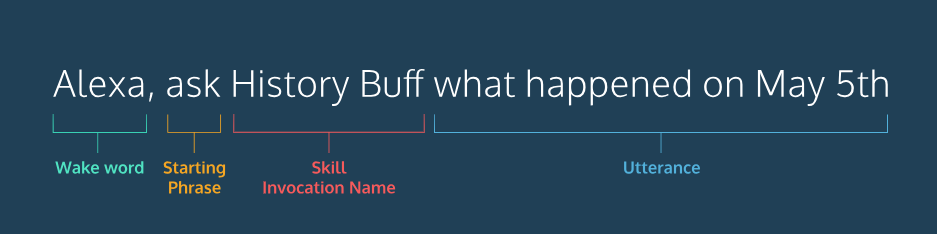
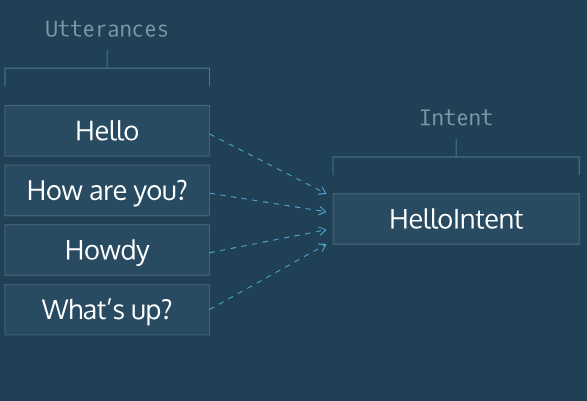
**Build your first Alexa Skill**

* Alexa is the cloud based service that handles all the speech recognition, machine learning, and natural language understanding for Alexa enabled devices like Amazon Echo, Echo Dot, Amazon Tap and Echo Show.
* Amazon Resource Name(ARN) is the unique identifier we use to connect our Lambda function to our Voice User Interface.
* You can test your skills using the browser based simulator – echoism.io



Utterances help Alexa connect the intents to phrases spoken by the user. In this case, these sample utterances will help Alexa map the spoken user input to our "HelloIntent".



단계 1. Create the interaction model, also called the Voice User Interface(VUI)

* Skill Information에 들어가서, Skill Name을 적고, Skill Model은 Custom으로 설정
* Create Skill 클릭
* 그러면 Skill Builder로 넘어감.
* Interaction Model => Invocation
* Skill Invocation Name 설정
* Save Model 클릭
* Intents에서 Add 버튼을 눌러서, Intent의 명령어를 적고(HelloIntent) 엔터를 누른다. 혹은 Create Custom Intent button을 누른다.
* Utterance를 입력하고 나서, Save Model => Build Model

단계 2. Connect Lambda Function(Backend)

* To connect the frontend to a Lambda function(the backend) you must provide its Amazon Resource Name(ARN), which is a unique identifier that represents our Lambda function.
* Build tab에서 Endpoint section => AWS Lambda ARN as the Service Endpoint Type.
* ARN을 Default Region text box에 넣는다
* Save Endpoints.

단계 3. Use Echoism.io

* Log in with your Amazon Account(connected to your Amazon developer account)
* click & hold the microphone button( or hold down the space bar on your keyboard) and say “Open Code Academy”
* Now say “Alexa, tell code academy to say hello”

