

Apartado 17: Programación de un skill de Alexa

Autor:
Miguel Ángel García Cumbreiras

Introducción

- Alexa es el asistente virtual controlado por voz creado por Amazon, y lanzado en noviembre de 2014.
- El funcionamiento de Alexa es muy parecido al de otros asistentes como Google Assistant, Siri y Cortana. Empiezas invocándolo diciendo su nombre, momento en el que el altavoz o dispositivo en el que esté integrado se pondrá a escuchar. Entonces debes decirle un comando con tu voz, y el asistente reconocerá lo que le preguntas y te dirá una respuesta.
- Las funciones de Alexa dependen de dos elementos claves. Por una parte están los comandos de voz que integra, y con los que puedes realizarle una gran variedad de peticiones. Y luego están las skills, que son complementos que le puedes instalar para añadirles aún más funcionalidades.

Introducción



Programación de skills

Alexa tiene un número de habilidades incorporadas, pero los desarrolladores pueden construir nuevas habilidades personalizadas, usando Alexa Skill Kit (ASK).

El ASK es una colección de APIs y herramientas que permite interactuar con Alexa con reconocimiento de voz, codificación de texto a voz y el procesamiento del lenguaje natural. ASK ayuda a los desarrolladores a desarrollar habilidades de forma rápida y sencilla.

La versión ASK SDK v2 es la última, y permite desarrollar Skills para Alexa en Español en los siguientes lenguajes de programación:

- Java
- Node.js
- Python

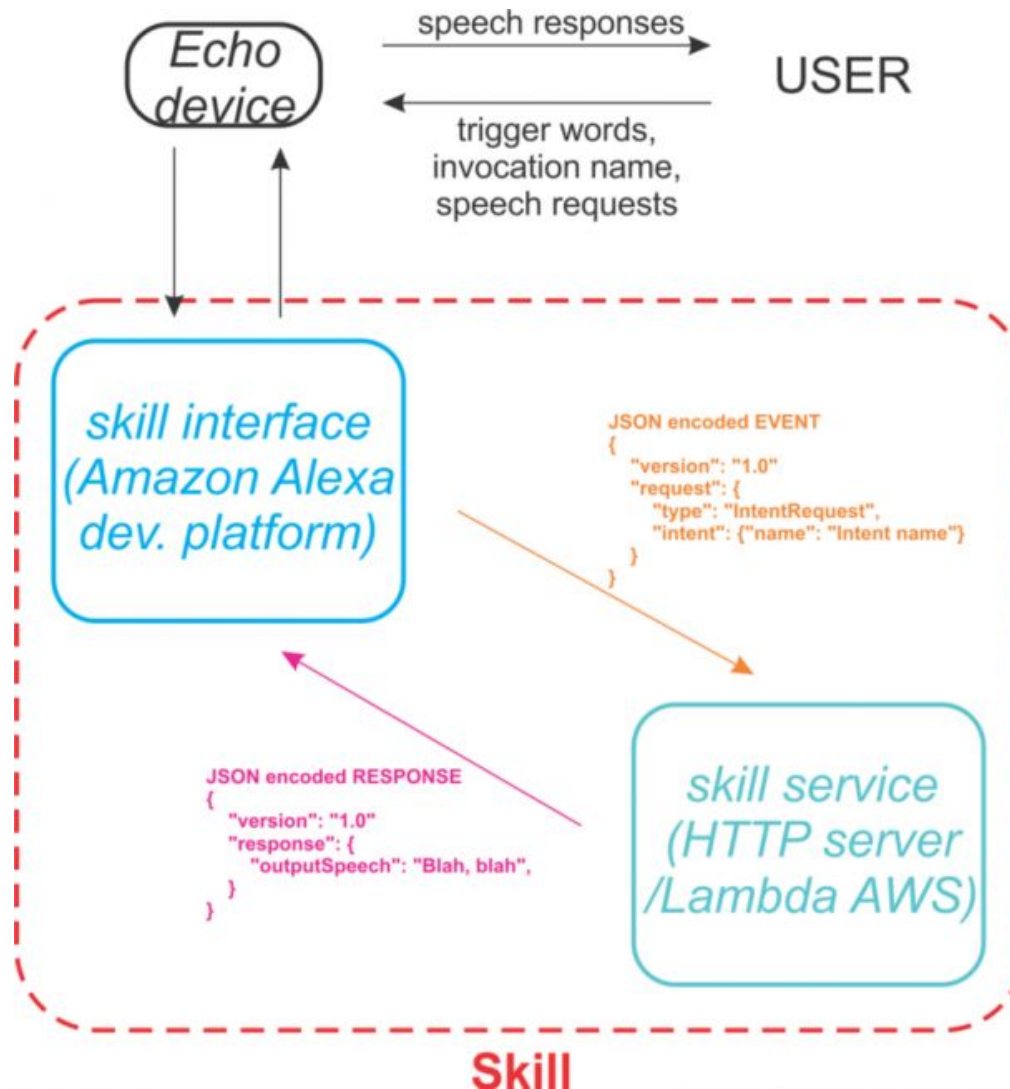
Cómo funciona la comunicación dentro del Custom Alexa Skill

