bdemers
BrianDemers
developer.okta.com
OktaDev

Generate Code that Doesn’t Suck

Brian Demers
Developer Advocate @ Okta
bdemers@apache.org

Who is this guy?

Stormpath

{okta}

THE APACHE SOFTWARE FOUNDATION

codemonkey.fm

openstack

Java

CHEF

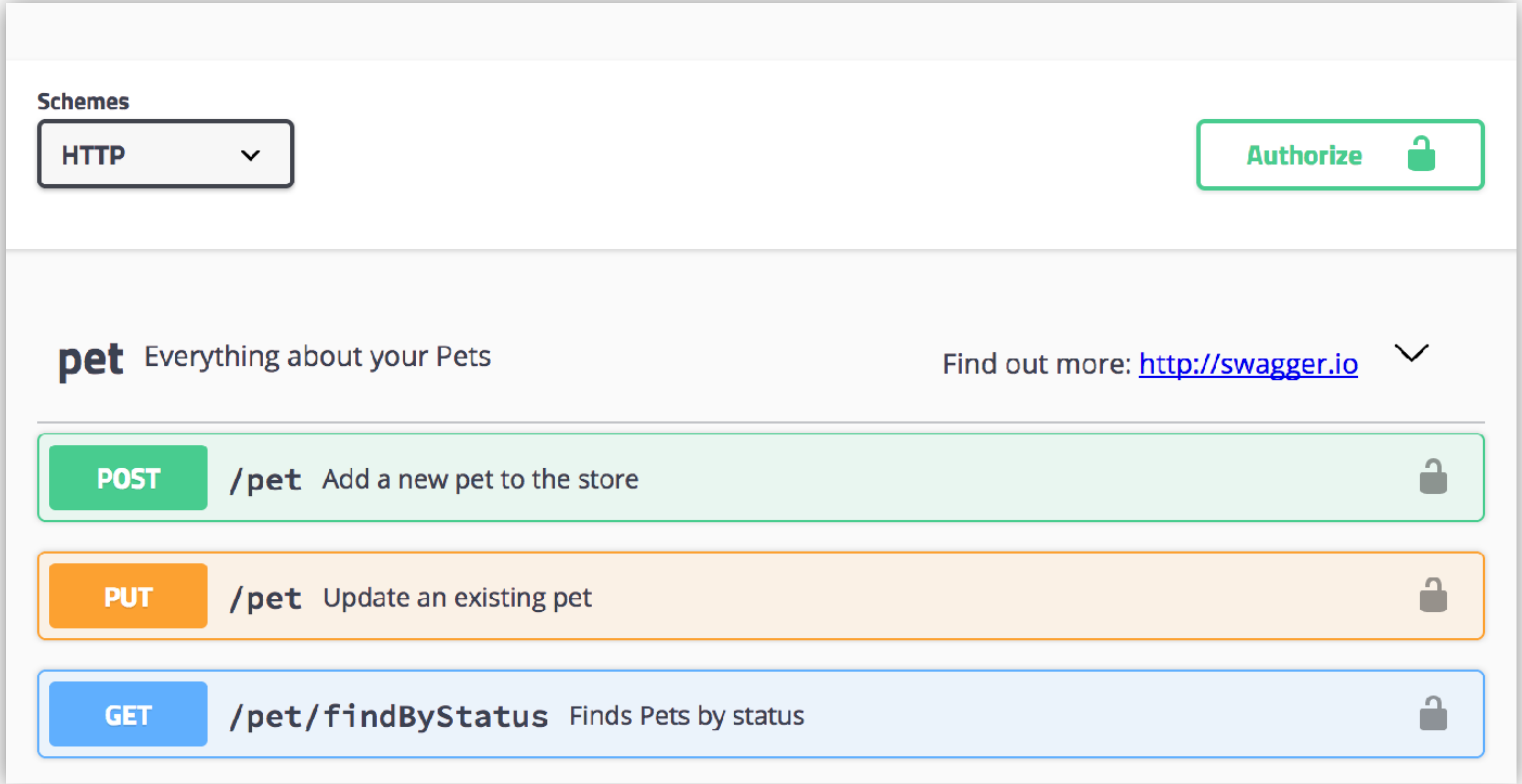
Maven

Sonatype Nexus

Overview

- What is Swagger / OpenApi
- What is generated code?
- Why generate code?
- Why is it perceived as bad?
  - Examples of code that sucks
- How do we make it better?
  - Examples of code that doesn’t suck

What is swagger?



Schemes

HTTP

Authorize

pet Everything about your Pets

Find out more: http://swagger.io

POST /pet Add a new pet to the store

PUT /pet Update an existing pet

GET /pet/findByStatus Finds Pets by status

Swagger

```yaml
basePath: "/api"
paths:
  "/pet":
    post:
      operationId: addPet
      consumes:
      - application/json
      produces:
      - application/json
      parameters:
      - in: body
        name: body
        required: true
        schema:
          "$ref": “#/definitions/Pet"
                                                …
```

```yaml
definitions:
  Pet:
    type: object
    properties:
      id:
        type: integer
      name:
        type: string
```

