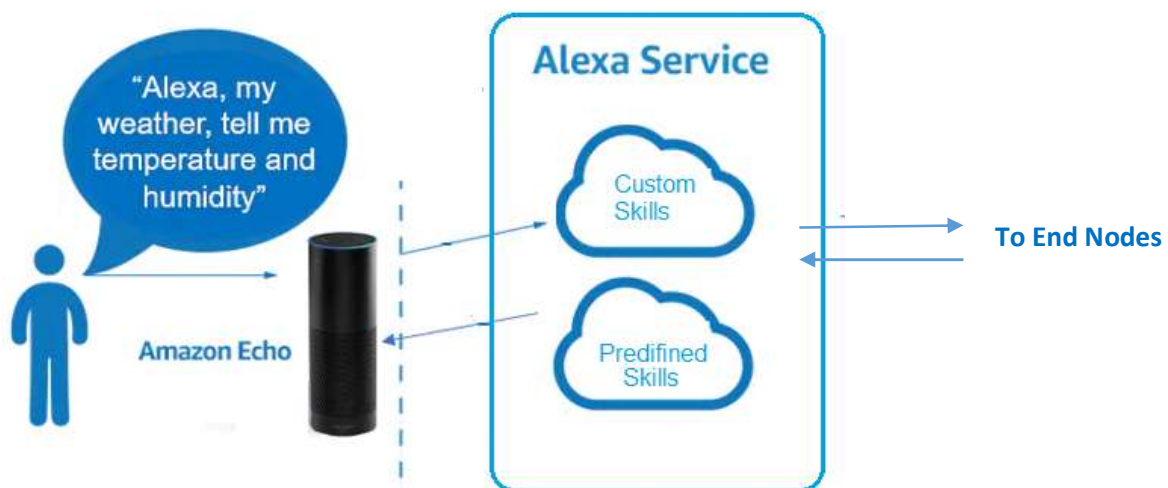


Voice Controlled Smart Devices with Amazon Alexa & AWS Lambda – part 1

1- Alexa Service & Alexa Skills

The Alexa Voice Service (AVS) allows developers to enable voice interactions on connected products.



A user, with an Alexa Service Account, can host many Skills. These skills can be predefined (Home Automation skill, Movies Skill, etc..) or Custom skills, developed by the user.

Skills are like apps that help to define interaction between user (speech) and a device.

A skill is *triggered* by an “**Skill Invocation name**”.

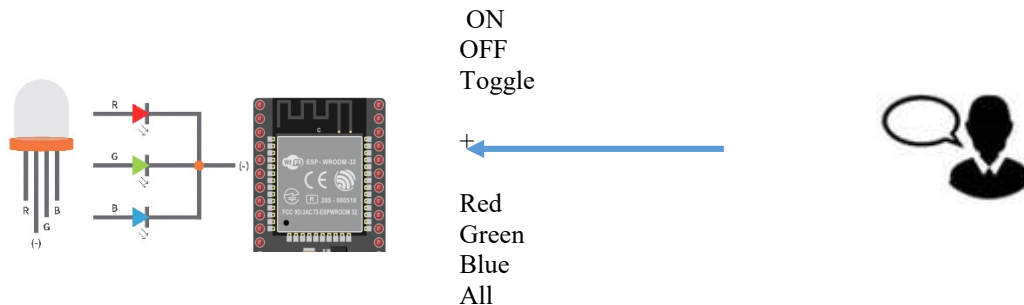
A Skill is composed of a set on **Intents**, each **Intent** represents an action that fulfills a user's spoken request. For example, in a Home Automation Skill, an Intent can be used to control the home lights, another to control the alarm system, another for the TV, etc.

For each Intent, we define a set of likely spoken phrases that include as many representative phrases as possible: **sample utterances**.

Each Utterance (of an Intent) contains generally one or many **Slots**. Each Slot can have a value among a set of values (Slot Type). These values are the words that define the instruction sent to the device.

2- System Description

In this project, a Smart Device is controlled through Voice commands. More precisely, an RGB Led (Connected to an SoC such as ESP32) is controlled in such a way to **turn on**, **turn off** or **toggle** one or all of the **3 Leds** (Red, Blue and Green).



