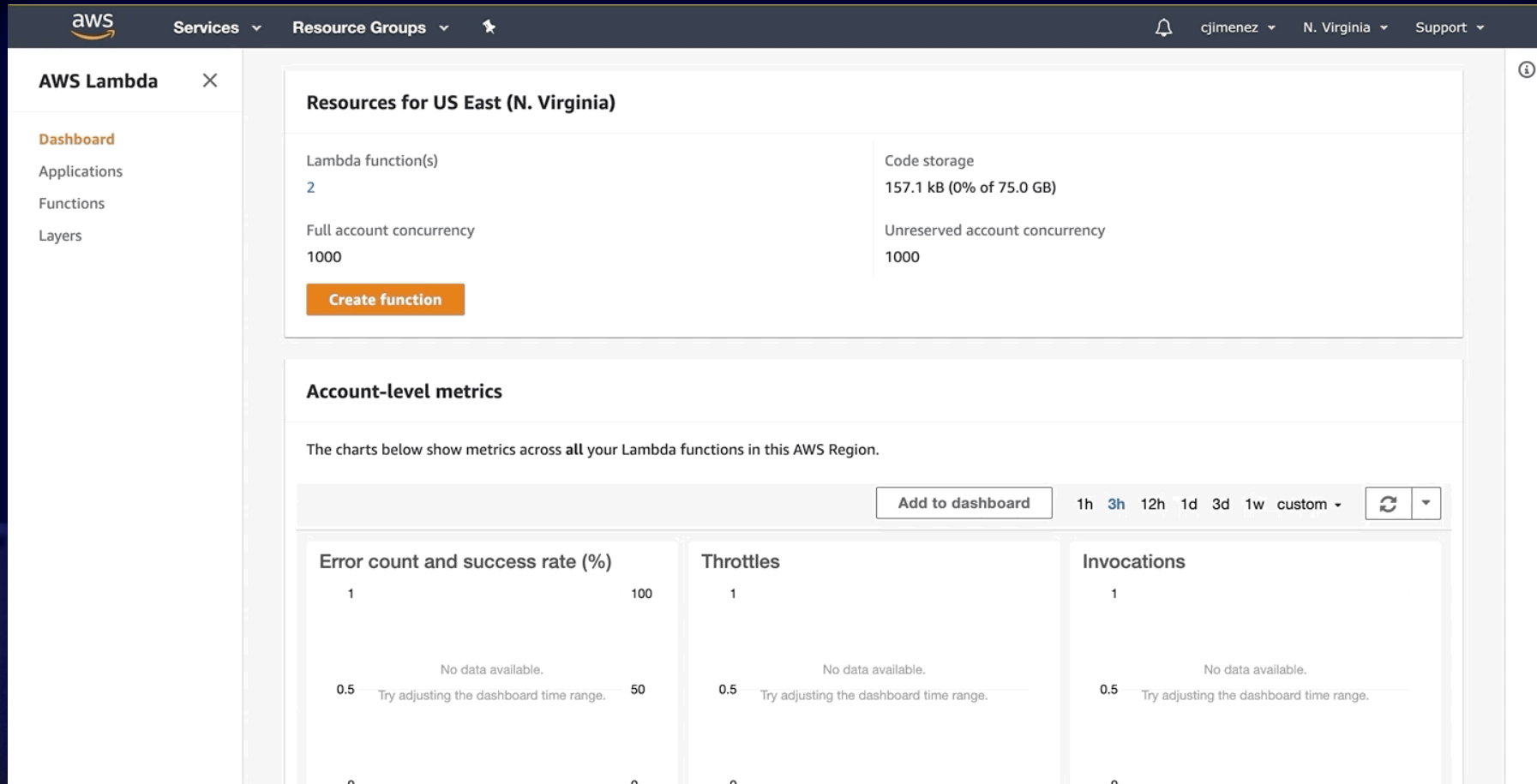
1. Export JSON / Generate js code
2. Create serverless repository on the Lambda Console
3. Associate serverless repository to your skill using the Amazon Resource Name (ARN)
4. Paste the generated code(Working Alexa Skill)
5. Customize the code



