

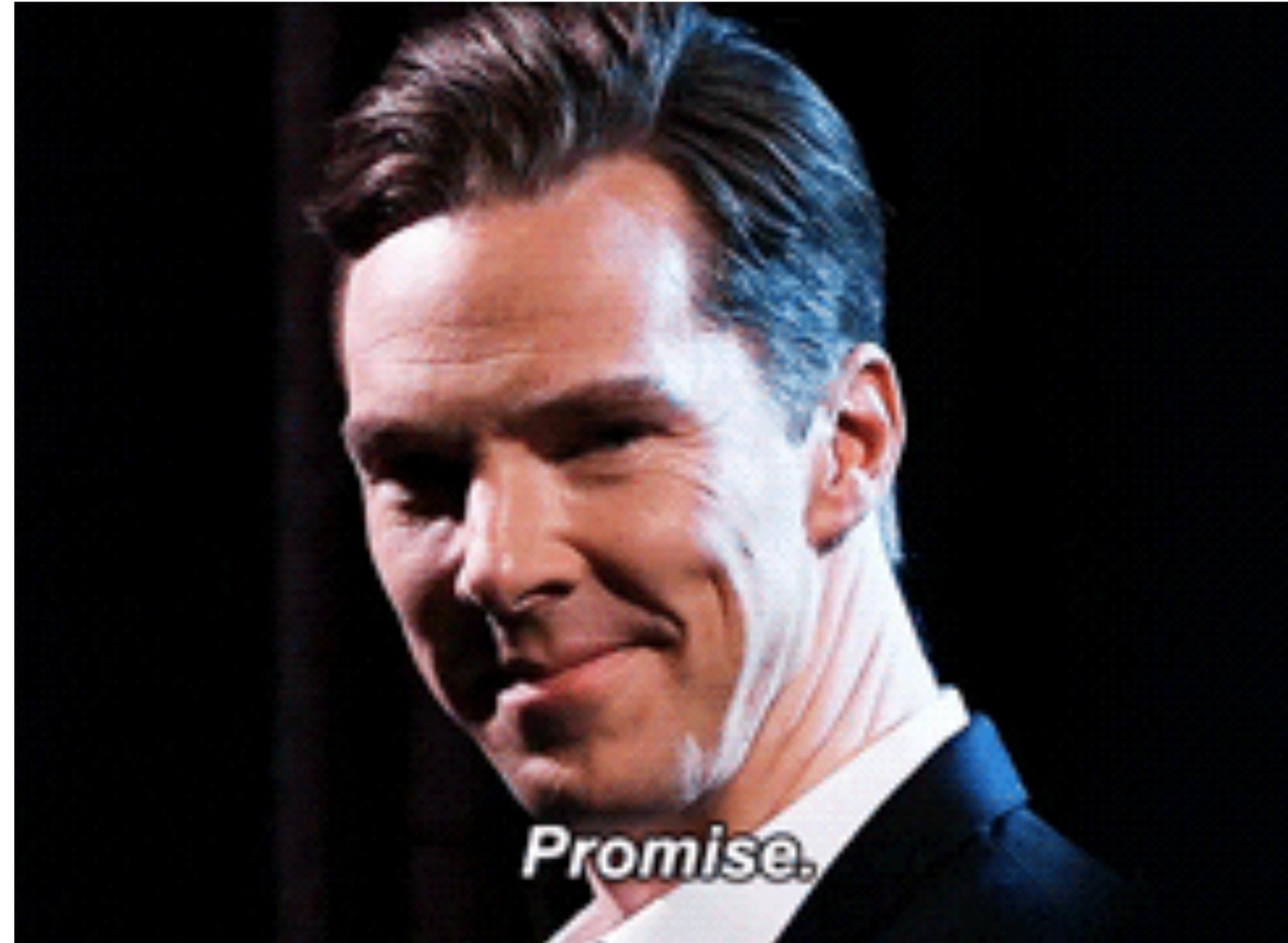
Alexa! Bake me a new Java repository

An introduction to voice bots productivity

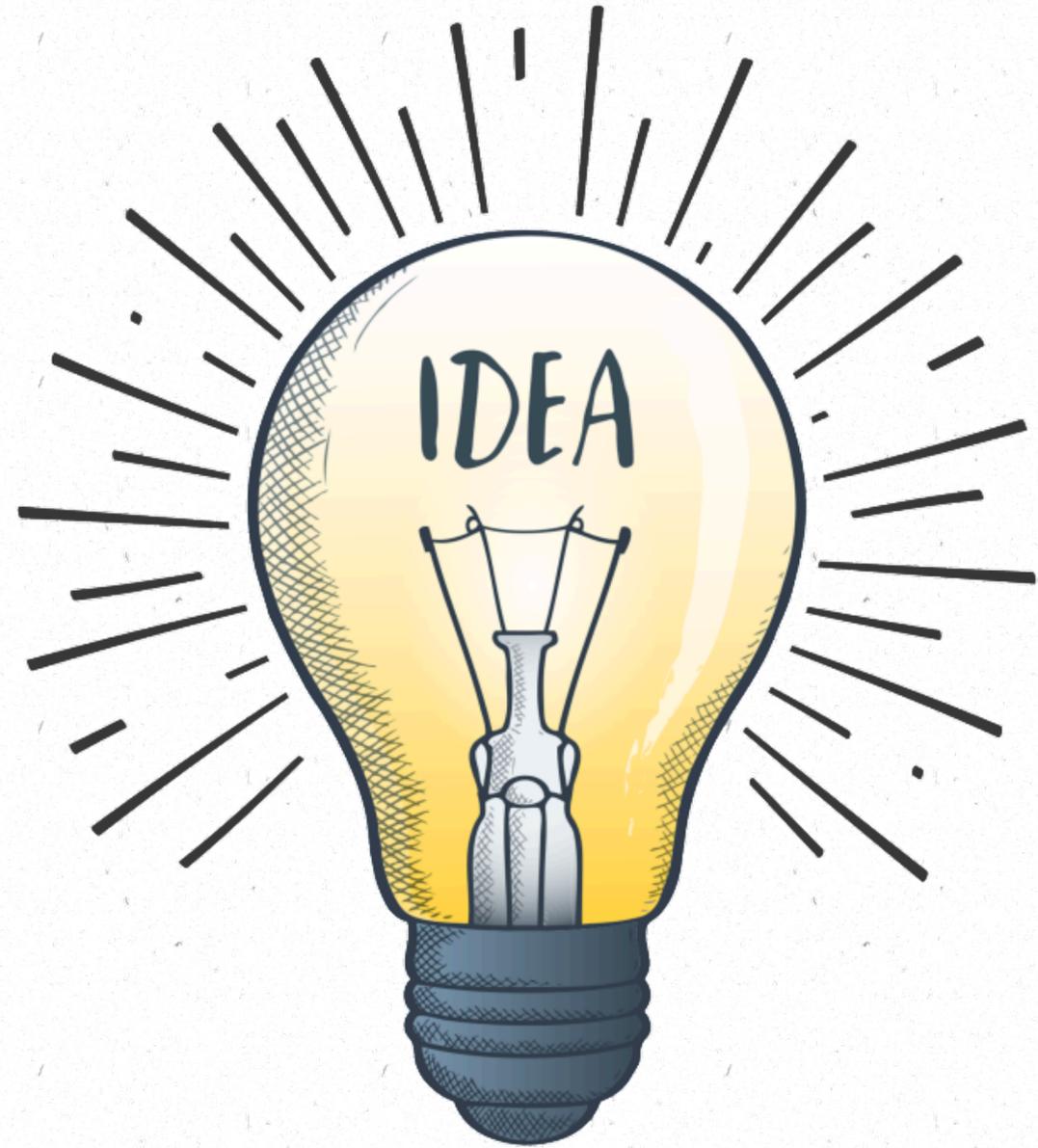
In 40 minutes

You will know

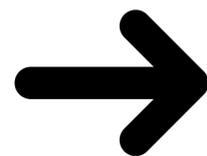
- What template repositories are
- The jargon used to build conversational apps
- How to create your own basic skills / apps using Alexa or DialogFlow



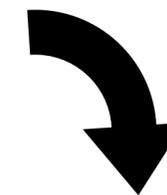
Idea sparks



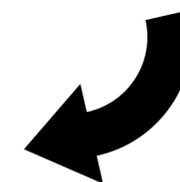
Email myself



**Create GitHub
project**



Scaffolding

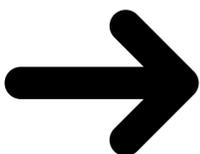


**Clone the
project**

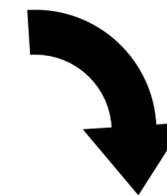


**Start
cranking**

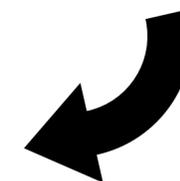
Email myself



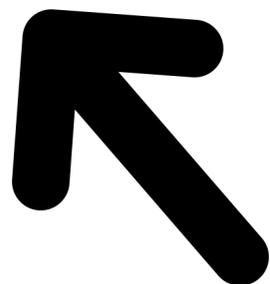
Create GitHub
project



Scaffolding



Clone the
project



Stop
working

Go to bed



Github template repos to the rescue!

