Building Alexa Skills

@natevw

Full Stack TC, 2018 Jan. 9

Amazon Alexa

“Voice UI”
compare Apple Siri, Google Assistant, MS Cortana(, Samsung Bixby…)

You might think of one of these (Echo Dot), but it’s really more abstract.

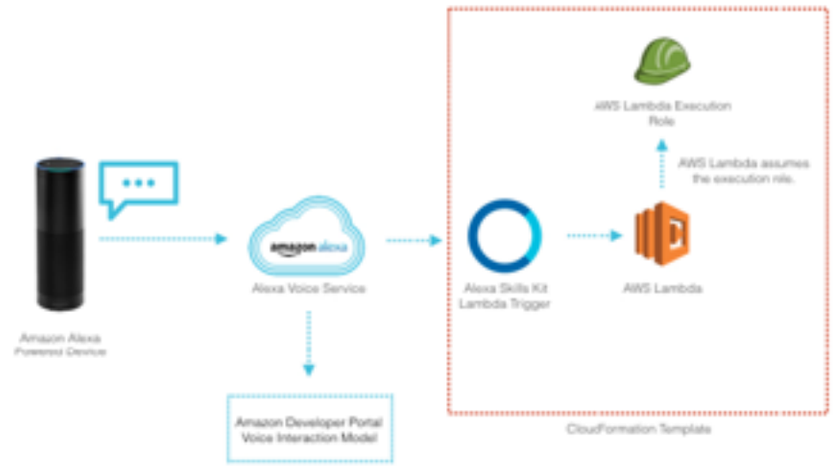
Amazon Alexa

Really a service that could run on any device, even 3rd party.

“Skills”

We’re going to learn about building skills for Alexa, “custom skills” — apps for Voice UI, conversational design. Where do these fit into the service as a whole?

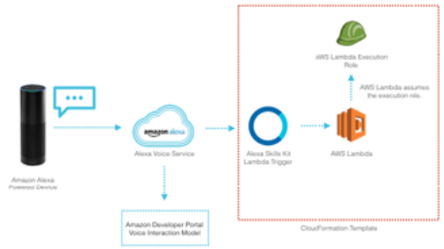
Technical overview



Source: https://developer.amazon.com/blogs/alexa/post/Tx27NAUCY0KQ34D/rapidly-create-your-alexa-skill-backend-with-aws-cloudformation

Alexa as a system — work happens all over! Watch word on device, audio (?) sent to Alexa's cloud servers, which then call out to our servers to handle.

Technical overview



- Automatic Speech Recognition
- Natural Language Understanding “four miles” vs. “for Miles”
- …leading to some Desired Outcome

https://youtu.be/umJsITGzXd0?t=34

Step back: what it takes to get this to work
Advanced technology?? [in some ways]
“understanding” — (Knowledge Navigator: https://youtu.be/umJsITGzXd0?t=34)

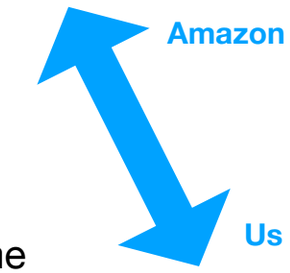
Technical overview

- Automatic Speech Recognition
- Natural Language Understanding
- …leading to some Desired Outcome

We don’t have to worry about all this. Amazon takes complete responsibility for the first, we really only have influence over the second [understatement?].

Technical overview

- Automatic Speech Recognition
- Natural Language Understanding
- …leading to some Desired Outcome



We don’t have to worry about all this. Amazon takes complete responsibility for the first, we really only have influence over the second [understatement?].

Technical overview

Returning to diagram, notice the dotted line boxes: our responsibility.
“Voice interaction model” → Natural Language Understanding
“Lambda Trigger” (in this example) → the outcome

“Understanding” ends up being a relatively simple model! Let’s start thinking about how these connect, via a bit more terminology.

