Voice Based Virtual Library Assistant

Course Number: CSCI 59000 – 25096

Course Name: Advanced Mobility & Cloud Computing
Course Project

Akhil Nayabu
anayabu@iupui.edu

Table of Contents

Overview:

In recent times there has been significant growth in areas of machine learning, artificial intelligence and natural language processing. Due to such significant growth these areas, we have lot API's that have opened up for community developers to develop apps leveraging the capabilities of AI, natural language understanding. As part of this project we would be building a voice intelligent virtual assistant to serve as a customer service representative in a college library.

Why Amazon Alexa?

Amazon has launched Amazon echo / dot in recent times which serves the consumer's as a smart speaker and voice enabled intelligent assistant. Amazon has provided the developers with a platform where the developer can leverage their API's to build such voice intelligent virtual assistants. In the space of voice intelligent assistants, Amazon has been in this space longer than its competitors like Google (Google Home). Amazon has opened up the platform and SDK for the developers to build these apps (called as skill in Amazon's jargon). Amazon has good community forums and Amazon skill kit looks more mature and a stable API.

For the implementation of this voice intelligent virtual assistant we have chosen Amazon Skill Kit (ASK) and their node.js SDK as the technology stack.

Use case:

This use case helps to enhance the customer service experience at library help desks. Library front desk staff spent of their valuable time answering more trivial questions like library hours, find a book, computer availability and study room reservations. These are very simple question which can be answered without much of a hassle by just doing a search on the repository. This implementation help to answer such basic questions in form of voice enabled library assistants. This use case helps the library staff to focus on much more critical and complicated question to solve rather spending man hours on answering such question.

In our use case the library virtual assistant can be made available 24/7 answer questions limited to below:

- Knowing library hours
- Finding a book
- Computer availability (Future enhancement)
- Study room reservation (Future enhancement)

All of these above information can be easily answered by just performing a simple lookup on the database. Amazon Alexa helps to understand various slangs and various utterances. Amazon Alexa skill can understand what the user speaks by performing natural language classification and based on the intent of the user, this application can return a response to the user.

How Alexa Works?

For an Alexa skill app, we have two components that are involved in developing the skills. They are:

- Amazon Alexa Skill Platform
- Amazon Skill User's Application service.

Amazon Alexa Skill Platform:

In the Amazon Alexa Skill platform, we define the interaction model of your skill. Interaction model is analogous to graphical user interface (GUI) in any traditional mobile or web app. Here in this interaction model we define what operations / requests can the user perform and its user inputs. Also, we specify what all possible sample utterances that a user can use to perform that request.

User's Application Service:

In the Alexa Skill Platform, we configure the back-end data application service which performs the most of the business logic, gets the data from database and converts it into readable English text so that the Amazon Echo / other speaking devices can read. This back-end application service can be hosted in your own server or can be hosted in AWS Lambda platform. (AWS Lambda is a platform that Amazon offers to run your code without having to manage or buy are server. It is termed as Serverless Compute).

This application service must be implemented using Alexa SDK. This application service handles the requests that are routed from the skills platform. The application service receives a JSON request and this JSON request contains the intent the user is trying to perform and its user inputs (also called as slots). Based on the intent and the inputs, as specific event handler method in the application service is invoked. In these methods, actual data is fetched and converted into English readable text and wrapped in SSML tags and returned to the skills platform in form of JSON response.

Flow:

1. When a user speaks to an echo or a dot, the speech is streamed onto the Alexa Skills platform present in AWS.
2. Then this Skills platform will parse through the voice text and using some natural language classifiers and deep learning, it identifies what the user is trying to ask or perform. Based on the intents defined for this skill and their sample utterances, the skills platform will use these deep learning techniques to associate the user voice stream to specific intent.
3. Once the user intent and input are identified, then the request is forwarded to the application service.
4. When the request reaches the application service based on the intent in the request a specific event handler method in the application service is triggered and the response is returned back to the skill platform.
5. And the skills platform sends back the response to the actual device. Based on this response the device speaks the output.
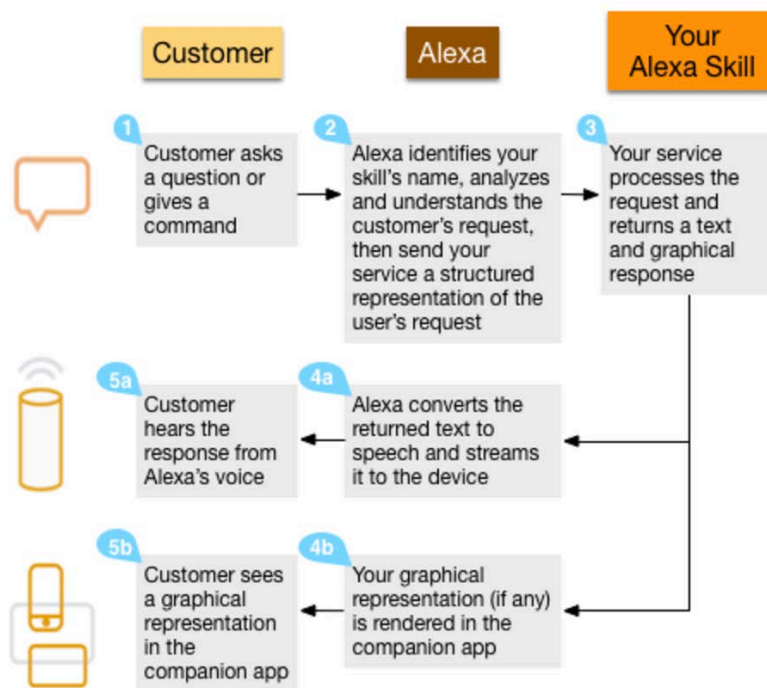
Interaction Flow