Parts of Swagger

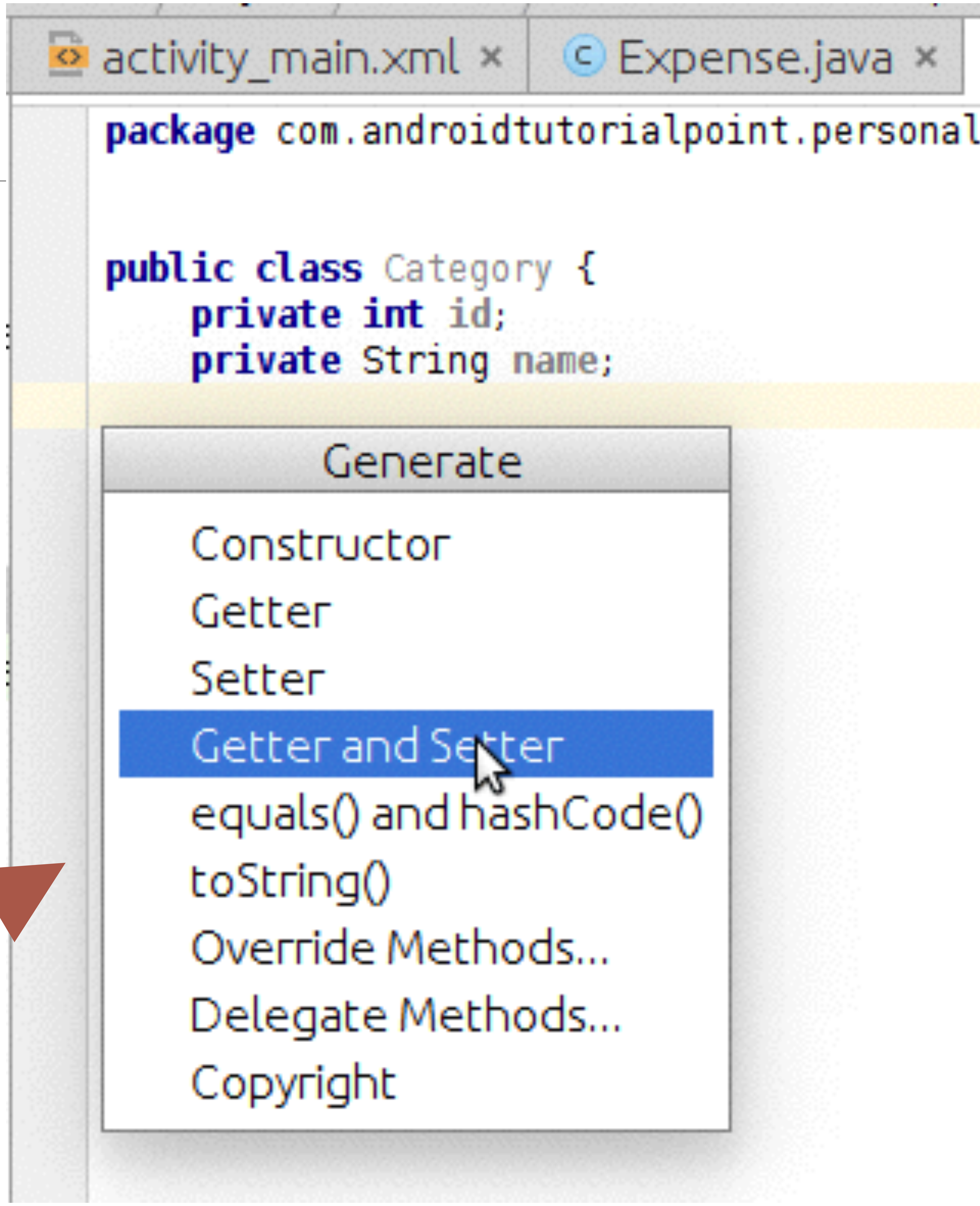
- Swagger Editor
- Swagger UI
- Swagger Codegen

“Sweet! It can generate code too!”

–Nobody, Ever

What is Generated Code

- **NOT** talking about IDE generated or boiler plate
- Code you do **not** change directly
- Code you do **not** check in (depending on the language)
- Has some sort of **manifest** (swagger.json, WSDL, RAML)



```
package com.androidtutorialpoint.personal


public class Category {
    private int id;
    private String name;
```