Technical overview

oh, and to be clear as far as the big system goes our servers (that drive the outcome) don’t have to be Amazon-hosted or limited to just Alexa processing.

more later, but let’s talk Voice Interaction terminology when building skills

“Utterances” and “Intents”

(Understood language and outcomes)

These are Amazon’s terms, key to configuring a custom skill and thinking about implementation. Let’s start with intents

Intents

= What can the user do?

(broadly speaking) discrete actions, some analogy with GUI buttons or command line tools

Example: Egg Tracker

bring this home a bit

Example: Egg Tracker

Users can…

bring this home a bit

Example: Egg Tracker

Users can…

1. Store a record when hens lay eggs

bring this home a bit

Example: Egg Tracker

Users can…

1. Store a record when hens lay eggs
2. Retrieve records in summary form

bring this home a bit

Intents

Two things the user can do, two CUSTOM intents. These are basically string constants (enumerations) that identify actions when Amazon's computers are talking to our computers. Each one can get some specific configuration when we set up our skill, especially…

Intents

1. Store a record when hens lay eggs: **`LogEntry`**

Two things the user can do, two CUSTOM intents. These are basically string constants (enumerations) that identify actions when Amazon's computers are talking to our computers. Each one can get some specific configuration when we set up our skill, especially…

Intents

1. Store a record when hens lay eggs: **`LogEntry`**
2. Retrieve records in summary form: **`ShowStats`**

Two things the user can do, two CUSTOM intents. These are basically string constants (enumerations) that identify actions when Amazon's computers are talking to our computers. Each one can get some specific configuration when we set up our skill, especially…

(show Egg Tracker configuration)

Utterances

= What will the user say?

“utterances” must be configured for each intent.

Utterances

how many eggs
tell how many eggs
what is the egg count
how many eggs has the chicken laid
tell how many eggs the chickens have laid
tell me how many eggs have we gotten
how many eggs have we collected
how many eggs have we gathered
how many eggs have the hens laid
how many eggs have the chickens laid
tell me how many eggs the hens have laid
tell me how many eggs the chickens have laid
tell how many eggs the hens have laid
tell me how many eggs has the hen laid
tell me how many eggs has the chicken laid
tell how many eggs has the hen laid
tell how many eggs has the chicken laid
how many eggs has the hen laid
tell me how many eggs the hen has laid

Here are some [English] utterances for the ShowStats intent. All sorts of ways to ask the same question.

Utterance slots

{count} eggs
record {count} egg
our chicken laid {count} eggs
I gathered {count} more eggs
we gathered {count} egg
we gathered {count} eggs
we gathered {count} more egg
we gathered {count} more eggs
{count} egg laid
{count} egg found
{count} egg gathered
{count} egg collected
{count} eggs laid
{count} eggs found
{count} eggs gathered
{count} eggs collected
record {count} egg laid
record {count} egg found
record {count} egg gathered
record {count} egg collected

The LogEntry intent needs “a variable”. Amazon calls these “slots”. Can have multiple. (Also: utterances can be incomplete…)

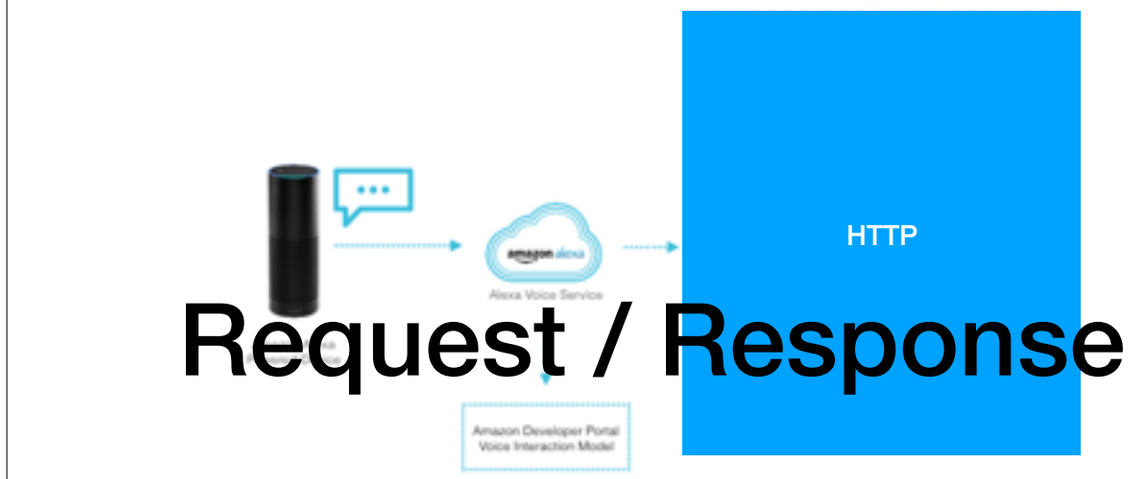
(show Egg Tracker configuration details)