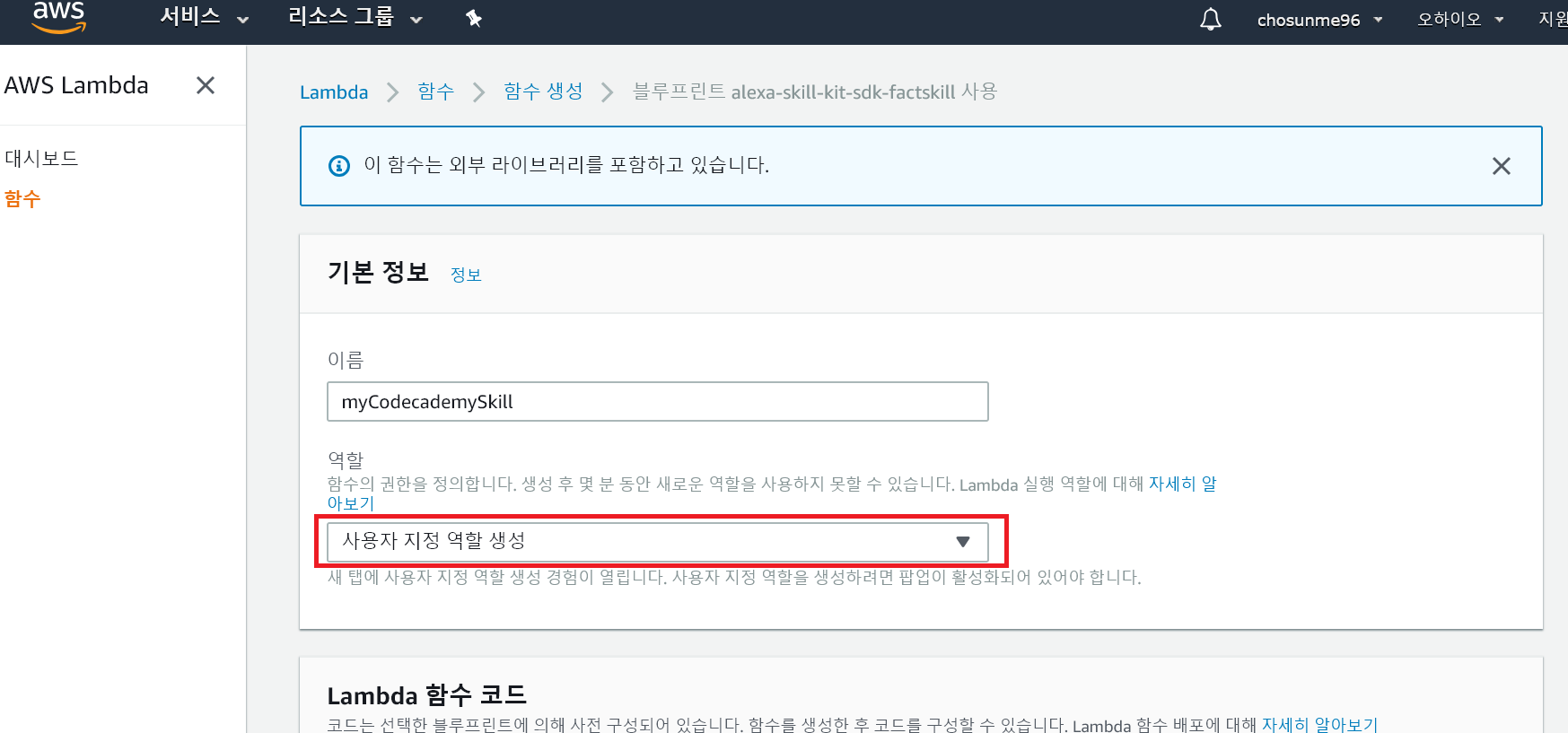
단계 4. Using the Service Simulator

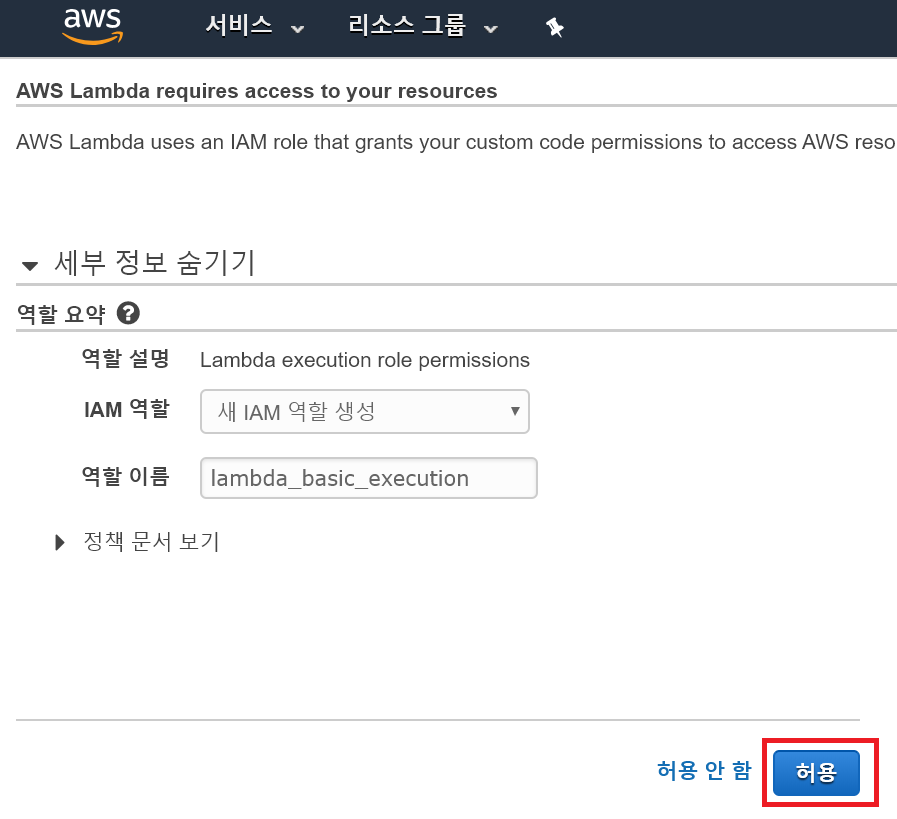
* Test 탭에 들어간다.
* toggle button을 눌러서 enable testing
* Type “tell code academy hello” in the box
* Enter