Not this

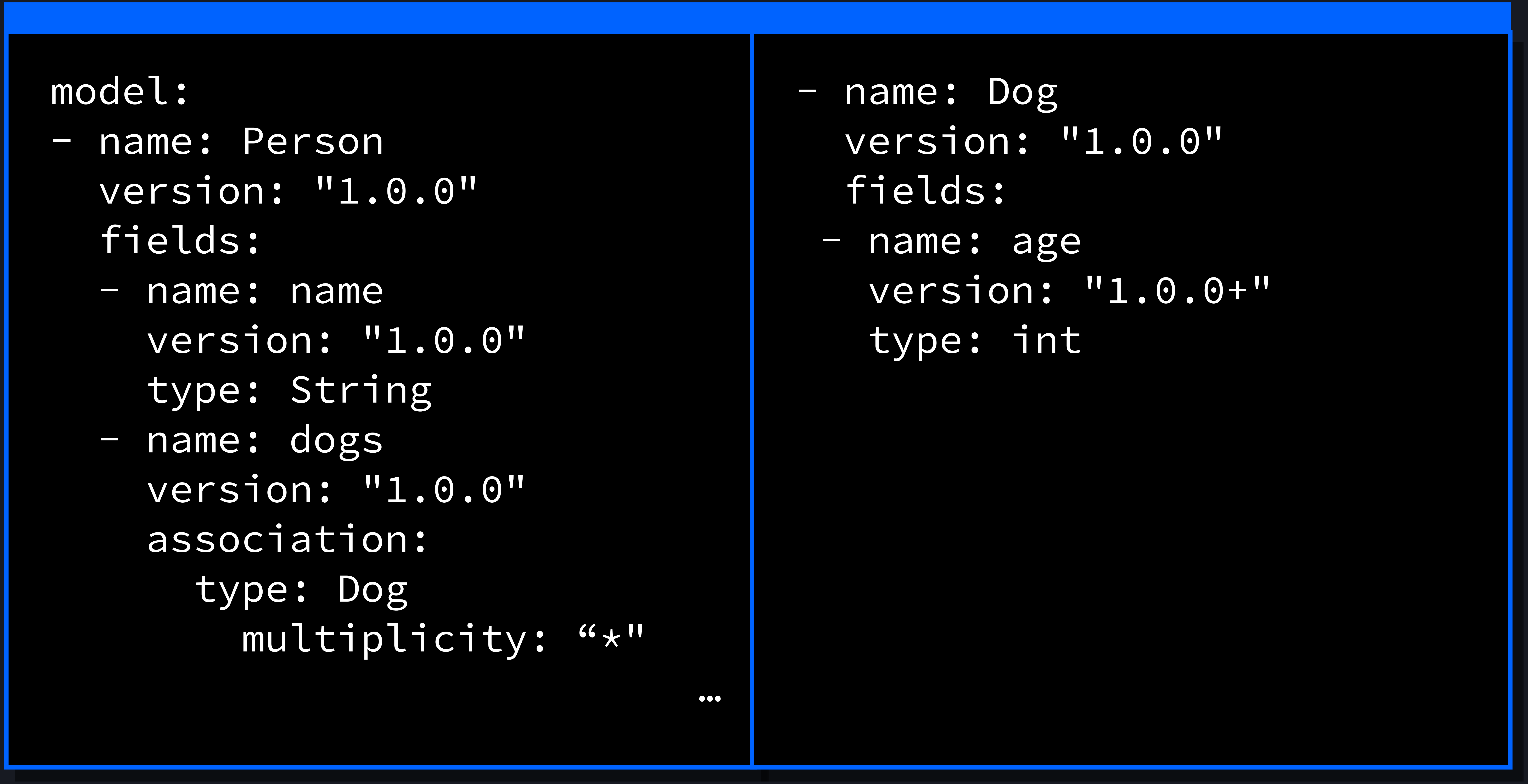
Lombok

projectlombok.org

```
@Data
public class Company {
    private final Person founder;
    private String name;
    private List<Person> employees;
}
```

Modello

codehaus-plexus.github.io/modello

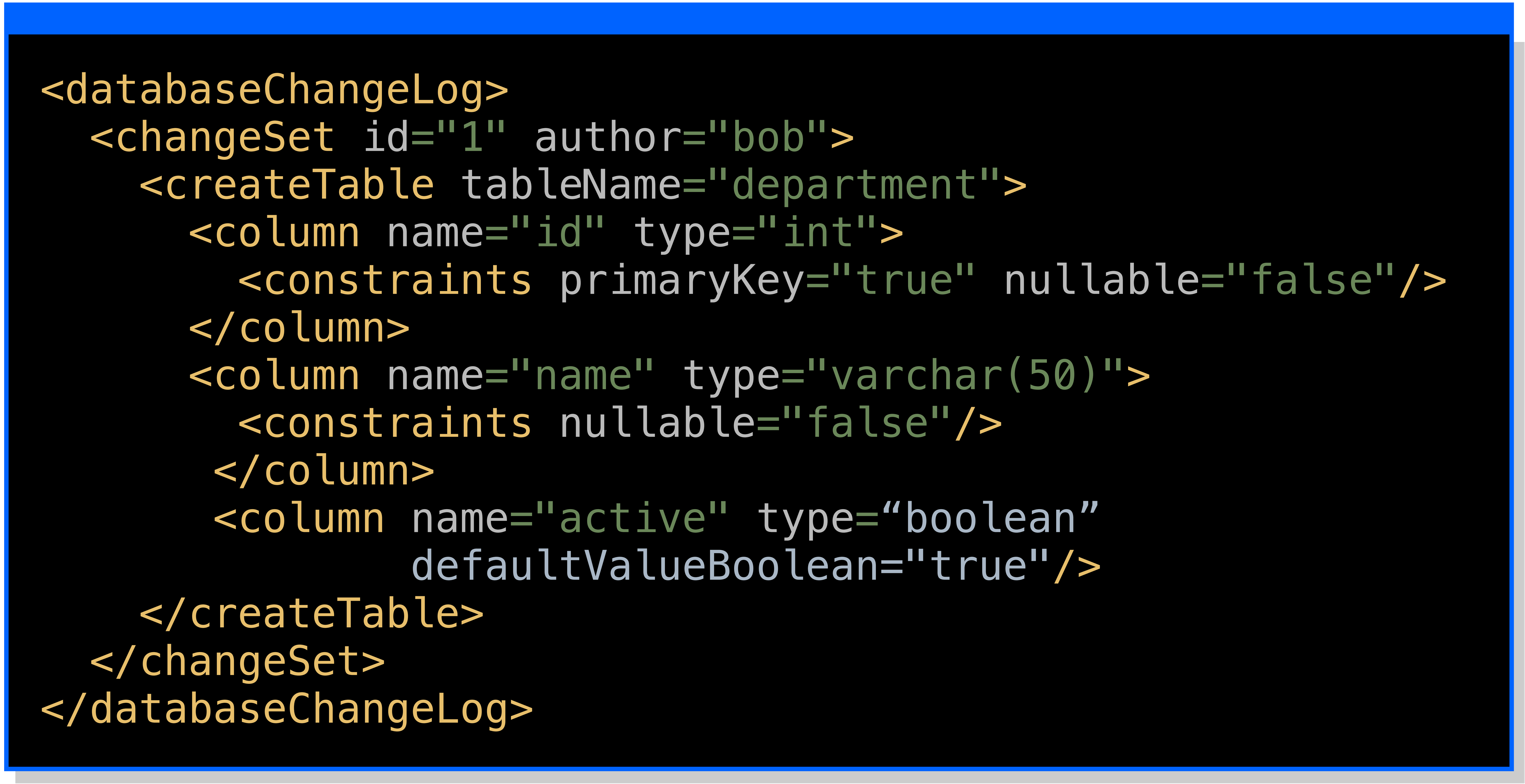


```
model:
- name: Person
  version: "1.0.0"
  fields:
  - name: name
    version: "1.0.0"
    type: String
  - name: dogs
    version: "1.0.0"
    association:
      type: Dog
        multiplicity: “*"
                          …
```

```
- name: Dog
  version: "1.0.0"
  fields:
 - name: age
   version: "1.0.0+"
   type: int
```