Obviously, this example could be applied to any application controlling Lights in a home. Just the words (Red, Green, Blue) will be replaced by appropriate words (bedroom light for example, etc..)

3- Solution Architecture

The Alexa voice control solution is composed of:

- **The Alexa echo dot** (optional) : it processes the spoken sentence and sends it to the Amazon Alexa Service.
- Three “Cloud Hosted Services”. The Amazon Alexa Service, the AWS Lambda Service and A MQTT Broker.



- **The amazon Alexa Service**: It identifies the Skill to be triggered, detects the Intent and the slots contained in the spoken sentence (after processing by Alexa).

When an Intent is detected, the Amazon Alexa Service builds a JSON formatted data and sends it to a web service endpoint to handle the skill request. In case of a sentence containing the words (slot values) green and on, the json data has the following format

```
{
  "request": {
    "type": "IntentRequest",
    "requestId": "amzn1.echo-api.request.cdb195cb-3741-4...",
    "locale": "en-GB",
    "timestamp": "2021-11-08T12:45:41Z",
    "intent": {
      "name": "rgbled",
      "confirmationStatus": "NONE",
      "slots": {
        "rgbstate": {
          "name": "rgbstate",
          "value": "on",
        },
        "rgbcolor": {
          "name": "rgbcolor",
          "value": "green",
        }
      }
    }
  }
}
```

The AWS Lambda Service:

AWS Lambda is a [serverless compute](#) service that runs your code in response to events and automatically manages the underlying compute resources for you. These events may include changes in state or an update.

In this example, a Lambda Function (Node JS) will be deployed : it receives the JSON formatted data and publishes mqtt messages to a MQTT Broker.

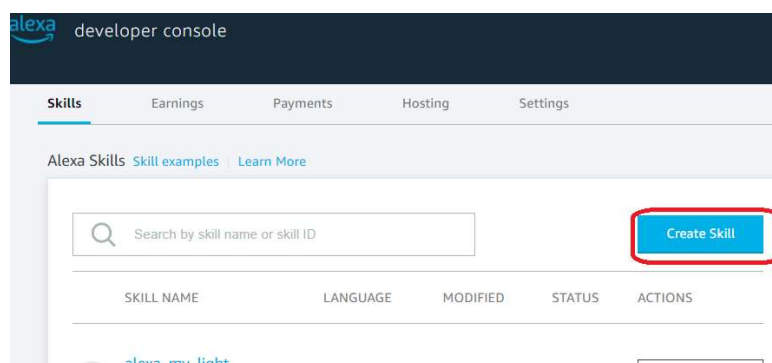
The MQTT Broker:

It receives the published messages and sends them to the subscribed devices. In this case, the SoC that controls the RGB leds.

4- [Solution Implementation on AWS](#)

4-1 Skill creation

- 1- Create an [Amazon Account](#) to access Alexa Developer Console.
- 2- Click on [Create Skill](#),



- 3- type the skill name , choose the language (English), the Model (Custom) and the programming language (NodeJS) and click **“Create Skill”**

Skill name

mydevice **1**

8/50 characters

5

Cancel Create skill

Model: Custom
Host: Alexa-hosted (Node.js)
Hosting Region: EU (Ireland) ▼

Primary locale

A locale refers to a language and the location (country) in which its spoken. Your primary locale is what you will start building your skill in. You can add locales after your skill is created.

English (UK) **2**

Sync locales ☐

Sync all locales with the same language to the Primary locale. Changes you make to your skill in the Primary locale automatically propagate to all other locales of the same language. You can manage these settings or turn off this feature anytime in Language settings. [Learn more](#)

1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

| | | |
|--|---|--|
| Custom | Flash Briefing | Smart Home |
| Design a unique experience for your users. A custom model enables you to create all of your skill's interactions. 3 | Give users control of their news feed. This pre-built model lets users control what updates they listen to. "Alexa, what's in the news?" | Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up. "Alexa, turn on the kitchen lights" |

2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

| | | |
|---|---|---|
| Alexa-hosted (Node.js) | Alexa-hosted (Python) | Provision your own |
| Alexa will host skills in your account and get you started with a Node.js template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. Learn more 4 | Alexa will host skills in your account and get you started with a Python template. You will gain access to AWS Lambda endpoints in all Alexa service regions, a DynamoDB table for data persistence, and S3 for media storage. Learn more | Provision your own endpoint and backend resources for your skill. This is recommended for skills that have significant data transfer requirements. You will not gain access to the console's code editor. |

4- Choose « Start From Scratch » and click on **Continue with template**.