Export JSON / Generate js code

The screenshot displays the Alexa Developer Console interface. At the top, the navigation bar includes 'alex developer console', a search icon, 'FR', and a menu icon. Below this, a secondary navigation bar shows 'Your Skills', 'My New Skill', 'Build', 'Code', 'Test', 'Distribution', 'Certification', and 'Analytics'. The 'Build' tab is active. On the right side of this bar are buttons for 'Save Model', 'View Model Versions', 'Build Model', 'Update live skill', and 'Evaluate Model'. A language dropdown menu is set to 'English (US)'. The left sidebar contains a tree view with categories: 'CUSTOM', 'Interaction Model', 'Utterance Conflicts (0)', 'Invocation', 'Intents (6)' (with sub-items: HelloWorldIntent, newIntent, library, and Built-In Intents (4) including AMAZON.CancelIntent, AMAZON.HelpIntent, AMAZON.StopIntent, and AMAZON.NavigateHomeIntent), and 'Slot Types (1)'. The main content area features a light blue notification box about slot value updates. Below it is the 'Slot Types / library' section, which includes another notification box about catalog management. The 'Slot Values (2)' section has 'Bulk Edit' and 'Export' buttons, a search bar, and a text input field for adding new values. At the bottom, a table header is visible with columns for 'VALUE', 'ID (OPTIONAL)', and 'SYNONYMS (OPTIONAL)'.

Create Serverless Repository



The screenshot displays the AWS Lambda console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (cjimenez, N. Virginia, Support). The left sidebar shows the 'AWS Lambda' menu with options for Dashboard, Applications, Functions, and Layers. The main content area is titled 'Resources for US East (N. Virginia)' and provides a summary of account-level resources:

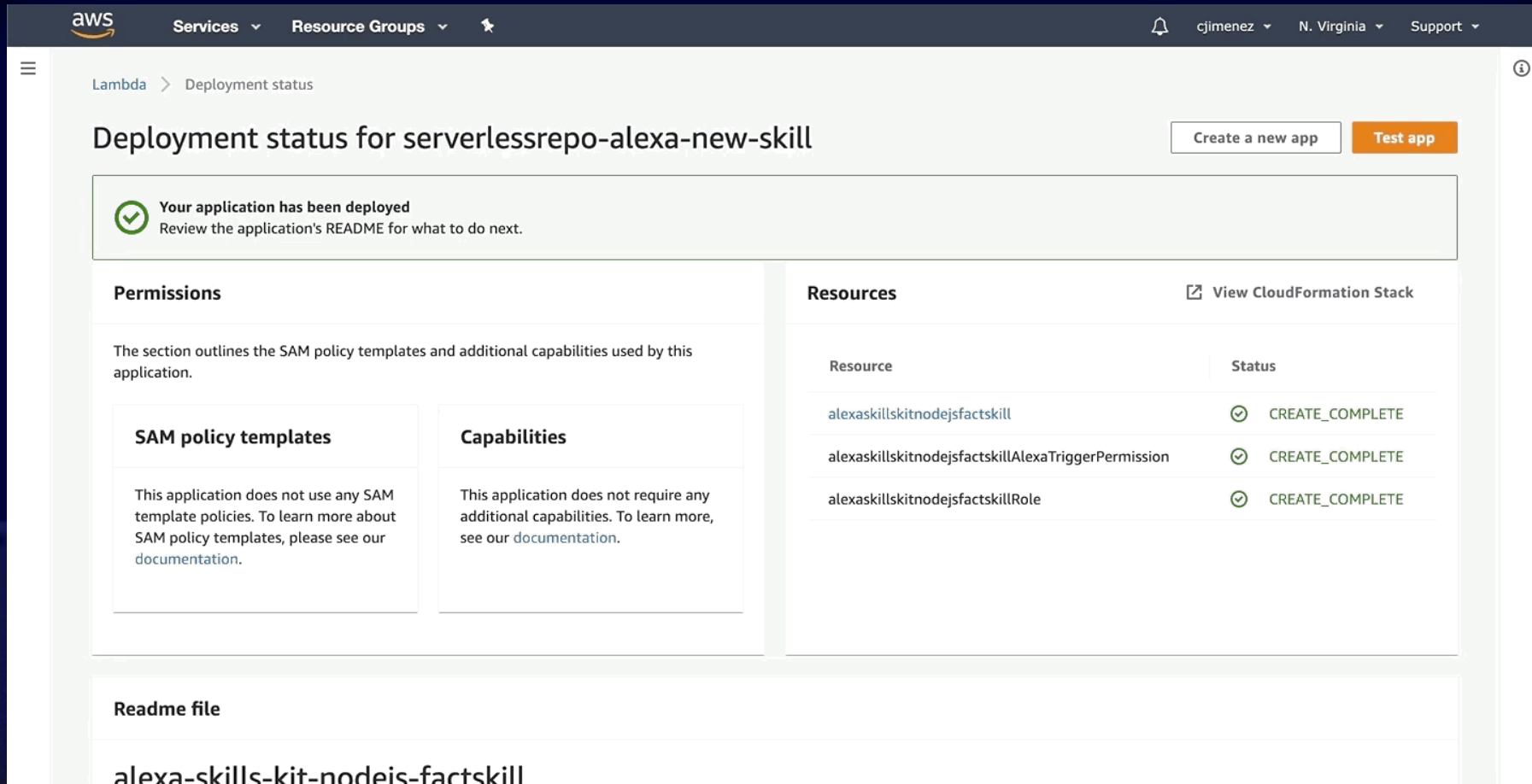
Resource	Value
Lambda function(s)	2
Code storage	157.1 kB (0% of 75.0 GB)
Full account concurrency	1000
Unreserved account concurrency	1000