Liquibase

liquibase.org

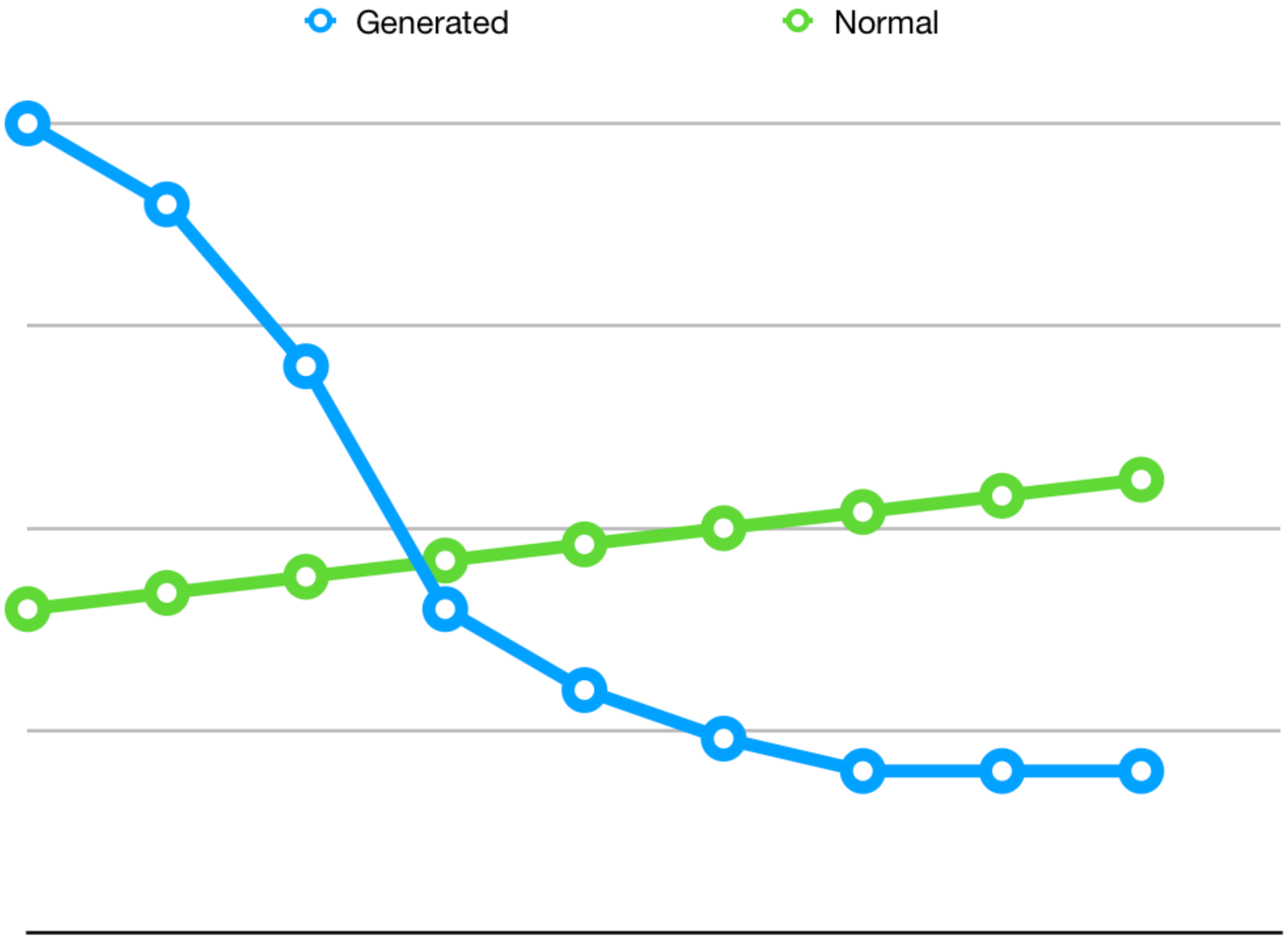


```
<databaseChangeLog>
  <changeSet id="1" author="bob">
    <createTable tableName="department">
      <column name="id" type="int">
        <constraints primaryKey="true" nullable="false"/>
      </column>
      <column name="name" type="varchar(50)">
        <constraints nullable="false"/>
       </column>
       <column name="active" type=“boolean”
               defaultValueBoolean="true"/>
    </createTable>
  </changeSet>
</databaseChangeLog>
```