**Create Your Lambda Function**

단계 1. Create Lambda Function

* Create Function -> Choose Blueprint
* Type Alexa in the search/filter box, and hit enter.
* Choose Blueprint named alexa-skill-kit-sdk-factskill.
* Give your function a Name that you can remember what it does. (Descriptive)
* Set appropriate permission for Lambda function by choosing an IAM role. (Identity and Access Management)
* 특정한 게 없다면 click on Existing Role drop-down and select “Lambda\_basic\_execution”





* 그 다음에는 ‘함수 생성’

단계 2. Triggers and Review Lambda Function

* Click on Triggers tab
* Click on + Add Trigger
* Click on the dashed box and select Alexa Skills Kit from the list

"use strict";

var Alexa = require("alexa-sdk");

var handlers = {

"HelloIntent": function () {

this.response.speak("Hello, Seon Mi");

this.emit(":responseReady");

},

"myFavoriteLanguageIntent" : function(){

this.emit(":tell", "Hey Seon Mi, long time no talk. Your favorite language is Javascript. Bye!");

},

"LaunchRequest": function () {

this.response.speak("Seon Mi, Welcome to Codecademy");

this.emit(":responseReady");

}

};

exports.handler = function(event, context, callback){

var alexa = Alexa.handler(event, context);

alexa.registerHandlers(handlers);

alexa.execute();

};

**Slots**

단계 1. Create a Slot

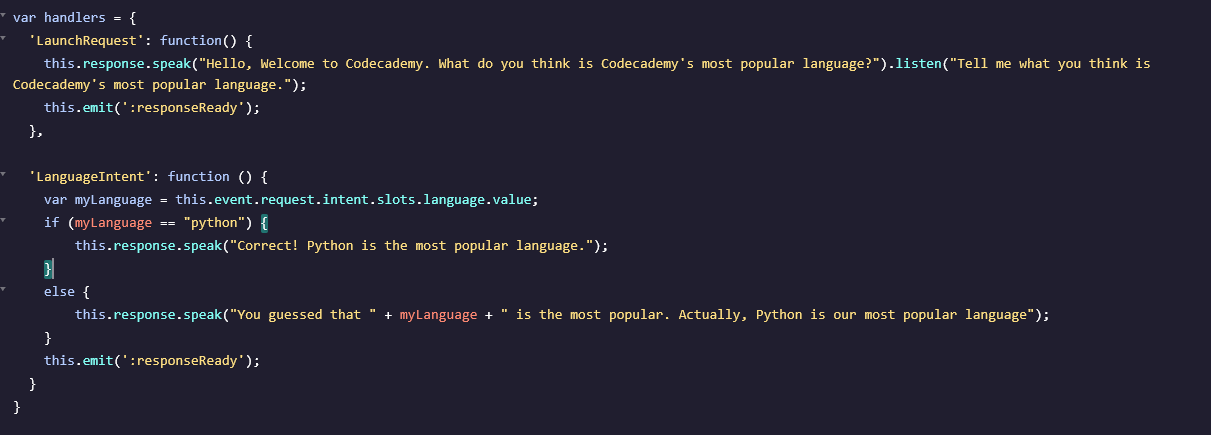
* Interaction Model : Slot Types
* There are Built-in Slots & Custom Slots
* Build 창에서 Slot types 선택해서, Add + 누르기

단계 2. Create an Intent & Connect the Slot

* Create a new Intent called LanguageIntent
* Add “Is it {language}” as a sample utterance to the LanguageIntent and press enter
* On the right side, and under Intent Slots, select “Choose a slot type” and select programmingLangugageType
* Add more utterances. “Could it be {language}”, I think it’s {language}

단계 3. Read a Slot that is inputted

* this.event.request.intent.slots.yourSlotName.value
* 이번 예시에서는 this.event.request.intent.slots.language.value가 된다.



**Add Persistence to Your Skill**

* Add a notion of short-term memory
* Add an option for users to choose
* use Session Attributes to remember and update user-provided details throughout a session.

Session Attributes