The Speech recognition and speech to intent method mapping is taken by the Amazon's Alexa skill platform itself. The developers need to worry on about the application service where we write code for our business logic and fetching data. Although the interaction model consisting of intent definitions and sample utterances need to be provided by the user itself.

Implementation:

Before we discuss about implementation of the project, let us look at terminology that we need to know in Alexa skill kit SDK.

Intent: An intent is something that user is wants to do. These intents are used to tell lambda function what to do. It is an action which will fulfill the user's request. They can have arguments called as slots which is discussed later.

- **Built in Intents**: These are the intents for common actions that you can choose to implement in your custom skill without providing any sample utterances. **Examples:**
  AMAZON.HelpIntent
  AMAZON.StopIntent

  Like said these intents do not need any sample utterances to be invoked.

Intents JSON	Code to be implemented for these Intents

- **Custom Intents**: These are the intents for which you need mention what are the sample utterances so that these intents will only trigger when these utterances are made. **Examples**:
  From our project:

Intents JSON	Sample Utterances for these Intents

Slots: slots are the types which define how data is stored is recognized and handled. We can both custom and built in slots:

- **Built in slots:** Slot types which are already defined by amazon. These built in slot types usually fall into 2 categories : **Numbers, Dates, and Times** , **List of Items**
- **Custom Slot types:** It defines a list of values for a slot. They are used for lists of items are not in built in slots provided by amazon.

As stated before slots are the arguments in user's intent.
Example:
LibraryHoursIntent What are the library hours for {date} => {date} is the slot.

```
{
  "slots": [
    {
      "name": "date",
      "type": "AMAZON.DATE"
    }
  ],
  "intent": "LibraryHoursIntent"
},
```

       Slot for LibraryHoursIntent

Utterances: Utterances are the phrases which people use to request something. These phrases will vary, since we can ask same question n-possible ways as every user might not request in the same way.

Sample Utterances in project:
What are the library hours
what are the library hours for tomorrow?
When does the library Wednesday?

LibraryHoursSemesterIntent When does the library open in {semester} semester

          Intent   Custom Slot
                            Utterance
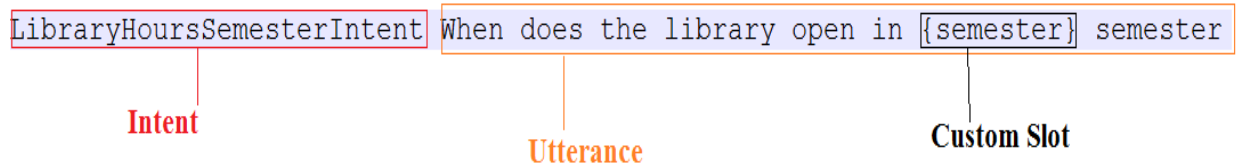
Each of these Intents will be mapped to a function in Node.js where it will process the request and send appropriate response to the user.

```javascript
"LibraryHoursIntent": function() {
    logger(appLogLevel, "INFO", "LibraryHoursIntent", JSON.stringify(this.event.request));
    var day;
    var date;

    if( this.event.request.intent.slots.date.value !== undefined ){
        day =  this.event.request.intent.slots.date.value;
        date = new Date(day)
    }
    else{
        date = new Date();
    }

    var hrsObj = getLibraryHoursByDay(date.getDay());

    var speechOutput = buildLibHrsByDayResponse(hrsObj);

    this.attributes['previousContext'] = 'CONTINUE';
    this.attributes["speechOutput"] = speechOutput;
    this.attributes["repromptText"] = speechOutput;

    this.emit(":ask", speechOutput, speechOutput);

},
```

So each time LibraryHoursIntent is triggered the above written function will be invoked. When user asks for library hours, then this function is called, it checks if the user has mentioned any day or date, if he did it will return library hours for the mentioned day or will return the library hours for current day. Similarly each intent has its functioned mapped to it. This script is deployed in AWS lambda.