Why?

FEDERAL RESERVE
AB 39048514
B2
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
Treasurer of the United States.
G1
100

Cost of Code Over Time (we hope)

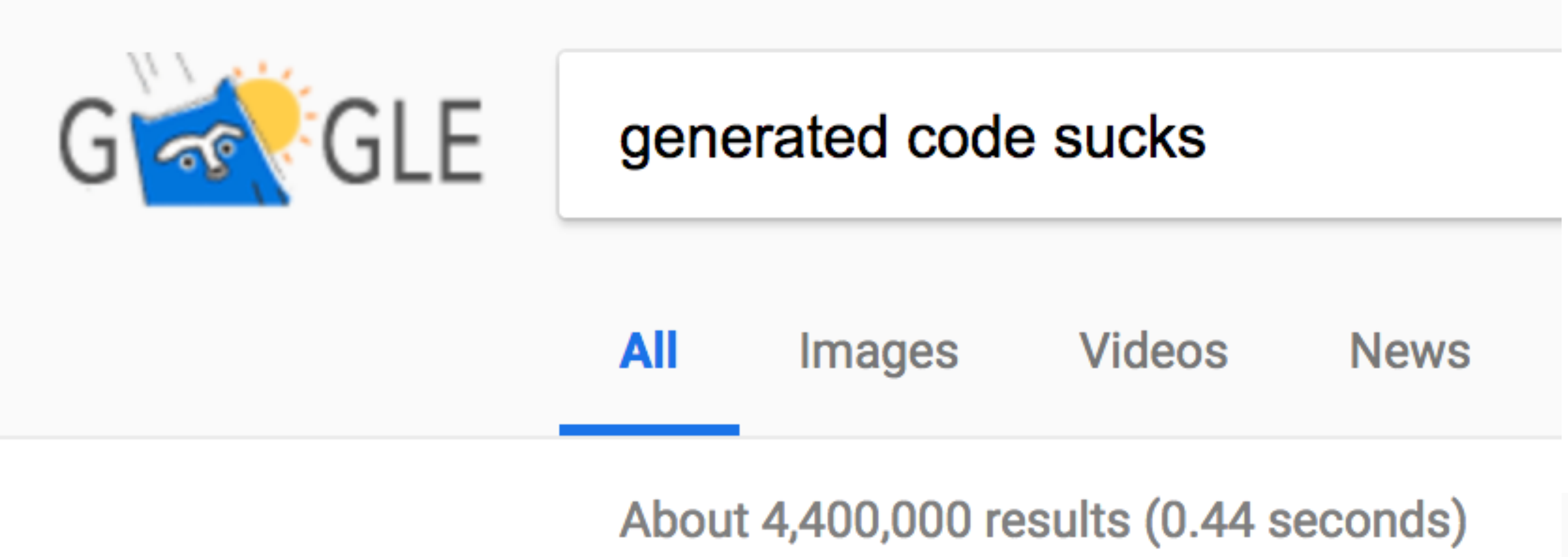


(fake data)

Why is it perceived as bad?