June 6, 2019 — Features, Product

Generate new repositories with repository templates



Bryn Jackson

Sharing boilerplate code across codebases is a constant pattern in software development. Bootstrapping a new project with our favorite tools and directory structures helps programmers go from idea to “Hello world!” more efficiently and with less manual configuration.

Today, we’re excited to introduce repository templates to make boilerplate code management and distribution a first-class citizen on GitHub. To get started, all you need to do is mark a repository as a template, and you’ll immediately be able to use it to generate new repositories with all of the template repository’s files and folders.

Share

 Twitter

 Facebook

 LinkedIn

This is what a template looks like

Spring-Boot-Framework / **Spring-Boot-Application-Template** Template Watch 7 Star 40 Fork 142

[Code](#) [Issues 1](#) [Pull requests 1](#) [Discussions](#) [Actions](#) [Projects 1](#) [Wiki](#) [Security](#) [Insights](#)

master 3 branches 6 tags [Go to file](#) [Add file](#) [Code](#) [Use this template](#)

 **AnanthaRajuC** MODIFIED CHANGELOG.md ✖ a4980a3 4 days ago 🕒 363 commits

 .circleci	Add .circleci/config.yml	7 months ago
 .github/ISSUE_TEMPLATE	Update issue templates	6 months ago
 .mvn/wrapper	initial commit	3 years ago
 bin	actuator properties, css added	3 years ago

About

Spring Boot Web App, Flyway, MySQL, H2DB, Bootstrap, Thymeleaf, JWT, Swagger, API Rate Limiting, Docker, RBAC, i18n

[anantharajuc.github.io/spring_boot...](#)

[mysql](#) [i18n](#) [bootstrap](#) [fragments](#)
[docker](#) [spring-boot](#) [thymeleaf](#)