El skill de Alexa consta de dos componentes principales: skill interface y el skill service.

Skill interface procesa las peticiones de voz del usuario y luego las mapea a intenciones dentro del modelo de interacción. Los intentos son acciones que cumplen con las peticiones habladas del usuario. Cada intención tiene al menos una expresión, una palabra, frase u oración predefinida que el usuario puede decir para invocar la intención. Si se detecta una intención específica, la interfaz de habilidades crea un evento codificado en json, que se transmite al servicio de habilidades.

Skill service determina qué acciones tomar en respuesta al evento codificado JSON recibido de la interfaz de habilidades. Al tomar una decisión, el servicio de habilidades devuelve una respuesta codificada JSON a la interfaz de habilidades para su posterior procesamiento. Después de procesar, la respuesta de voz se envía de vuelta al usuario a través del Echo machine.

Cómo funciona la comunicación dentro del Custom Alexa Skill



Creando un skill con Python (1)

Requisitos:

- Cuenta en la Consola de AWS, para poder utilizar Lambda.
- Cuenta de desarrollador de Alexa, para crear skills.
- Conocimientos básicos de línea de comandos.
- Conocimientos de Python

Dividimos una skill en:

1. Front-end
2. Back-end

Creando un skill con Python (2)

Front-end

- Objetivo de la skill
- Nombre
- Diálogos de interacción o script (happy path)
- Nombre de apertura (invocation name)
- Intenciones (intents)
- Enunciados (utterances)
- Variables (slots)

Back-end

- Función Lambda
- Otros servicios como: S3, IAM, etc.

Creando un skill con Python (3)

Front-end

1. Nombre y objetivo de la skill: pensemos...
2. Happy Path: tenemos que ver cómo sería la interacción entre Alexa y el usuario en el mejor de los casos (esto se conoce como happy path). Para esto existen unas cuantas herramientas, como Highland 2 (<https://quoteunquoteapps.com/highland-2/>).

Algunas recomendaciones:

- Mantener las interacciones breves.
- Redactar de la forma en la que hablamos y no de la manera en la que leemos o escribimos.
- Evitar frases repetitivas.
- Indicar claramente cuando se espera que el usuario provee información.
- No asumir que el usuario sabe que hacer o lo que va a pasar.
- Presentar las opciones de forma clara.
- En general, no proveer más de tres opciones a la vez.
- Solicitar información en trozos o fragmentos pequeños.

Creando un skill con Python (4)

Ahora ya podemos crear la skill:

- consola de desarrolladores de ASK (ASK Developer Console) para crear nuestra skill.
- Hacemos click en Create Skill.

```
Skill Name: xxxxxxx  
Default Language: Seleccionamos Spanish (ES)  
El tipo de skill que queremos es custom
```

Ahora decidimos cómo vamos a invocar nuestro skill. Vamos a utilizar el mismo nombre de la skill, que será el nombre de apertura, o invocation name.

Creando un skill con Python (5)

Intents

se pueden utilizar las intenciones predefinidas que Amazon ofrece, de hecho al momento de crear una skill hay al menos 4 intenciones por defecto. Esto depende de la skill que queremos crear, y podemos optar por una skill personalizada (custom skill).

Vamos a crear un custom intent : ListItemsIntent (click en Create custom intent).