In fact, there are an important number of existing skills (Home domotocs skill, movies skill, etc...) that can be used directly

Choose a template to add to your skill

Select a skill template from the list below or import a skill shared by the Alexa community as a public Git repository.

[Import skill](#)[Continue with template](#)

Start from Scratch

This skill gets you started with the required intents and with code demonstrating "Hello World" functionality if you are building an Alexa-hosted skill.

[Learn more](#)

By [Alexa](#)

Fact Skill

Build an engaging fact skill about any topic. Alexa will select a fact at random and share it with the user when the skill is invoked. [Learn more](#)

Includes: custom intents, Personalization

By [Alexa](#)

High-Low Game Skill

Try to guess a target number in a given range and Alexa will tell you if the number she had in mind was higher or lower. [Learn more](#)

Includes: slots, custom intents, data persistence

By [Alexa](#)

Pet Tales Skill

Build a compelling multi-turn conversational audio and visual e for a user looking for her favorite [Learn more](#)

Includes: APL for Audio, APL, c intents, data persistence

By [Alexa](#)

Scheduling Skill

Build a skill to allow users to schedule appointments on your calendar service

Survey Skill

Build a stand-up or survey skill that uses a recorder to allow only authorized users

Intro to Alexa Conversations

This skill introduces you to Alexa Conversations by providing basic "Hello

Pizza Ordering Examp

An example food ordering skill wi Conversations demonstration use

5- In the new Interface, expand the “Interaction Model”, click on “JSON Editor”, “Drag and Drop a json file” and choose the file (skill_example.json attached).

1

2

3

If you want to develop the skill step by step, see Part2 (Alexa Skill Creation)

6- Save and build the model.