Deployment:

To have Alexa skill working we need to have both the components on Alexa Skill platform and User application service to be fully configured and in running state.

Below are the detailed steps for the setup / deployment:

Application Service: (For our project we have chosen AWS lambda as the platform where will be hosting our application service).
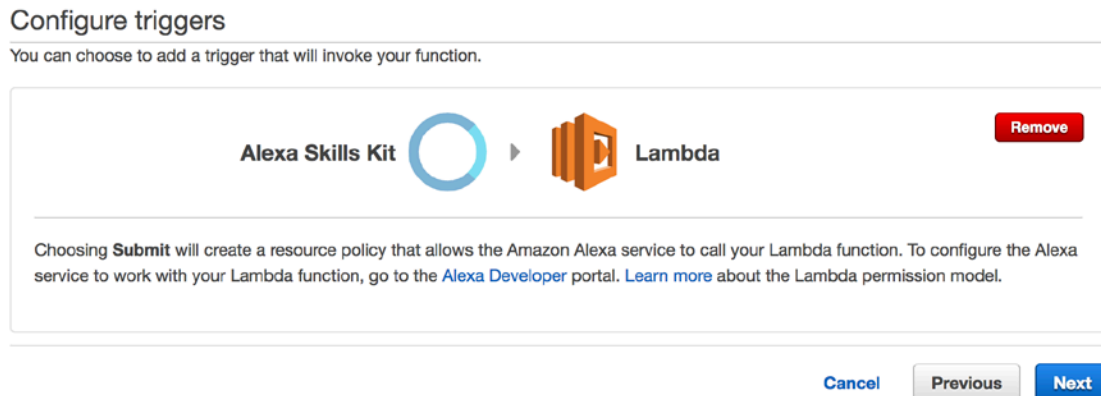
1. Setup an AWS developer account.
2. Once you are logged into the AWS console, select Lambda from the services.
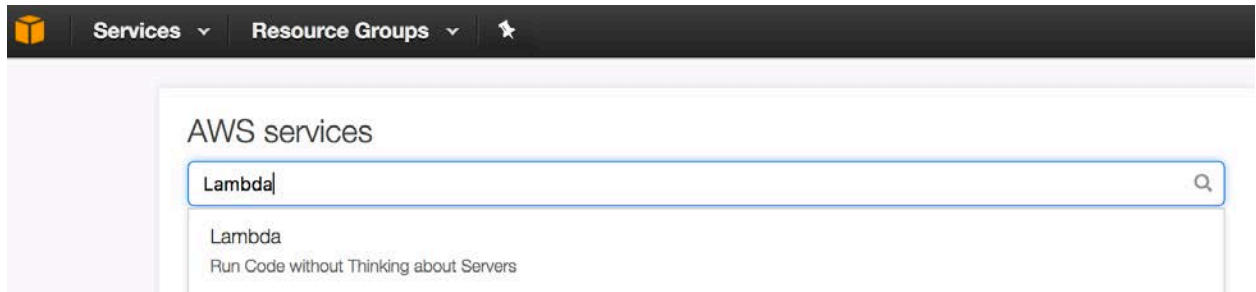
3. Once you land on the AWS Lambda home page, choose to create a new lambda function.
4. Choose the blue print as "Blank function"
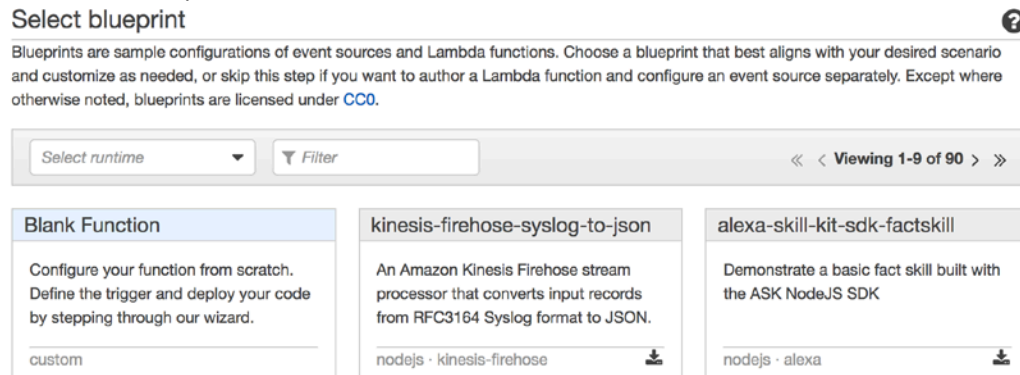
5. Choose the trigger to invoke the Lambda function as "Alexa Skills kit"

Note: Choose the AWS region as "N.Virginia". Alexa Skills Kit is currently available only here.