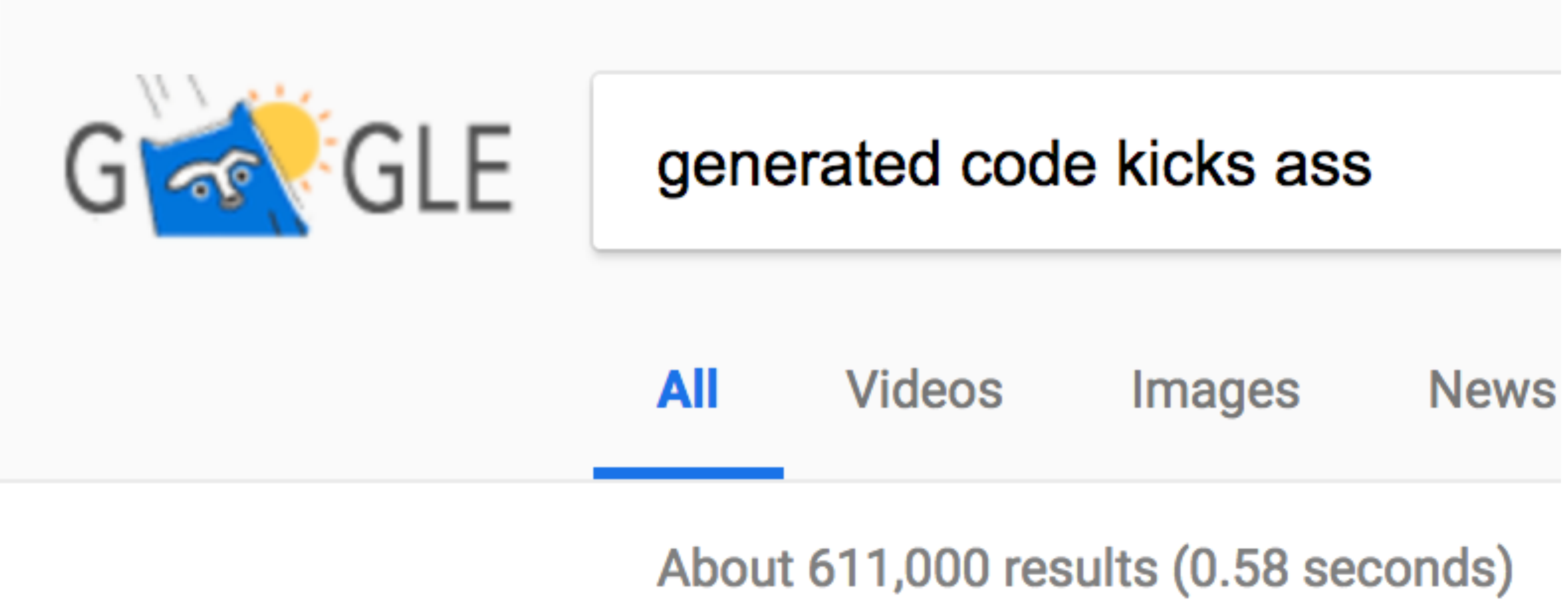
- Past stigma
- Complexity
- Ego
- Rigid output

Why do we think it sucks?



generated code sucks

About 4,400,000 results (0.44 seconds)

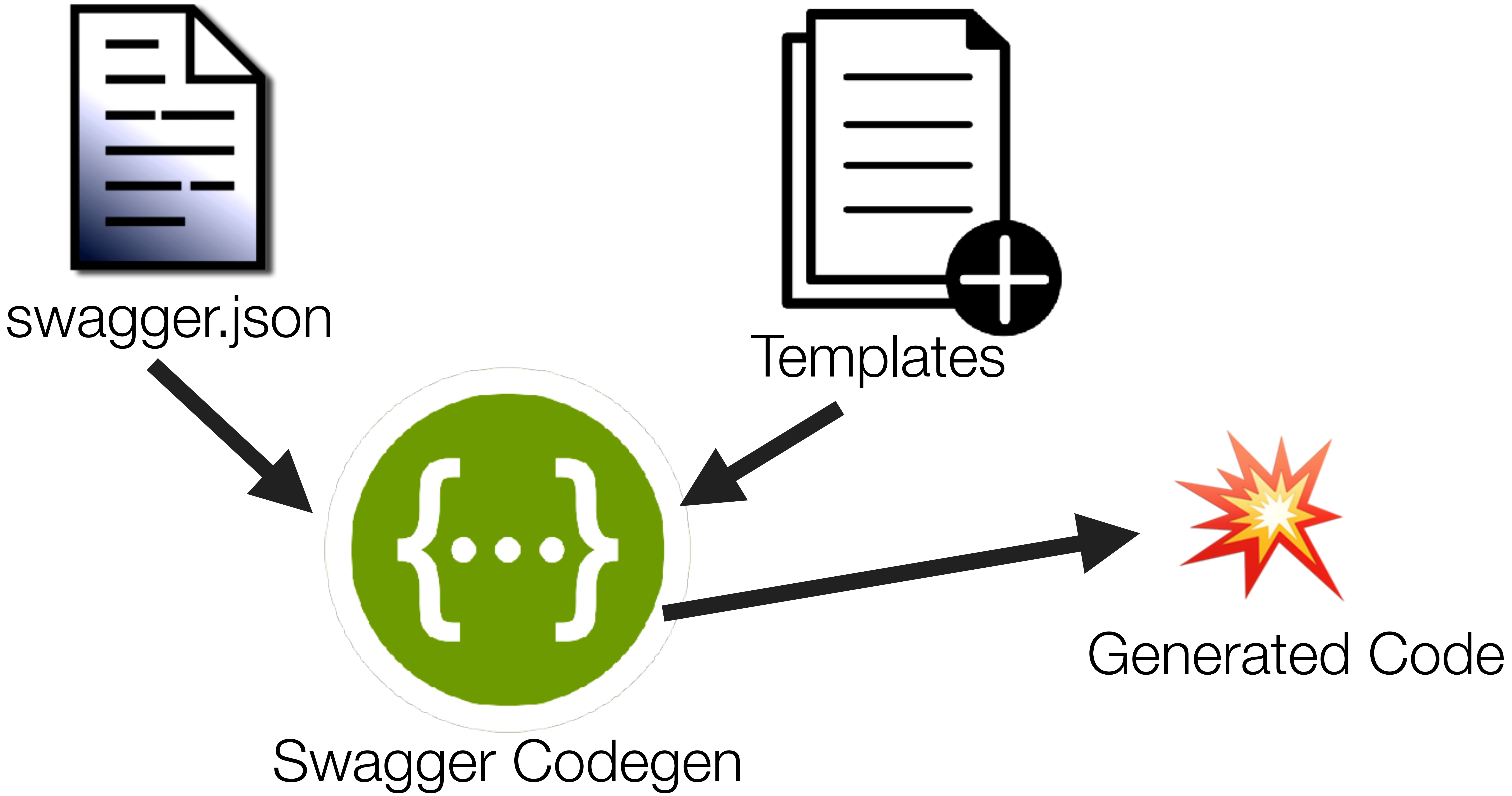


generated code kicks ass

About 611,000 results (0.58 seconds)