A 'Create function' button is visible below the resource summary. Below this, the 'Account-level metrics' section includes a descriptive text: 'The charts below show metrics across all your Lambda functions in this AWS Region.' The metrics are presented in three panels, each with a 'No data available' message and a suggestion to 'Try adjusting the dashboard time range.' The panels are:

- Error count and success rate (%)**: Shows 1 error and 100% success rate.
- Throttles**: Shows 1 throttle.
- Invocations**: Shows 1 invocation.

Each panel also features a 'Try adjusting the dashboard time range' message. The top right of the metrics section includes an 'Add to dashboard' button and a time range selector (1h, 3h, 12h, 1d, 3d, 1w, custom) with a refresh and dropdown icon.

Associate the Repository to Your Skill



The screenshot displays the AWS Lambda console interface for a deployment named "serverlessrepo-alexa-new-skill". At the top, the AWS logo and navigation menu are visible. The main heading is "Deployment status for serverlessrepo-alexa-new-skill", with buttons for "Create a new app" and "Test app". A green checkmark icon indicates that the application has been successfully deployed, with a message: "Your application has been deployed. Review the application's README for what to do next."

The "Permissions" section explains that the application uses SAM policy templates and additional capabilities. It is divided into two sub-sections:

- 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](#).

The "Resources" section, with a "View CloudFormation Stack" link, contains a table listing the resources used by the application:

Resource	Status
alexaskillskitnodejsfactskill	✔ CREATE_COMPLETE
alexaskillskitnodejsfactskillAlexaTriggerPermission	✔ CREATE_COMPLETE
alexaskillskitnodejsfactskillRole	✔ CREATE_COMPLETE

Below the resources table, there is a "Readme file" section with the text "alexa-skills-kit-nodejs-factskill".

Paste your generated code

The screenshot shows the Alexa Developer Console interface. At the top, there's a navigation bar with tabs for 'Your Skills', 'My New Skill', 'Build', 'Code', 'Test', 'Distribution', 'Certification', and 'Analytics'. The 'Build' tab is active. Below the navigation bar, there's a language selector set to 'English (US)' and three buttons: 'Save Endpoints', 'View Model Versions', and 'Update live skill'. The left sidebar contains a 'CUSTOM' section with 'Interaction Model' selected, and a list of intents including 'HelloWorldIntent', 'newIntent', 'library', and 'Built-In Intents (4)'. The main content area is titled 'Endpoint' and contains a lightbulb icon with the text: 'The Endpoint will receive POST requests when a user interacts with your Alexa Skill. The request body contains parameters that your service can use to perform logic and generate a JSON-formatted response. Learn more about AWS Lambda endpoints here. You can host your own HTTPS web service endpoint as long as the service meets the requirements described here.' Below this is the 'Service Endpoint Type' section, which is set to 'AWS Lambda ARN (Recommended)'. It shows the 'Your Skill ID' as 'amzn1.ask.skill.ec86bc12-fd26-4d2e-8ebd-041bee75766a' with a 'Copy to Clipboard' button. The 'Default Region (Required)' is set to 'arn:aws:lambda:us-east-1:815919558687:function:serverlessrepo-alexa-ne'. The 'North America (Optional)' field is empty, showing a placeholder: 'arn:aws:lambda:us-east-1:<aws_account_id>:function:<lambda_name>'. The background of the slide features a close-up of an Alexa device's top surface.