Creando un skill con Python (6)

Utterances (enunciados)

Son variables que pueden tener cualquier tipo y valor. De hecho Amazon nos ofrece, ya en Español, 24 de ellos: números, países, colores, ciudades, días de la semana, entre otros. Por ejemplo:

```
las que tú me digas  
quiero buscar 4 cosas  
no lo sé  
decide tú  
cualquiera  
escoge tú  
da igual  
no importa  
5  
las que sean
```

Creando un skill con Python (7)

Variables (slots)

Son las frases que utiliza el usuario para interactuar con Alexa. Agregamos todas los enunciados que consideremos relevantes. Es importante agregar una buena variedad, incluso se recomienda que sean unas 30. Por ejemplo:

```
las que tú me digas  
quiero buscar {numObj} cosas  
no lo sé  
decide tú  
cualquiera
```

Creando un skill con Python (8)

Cuando escribimos nuestros enunciados con {numbObj} incluido como variable, automáticamente se crea una variable con ese nombre, ahora debemos definir el tipo.

Para ello desplegamos la lista y seleccionamos Amazon.NUMBER. Es uno de los slots predefinidos que permite a Alexa comprender cualquier número. Una vez seleccionado el tipo, observamos que inmediatamente se agrega un Slot Type(A). Ahora podemos guardar los cambios haciendo click en Save Model.

Hacemos click en Build Model. Este proceso lleva unos 20–30 segundos.

Creando un skill con Python (9)

Back-end

AWS Lambda es la apuesta de Amazon por la arquitectura serverless. Se trata de una arquitectura de microservicios que destaca por “no tener servidor”.

AWS Lambda soporta los lenguajes de programación Java, NodeJS y Python. Todos los códigos ejecutados en Lambda Amazon Web Services se les conoce como función Lambda y están listos para cuando se requiera. En términos sencillos son como fórmulas de Excel que están predeterminadas.

Creando un skill con Python (10)

Back-end

1. Entramos en Amazon AWS a través de la consola.
2. En la sección de AWS Services escribimos lambda, y pulsaremos en crear función. Posibilidades:
 - a. Crear una función desde cero (**Author from scratch**).
 - b. Una función basada en una plantilla (Blueprints).
 - c. Una función que ha sido publicada por AWS o sus socios, por ejemplo seleccionando esta opción encontramos las funciones de los ejemplos del repositorio de GitHub de Alexa.

Creando un skill con Python (11)

Back-end

3. Le ponemos un nombre a la función. Seleccionamos Python 3.6 como Runtime. Para el rol, tenemos dos opciones; creamos uno o utilizamos uno existente, en nuestro caso utilizamos uno existente, por lo que seleccionamos la opción Choose an existing role. En el menú desplegable seleccionamos `lambda_basic_execution` como rol de la sección Existing Role. Click en Create function.

Creando un skill con Python (12)

Conectamos Front-end y Back-end

Para conectarlo necesitamos el ARN (Amazon Resource Names) de la función Lambda y la Skill ID.

En la ventana “Designer” indicaremos qué es lo que dispara esta función, en el caso nuestro es una Skill de Alexa. Una vez copiado el ARN, volvemos a la consola de desarrolladores de Alexa (ASK Developer Console) y hacemos click en Endpoint, de esta forma cumpliremos con el último requisito para completar la skill. De esta forma le estamos diciendo a la skill cuál será la función Lambda que utilizará.

Creando un skill con Python (13)

Conectamos Front-end y Back-end

Seleccionamos la opción AWS Lambda ARN. Esto habilita varios cuadros de texto: uno de ellos con el título Default Region, donde tenemos que pegar el ARN de nuestra función Lambda. Copiamos el Skill ID puesto que lo necesitaremos para conectar el Front-End con el Back-End.

Creando un skill con Python (14)

Conectamos Front-end y Back-end

Volvemos a la Consola de AWS y a nuestra función Lambda. Hacemos click en Alexa Skills Kit para insertar el Skill ID. También podemos observar que automáticamente se crea un trigger.

En el panel inferior se activa una caja de texto en la que insertamos la Skill ID permitiéndonos conectar el Front-End con el Back-End y luego hacemos click en Add. Entonces se activa el botón superior Save y lo pulsaremos.

Creando un skill con Python (15)

Conectamos Front-end y Back-end

Volvemos a la consola de desarrolladores de ASK y hacemos click en Save Endpoint. Ya tenemos todos los componentes de nuestra skill completos.

Un descanso...