WORDPRESS.ORG
CODE IS POETRY
WORDPRESS.ORG
CODE IS PO

How to Generate Code with Swagger



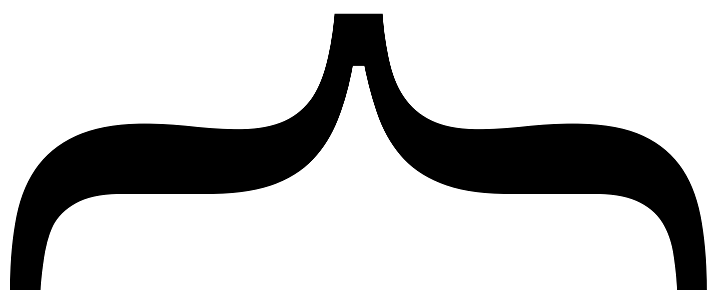
swagger.json

Templates

Swagger Codegen

Generated Code

Mustache

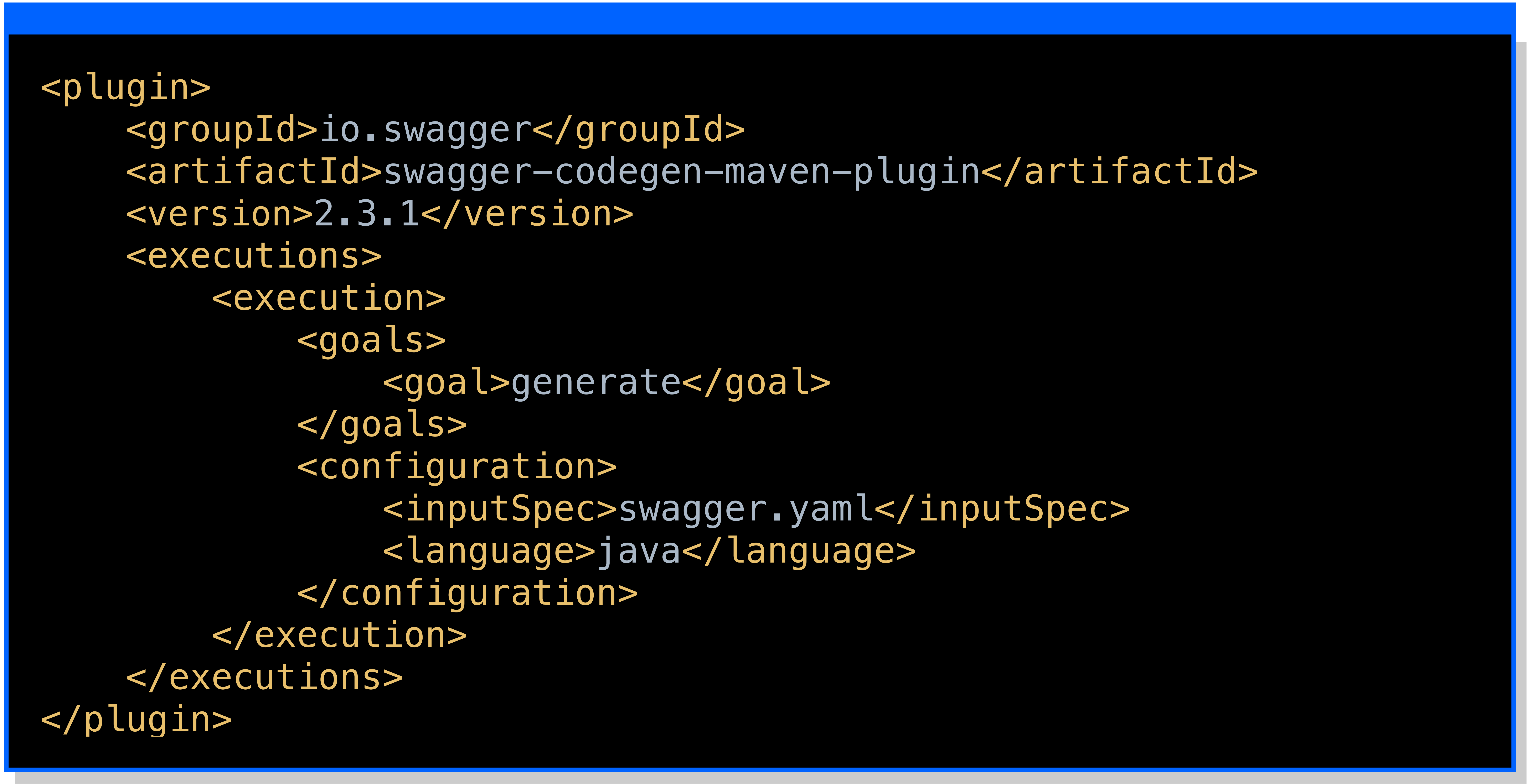


Logic-less templates.