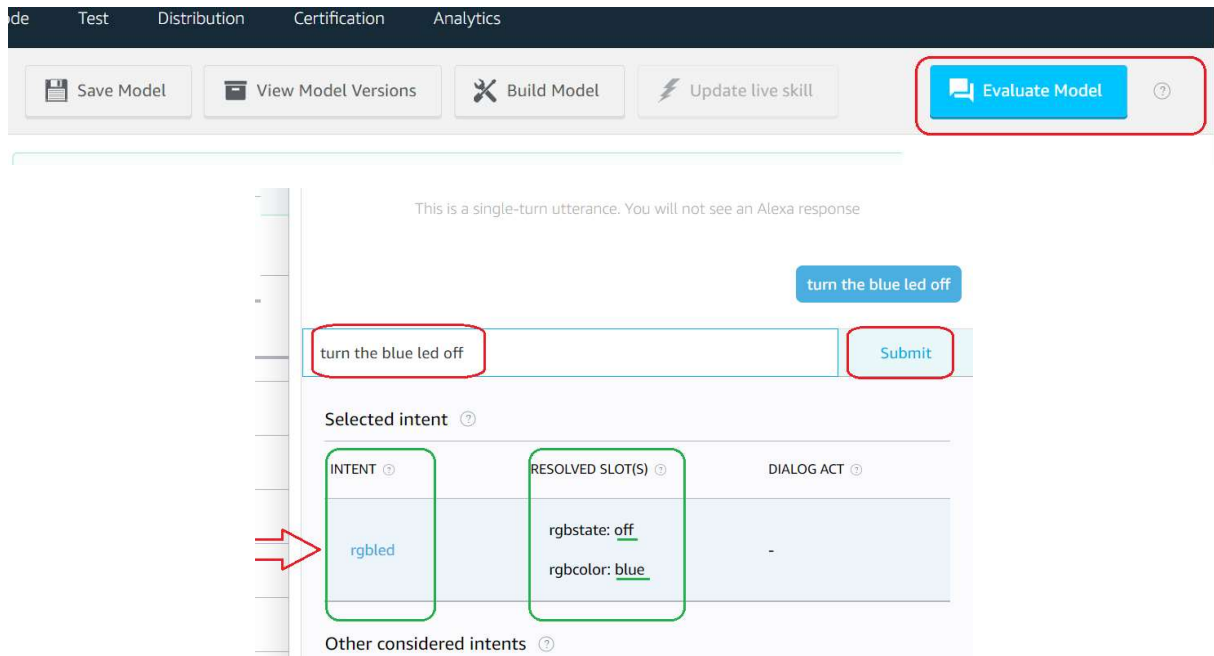
1

2

7- Now, it is possible to test if the built model is able to detect the slot values in one sentence (written then spoken).

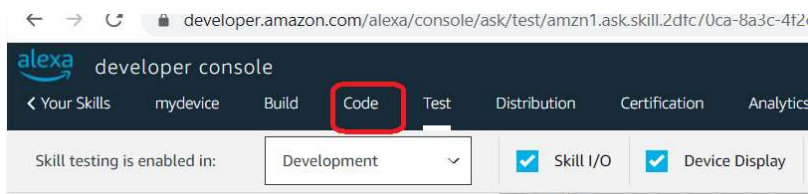
Click on Evaluate Model and write a sentence that contains 2 slot values (one for color and one for control), and submit the sentence.

A result will be displayed and shows if the intent and the slot values are resolved.