And this is how to create one

Settings

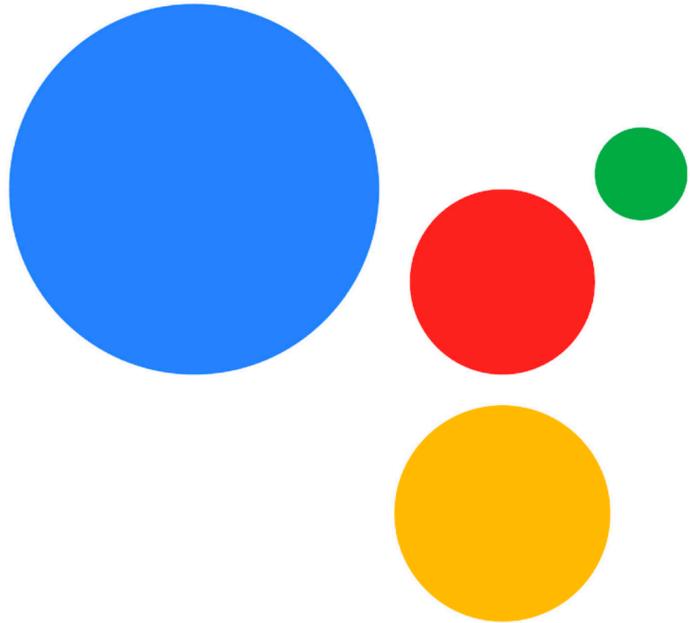
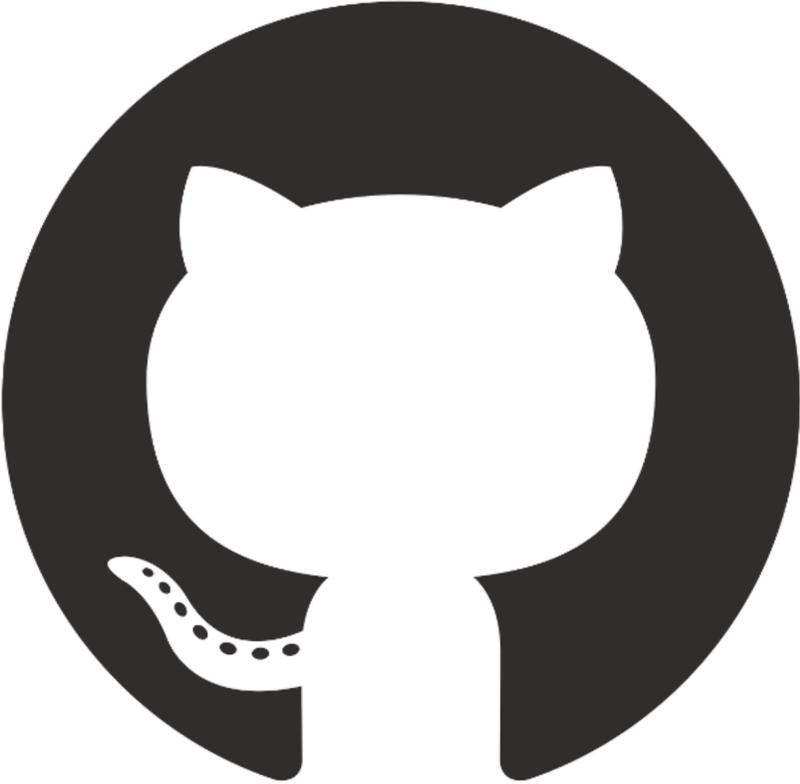
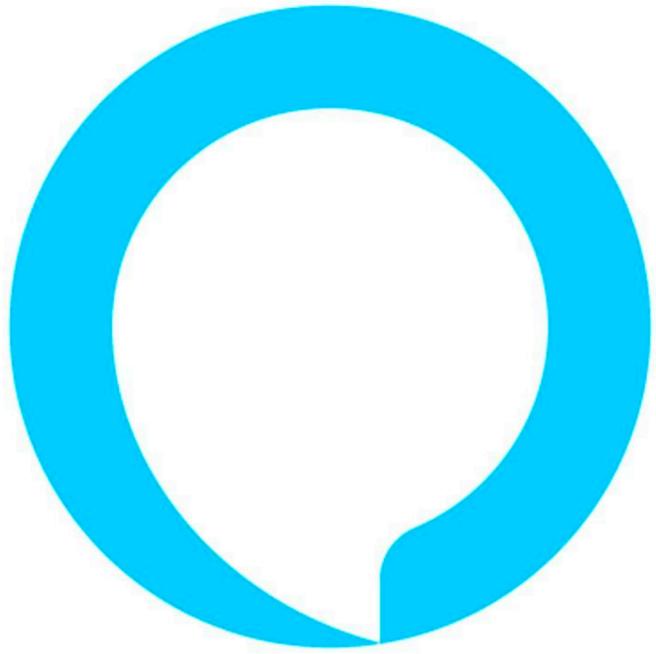
Repository name

epaper-dashboard

Rename

Template repository

Template repositories let users generate new repositories with the same directory structure and files. [Learn more.](#)