Except:

```
{{#returnType}}{{{returnType}}} {{/returnType}}{{^returnType}}void {{/returnType}}
```

Swagger CodeGen Maven plugin



```
<plugin>
    <groupId>io.swagger</groupId>
    <artifactId>swagger-codegen-maven-plugin</artifactId>
    <version>2.3.1</version>
    <executions>
        <execution>
            <goals>
                <goal>generate</goal>
            </goals>
            <configuration>
                <inputSpec>swagger.yaml</inputSpec>
                <language>java</language>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Jump out to standard-swagger

Issues with this approach

- Generates lots of crap
- No interfaces
- Generates separate models and client
- Exposes underlying implementations
- Generates non changing boiler plate

How do we make it better?

- Understand the effort involved
- Don’t generate the world
- Stub out a non-generated example
- Generate only the code that repeats
- Use Interfaces!

Example Maven Dependencies

```xml
<dependency>
    <groupId>com.okta.example.sdk</groupId>
    <artifactId>okta-example-sdk-api</artifactId>
    <scope>compile</scope>
</dependency>
<dependency>
    <groupId>com.okta.example.sdk</groupId>
    <artifactId>okta-example-sdk-impl</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>com.okta.example.sdk</groupId>
    <artifactId>okta-example-sdk-httpclient</artifactId>
    <scope>runtime</scope>
</dependency>
```

A better Swagger plugin for Maven

```xml
<plugin>
    <groupId>org.zalando.maven.plugins</groupId>
    <artifactId>swagger-codegen-maven-plugin</artifactId>
    <executions>
        <execution>
            <id>swagger-codegen-api</id>
            <goals>
                <goal>codegen</goal>
            </goals>
            <configuration>
                <language>custom_java</language>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Jump out to custom-swagger

Don’t generate everything

- Hand write the complex bits
- Be careful of with secrets
- Models and methods can coexist
- Do **NOT** generate tests

Common Mistakes

- Generating clients on the fly
- Not testing
- Generating unfamiliar languages
- Set it and forget it

Ways to improve

- Generate for use as a transport layer
- Let your users generate the code
- Think of the user experience
- Don’t generate it

When Not Too

- Tightly coupled
- Only a few objects to expose
- Complex logic
- Doesn't improve over a DIY

Questions?

- Gundremmingen_Nuclear_Power_Plant.jpg - Felix König
- https://commons.wikimedia.org/wiki/File:Gundremmingen_Nuclear_Power_Plant.jpg
- Creative Commons Attribution-Share Alike 3.0 Unported

- lego-hazmat.jpg - Jeff Eaton
- https://www.flickr.com/photos/jeffeaton/6586676977
- CC BY-SA 2.0

- Gordon_Dam.jpg - JJ Harrison
- https://commons.wikimedia.org/wiki/File:Gordon_Dam.jpg
- Creative Commons Attribution-Share Alike 3.0 Unported

- Windmills_D1-D4_(Thornton_Bank).jpg - Hans Hillewaert
- https://commons.wikimedia.org/wiki/File:Windmills_D1-D4_(Thornton_Bank).jpg
- Creative Commons Attribution-Share Alike 4.0 International

- Solar_panels_and_sky_-_Smaack_-_201206.jpg - Smaack
- https://commons.wikimedia.org/wiki/File:Solar_panels_and_sky_-_Smaack_-_201206.jpg
- Creative Commons Attribution-Share Alike 4.0 International

- Kirk_Mills_water_wheel.JPG - Smallbones
- https://commons.wikimedia.org/wiki/File:Kirk_Mills_water_wheel.JPG
- Creative Commons CC0 1.0 Universal Public Domain Dedication

- Archimedes-Mirror_by_Giulio_Parigi.jpg
- https://commons.wikimedia.org/wiki/File:Archimedes-Mirror_by_Giulio_Parigi.jpg
- Public Domain

- Kuremaa_mõisa_tuuleveski - Ivar Leidus
- https://commons.wikimedia.org/wiki/File:Kuremaa_m%C3%B5isa_tuuleveski.jpg
- Creative Commons Attribution-ShareAlike 3.0 Estonia

- money-roll.jpg - http://401kcalculator.org
- https://www.flickr.com/photos/68751915@N05/6757882889
- Creative Commons Attribution-ShareAlike 2.0

- code-is-poetry.jpg - Alexander Gounder
- https://www.flickr.com/photos/gounder/13060232824
- Creative Commons Attribution 2.0 Generic (CC BY 2.0)

- Icon_New_File_256x256.png
- https://commons.wikimedia.org/wiki/File:Icon_New_File_256x256.png
- Creative Commons CC0 1.0 Universal Public Domain Dedication

- Add_document_icon_(the_Noun_Project_27896).svg - Agus Purwanto, from The Noun Project
- https://commons.wikimedia.org/wiki/File:Add_document_icon_(the_Noun_Project_27896).svg
- Creative Commons Attribution 3.0 United States