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



Custom Code: HTTPGET

```
4 function httpGet(param) {
5   return new Promise(((resolve, reject) => {
6     var options = {
7       host: 'libanswers.fiu.edu',
8       path: param,
9       method: 'GET',
10      rejectUnauthorized: false
11    };
12
13    const request = https.request(options, (response) => {
14      response.setEncoding('utf8');
15      let returnData = '';
16
17      response.on('data', (chunk) => {
18        returnData += chunk;
19      });
20
21      response.on('end', () => {
22        resolve(JSON.parse(returnData));
23      });
24
25      response.on('error', (error) => {
26        reject(error);
27      });
28    });
29    request.end();
30  }));
31 }
```


Customize Node.js: Intent Overview

```
35
36 //SKIPPING CUSTOM INTENT HANDLERS
37 //EXAMPLE: IF A USER ASKS HOW TO PRINT, THIS HANDLER WILL RUN.
38
39 ▼ const howprint_Handler = {
40 ▼   canHandle(handlerInput) {
41     const request = handlerInput.requestEnvelope.request;
42     return request.type === 'IntentRequest' && request.intent.name === 'howprint';
43   },
44 ▼   async handle(handlerInput) {
45     const request = handlerInput.requestEnvelope.request;
46     const responseBuilder = handlerInput.responseBuilder;
47     let sessionAttributes = handlerInput.attributesManager.getSessionAttributes();
48
49     //GREETING THAT GIVES "SAY" A VALUE
50     let say = 'Hello from ' + invocationName + ', your F. I. U. Libraries voice assistant.';
51
52     //RETRIEVES THE ANSWER FROM YOUR INSTANCE OF LIBANSWERS
53     let answerId = /* ENTER THE ANSWER ID FROM YOUR SYSTEM */;
54
55     //INSPECT YOUR URL STRUCTURE TO MAKE SURE "PARAM" WILL BUILD THE URL CORRECTLY. IT IS YOUR "HOST" + "API VERSION" + "ANSWER ID" + "GROUP ID"
56     //MODIFY "PARAM" WITH YOUR GROUP ID & LIBANSWERS SYSTEM ID (iid)
57     let param = '/1.0/faqs/' + answerId + '?group_id=000&iid=000';
58     let slotStatus = '';
59     let resolvedSlot;
60
61     //SENDS THE REQUEST TO YOUR HTTPGET FUNCTION AND WAITS FOR THE PARSED JSON DATA
62     const response = await httpGet(param);
63     console.log(response);
64     say += slotStatus;
65
66     //USE DOT NOTATION TO FEED DATA INTO YOUR SKILL
67     return responseBuilder
68       .speak(response.faqs[0].answer)
69
70     //THE STANDARD CARD DISPLAYS TEXT ON ECHO SHOW SCREENS. I USED A "stripHtml" FUNCTION TO REMOVE HTML TAGS BEFORE DISPLAYING ON THE SCREEN.
71     .withStandardCard(stripHtml(response.faqs[0].question),
72       '\n' + stripHtml(response.faqs[0].answer),
73       welcomeCardImg.smallImageUrl, welcomeCardImg.largeImageUrl
74     )
75     .withShouldEndSession(true)
76     .getResponse();
77   },
78 };
```