(Other) Intents

e.g. AMAZON.HelpIntent

https://developer.amazon.com/docs/custom-skills/implement-the-built-in-intents.html

standard builtin intents, Amazon controls the utterances: HelpIntent, FallbackIntent, YesIntent/NoIntent. Also session (non-)intent requests.



Request / Response

(Skills!)

That’s more or less all there is to Alexa’s “understanding” as it pertains to making skills. The user says something, Amazon picks an intent and fills in any slots. (= Request) Your skill gets this simple little package, makes any outcome happen if necessary, and then needs to say something back. (= Response)
There’s a really common protocol that’s built around Request/Response: HTTP. And that’s how Amazon talks with our server. (Also like in HTTP: concept of sessions.)

(show Egg Tracker test console example)

Creating a skill

Really just three things we'll focus on tonight…

Creating a skill

1. “Invocation” (= name)

Really just three things we’ll focus on tonight…

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent

Really just three things we’ll focus on tonight…

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent
3. API endpoint (HTTP URL)

Really just three things we’ll focus on tonight…

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent
3. API endpoint (HTTP URL)
4. [Catalog/marketing stuff]

Really just three things we’ll focus on tonight…

(create a new skill via Alexa console)

Creating a skill

configured but haven’t really done step 3 yet, let’s first jump ahead and just see how that looks with Egg Tracker. if we have time we can play with this new skill we just added

Creating a skill

1. “Invocation” (= name)

configured but haven’t really done step 3 yet, let’s first jump ahead and just see how that looks with Egg Tracker. if we have time we can play with this new skill we just added

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent

configured but haven’t really done step 3 yet, let’s first jump ahead and just see how that looks with Egg Tracker. if we have time we can play with this new skill we just added

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent
3. API endpoint (HTTP URL)

configured but haven’t really done step 3 yet, let’s first jump ahead and just see how that looks with Egg Tracker. if we have time we can play with this new skill we just added

Creating a skill

1. “Invocation” (= name)
2. Utterances for each intent
3. API endpoint (HTTP URL)
4. [Catalog/marketing stuff]

configured but haven’t really done step 3 yet, let’s first jump ahead and just see how that looks with Egg Tracker. if we have time we can play with this new skill we just added

(cookie-cutter)

Creating a skill

also/full disclosure: there are other types of skill integrations. [cookie-cutter not to denigrate…]

(cookie-cutter)

Creating a skill

- Flash briefing
- Video/music
- Smart home
- etc.

also/full disclosure: there are other types of skill integrations. [cookie-cutter not to denigrate…]

(cookie-cutter)

Creating a skill

Pre-built models with different APIs, different configuration

although some overlap these are configured and implemented a bit differently, e.g. may not require Intent handling.

Creating a custom skill:

node.js alexa-sdk

Egg Tracker built in node.js using a helper library provided by Amazon, which mostly hides the underlying request/response details.

(walk through Egg Tracker lambda implementation)

Creating a custom skill:

HTTP in Python

(play with local “raw” Python server)

Links

Alexa docs:
https://developer.amazon.com/alexa-skills-kit

Egg Tracker:
https://github.com/natevw/chickening-alexa

Hanford Friday:
https://github.com/natevw/hanfordfriday-alexa