Well actually





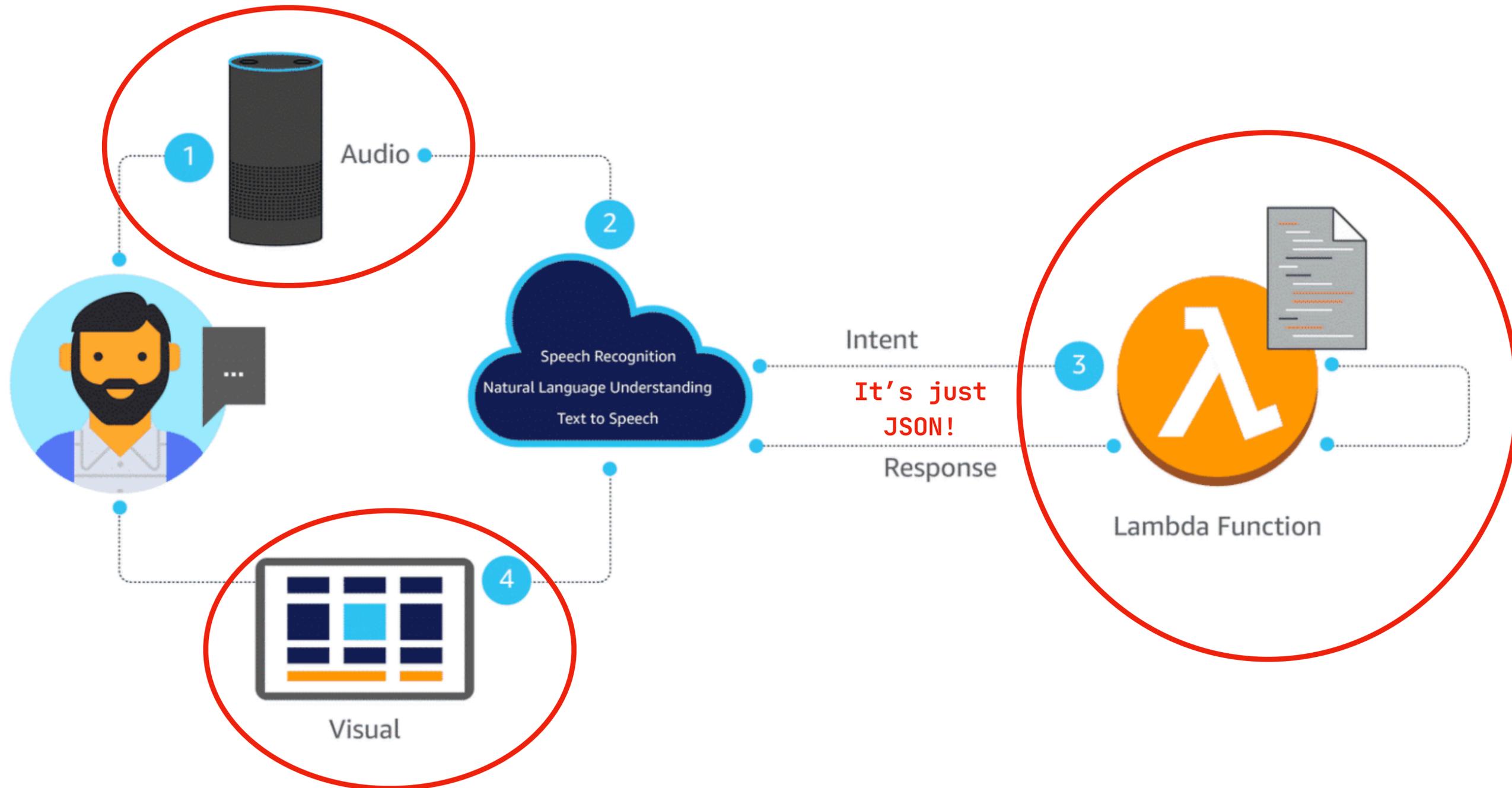
But don't worry!

- Jargon, interfaces are just about interchangeable!
- The main difference for me is that Alexa allows for 'private' skills.

Lots of magic are packed in Skills



Skills have a 'front' and a backend



Some mandatory jargon



The basics of conversational apps

- A ***skill*** : A complete app, essentially.
- ***Invocation name*** : How the user will activate your specific app.
- An ***invocation*** : When a user asks something to your skill
- An ***intent*** : The action that a user wants to perform.
- An ***utterance*** : In what way may the user express its intent.
- A ***slot*** : A bunch or arguments needed to perform the intent.
- A ***slot type*** : The type of data of the given slot

An illustrated example



And Amazon is usually quite smart

Utterance

Maps to

"I am going on a trip Friday."

TRAVEL_DATE

"I want to visit Portland."

TO_CITY

"I want to travel from Seattle to Portland next Friday."

FROM_CITY, TO_CITY, and TRAVEL_DATE

"I'm driving to Portland to go hiking."

MODE_OF_TRAVEL, TO_CITY, and ACTIVITIES

ASK : Alexa Skills Kit



```
$ brew install ask-cli  
$ ask new test-project  
$ cd test-project  
$ ask deploy
```

Alright! Time to work!



Takeaways : Github Templates

- A very fine idea! Less boilerplate is more happiness
- But the execution is lacking so far?
 - Not searchable on the website
 - Not findable either via the API
 - Very little apparent usage



Takeaways : Conversational apps

- Abstracts AWS away!
- Very easy to get started and build your own fun things
- Make sure to have enough utterances
- Can be lacking with technical terms, or ACCENTS!
- Building models take a while. Test using JSON allows for faster feedback loop
- **Extremely frustrating when it doesn't work**

Takeaways : Alexa vs DialogFlow

- **Alexa rocks because**
 - You don't want to use JS.
 - Better integration with AWS, SDKs in multiple languages
 - Private skills are there for you
- **DialogFlow rocks because**
 - Multiple out of the box integrations (Slack, web, ...)

Questions ?

[Hit me up on Twitter @jlengrand](#)

Slides / Code here :)