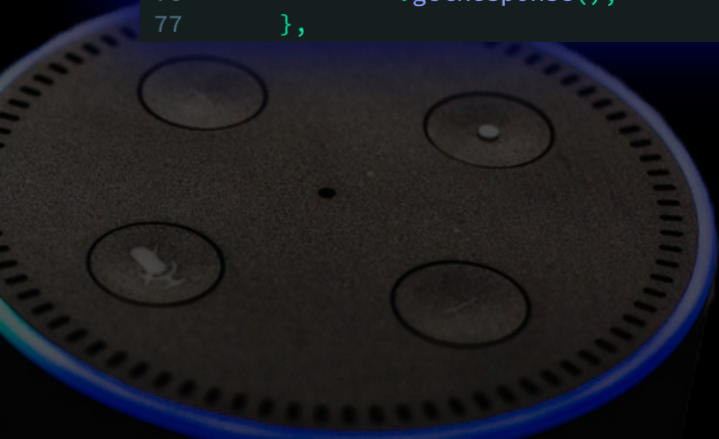
Customize Node.js: param

```
51
52 //RETRIEVES THE ANSWER FROM YOUR INSTANCE OF LIBANSWERS
53 let answerId = /* ENTER THE ANSWER ID FROM YOUR SYSTEM */;
54
55 //INSPECT YOUR URL STRUCTURE TO MAKE SURE "PARAM" WILL BUILD THE URL CORRECTLY. IT IS YOUR "HOST" + "API VERSION" + "ANSWER ID" + "GROUP ID"
56 //MODIFY "PARAM" WITH YOUR GROUP ID & LIBANSWERS SYSTEM ID (iid)
57 let param = '/1.0/faqs/' + answerId + '?group_id=000&iid=000';
58 let slotStatus = '';
59 let resolvedSlot;
60
61 //SENDS THE REQUEST TO YOUR HTTPGET FUNCTION AND WAITS FOR THE PARSED JSON DATA
62 const response = await httpGet(param);
63 console.log(response);
64 say += slotStatus;
65
```



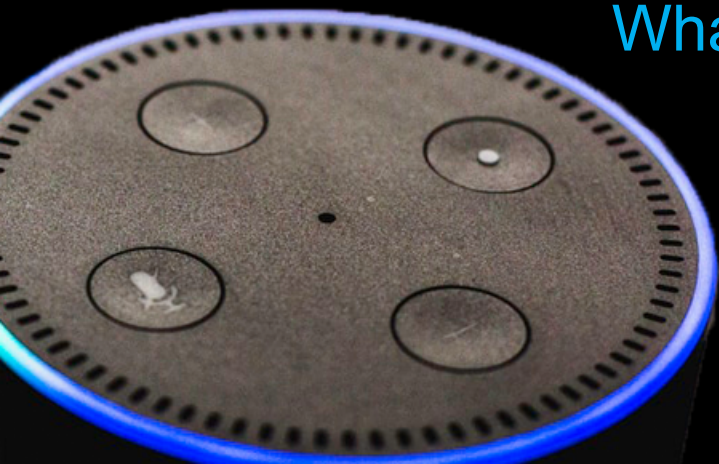
Customize Node.js: responseBuilder

```
65
66 //USE DOT NOTATION TO FEED DATA INTO YOUR SKILL
67 return responseBuilder
68     .speak(response.faqs[0].answer)
69
70 //THE STANDARD CARD DISPLAYS TEXT ON ECHO SHOW SCREENS. I USED A "stripHtml" FUNCTION TO REMOVE HTML TAGS BEFORE DISPLAYING ON THE SCREEN.
71     .withStandardCard(stripHtml(response.faqs[0].question),
72         '\n' + stripHtml(response.faqs[0].answer),
73         welcomeCardImg.smallImageUrl, welcomeCardImg.largeImageUrl
74     )
75     .withShouldEndSession(true)
76     .getResponse();
77 },
```



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.

Hennig, N. (2018). *Siri, Alexa, and other digital assistants : the librarian's quick guide*.

The Infinite Dial 2020 - Edison Research. (2020). Retrieved from <https://www.edisonresearch.com/the-infinite-dial-2020/>

Miller, R., & Washington, K. (2017). *Consumer Use of the Internet and Mobile Web 2020-2021*. Retrieved from <http://www.rkma.com.ezproxy.library.wisc.edu/wisc/cuim16/>

NPR. (2020). *The Smart Audio Report*. Retrieved from <https://www.nationalpublicmedia.com/insights/reports/smart-audio-report/>



Alexa, are there any questions?



Christopher M. Jimenez

Web Services Librarian
Florida International University



Christopher.Jimenez@fiu.edu



[@cjmnz8](https://twitter.com/cjmnz8)



github.com/cjmnz8/Echo-Show-Tell