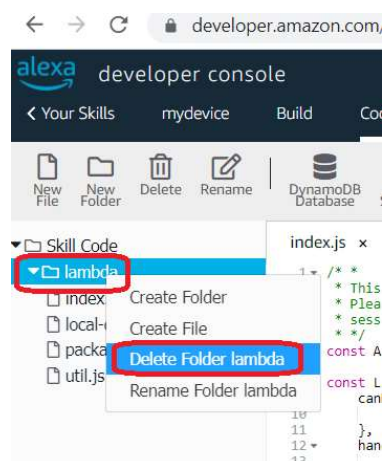


4-2 : Lambda Code

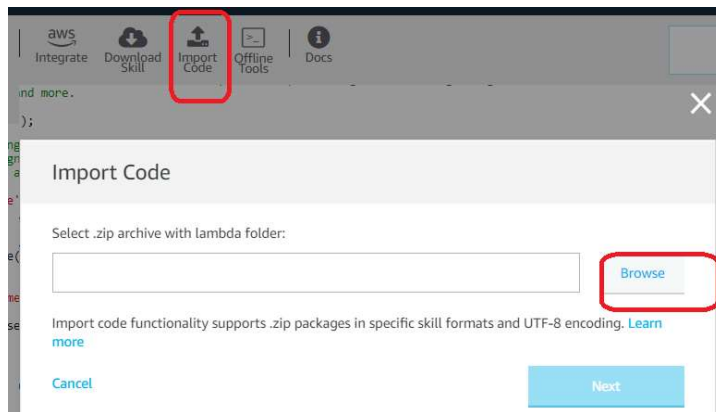
1- Switch to the code section



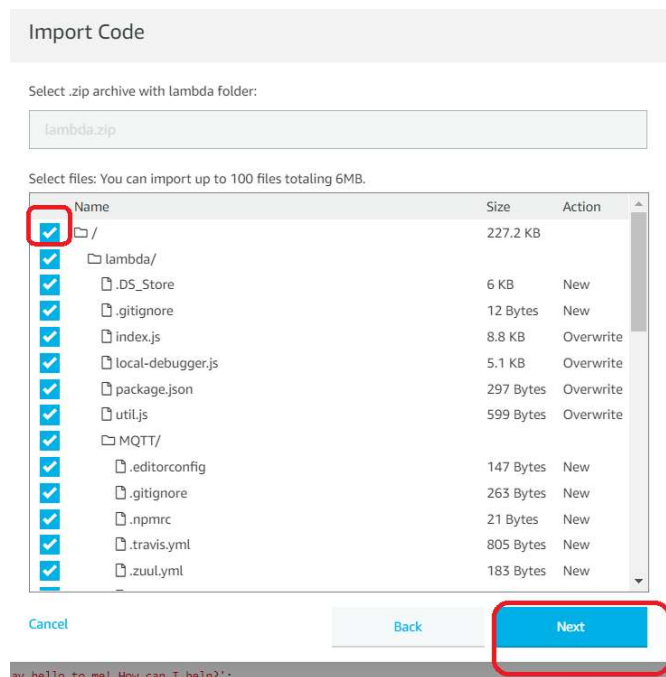
2- Delete the Lambda Folder



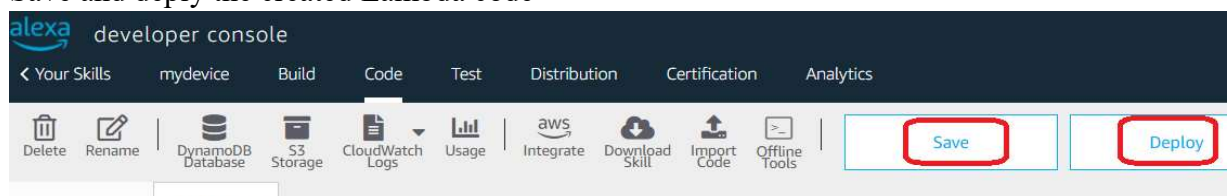
3- Click on Import Code, then browse and choose the attached lambda.zip file



4- Select all files and click next.



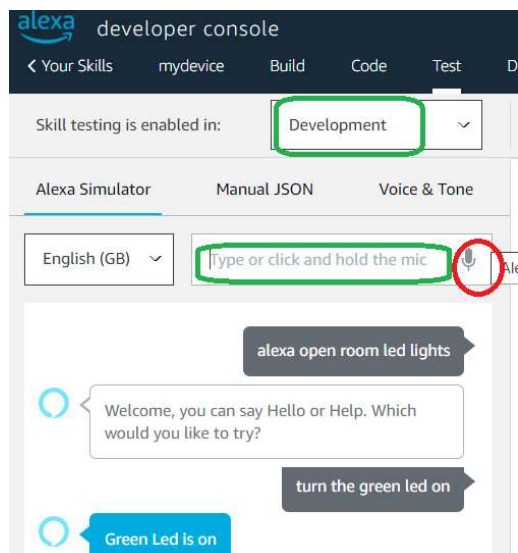
5- Save and deploy the created Lambda code



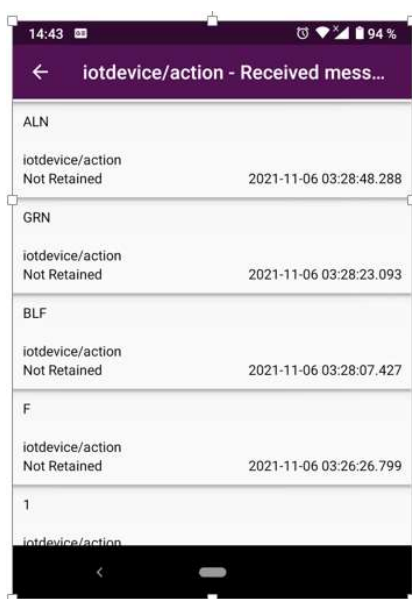
6- Test the Lambda Function:

Enable Development testing mode and

- type the sentence that triggers the skill (alexa, open room led lights). Or click on the Mic and say it.
- then a sentence to turn on or off one of the leds. Or click on the mic and say it.
- Alexa will send vocal responses.



- Alexa publishes also mqtt messages to the topic `iotdevice/action` (the mqtt account used in TP). Normally the messages will be displayed in the MQTT Client application when it subscribes to the topic.



The sent mqtt messages contain 2 letters for the color:

GR for Green
RD for Red
BL for Blue
AL for All

And one letter for the instruction

N for ON
F for OFF
T for Toggle

So

ALN means All Leds ON
GRN means Green Led ON
BLF means Blue Led OFF

Finally it is possible to test it with the ESP32 based Smart Device.