6. Enter the Lambda function name and choose the runtime as Node.js 4.3
7. Move the code from index.js from the project code files into the text box given in AWS.

Configure function

A Lambda function consists of the custom code you want to execute. Learn more about Lambda functions.

Name*	*myFunctionName*
Description	
Runtime*	Node.js 6.10 ▼

Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. Learn more about deploying Lambda functions.

Code entry type   Edit code inline ▼

```
1   exports.handler = (event, context, callback) => {
2       // TODO implement
3       callback(null, 'Hello from Lambda');
4   };
```

8. Create a custom role with name as lambda_basic_execution and click next.
9. Once we review the configurations click create function.
10. Once the lambda function is created, make a note of the Amazon Resource Number (ARN) so that this can be used to configure in the Alexa Skills platform. This ARN serves as web service endpoint.

Alexa Skills Platform: In Alexa Skill Developer platform, we configure the interaction model for our skill.

Below are the detailed steps for configuring the skill on this platform.

1. Log into Amazon Skill developer platform at https://developer.amazon.com
2. Choose Alexa Skills Kit from Alexa tab,
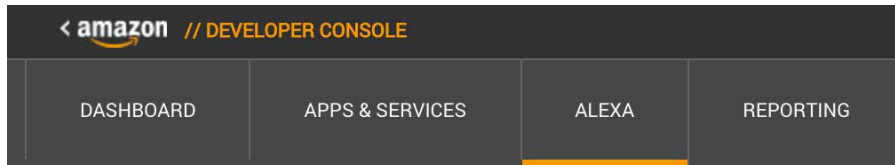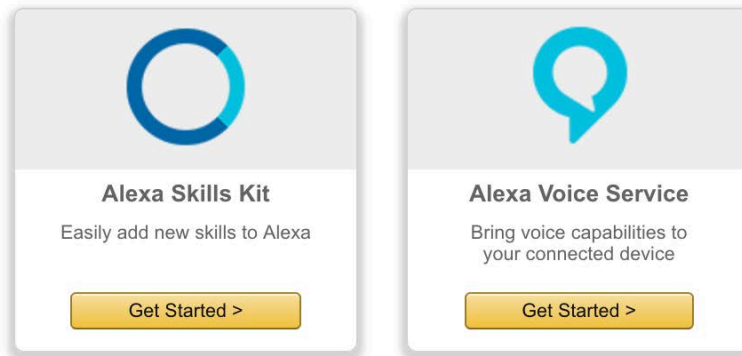
3. Choose Add new skill option
4. Configure the Skill information in step-1.
   Choose the skill type as "Custom Skill"
   Give a name to the skill, this name would be used to display the name on the Alexa Skill app store.
   Choose the invocation name. This phrase would be used to launch the skill on a Echo / Echo dot.

5. Configure all the Intent schemas. Here we list all the possible intents that the skill supports.

Copy the intent schema from skill_assests/library-assistant-intents.json into the Intent Schema text box.

6. In our skill, we use a custom slot to hold the semester name. This is like a user defined data type in other programming languages. Configure the custom slot as below:

7. Provide the sample utterances in the text box provided. This sample utterance is used to train the model for voice recognition and intent resolution.

Sample utterances list can be found in skill_assets/library-assistant-sample-utterances.txt

8. Once the interaction model is built, we need to configure your user's application data service.

For our project, we have hosted our application service in AWS lambda so configure the ARN and choose North America.

9. Once the configuration is complete, we are ready to test.

10. The Alexa skill can be tested even in https://echosim.io

Sample Interaction Flow:

Alexa, Jaguar Library

My name is Jinx. I am your IUPUI Library Virtual Assistant. I can help you with knowing Library hours, and Finding books

What are the library hours for fall semester ?

Library hours for fall semester are Monday to Friday 7:30 AM to 12:00 AM, Saturday 8:00 AM to 7:00 PM and Sunday 10:30 AM to 12:00 AM. Is there any thing else I can help you with ?

Help me find a book

Let's begin the book search..... What is the book name you are trying to find ?

Book name is Introduction to Java Programming

Who is the author of the book ?

Author is John Davis

I heard book name as **Introduction to Java Programming** and author as **John Davis**. Is that correct ?

Yes, That's correct.

Introduction to Java Programming : 2nd Edition written by John Davis is available in Indianapolis library, 2nd floor row number 24.
Is there any thing I can help you with ?

No.

Thank you for visiting. Good Bye !!!

References:

- https://developer.amazon.com/alexa-skills-kit#learn

- https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/alexa-skills-kit-interaction-model-reference

- https://www.slideshare.net/AmazonWebServices/mbl310-alexa-voice-service-under-the-hood

- http://mw17.mwconf.org/paper/a-case-study-on-using-voice-technology-to-assist-the-museum-visitor/