@johannes_fiala - Java Forum Stuttgart 2017

(HOW TO) USE SWAGGER TO DEVELOP REST APPLICATIONS

About me…

- Swagger-Codegen Contributor
  - Javascript, Jaxrs-CXF, BeanValidation for Java*, Jaxrs-Resteasy
- Other contributions
  - Spring REST / Swagger-Springfox
    - BeanValidation support
  - Swagger2Markup
    - BeanValidation support

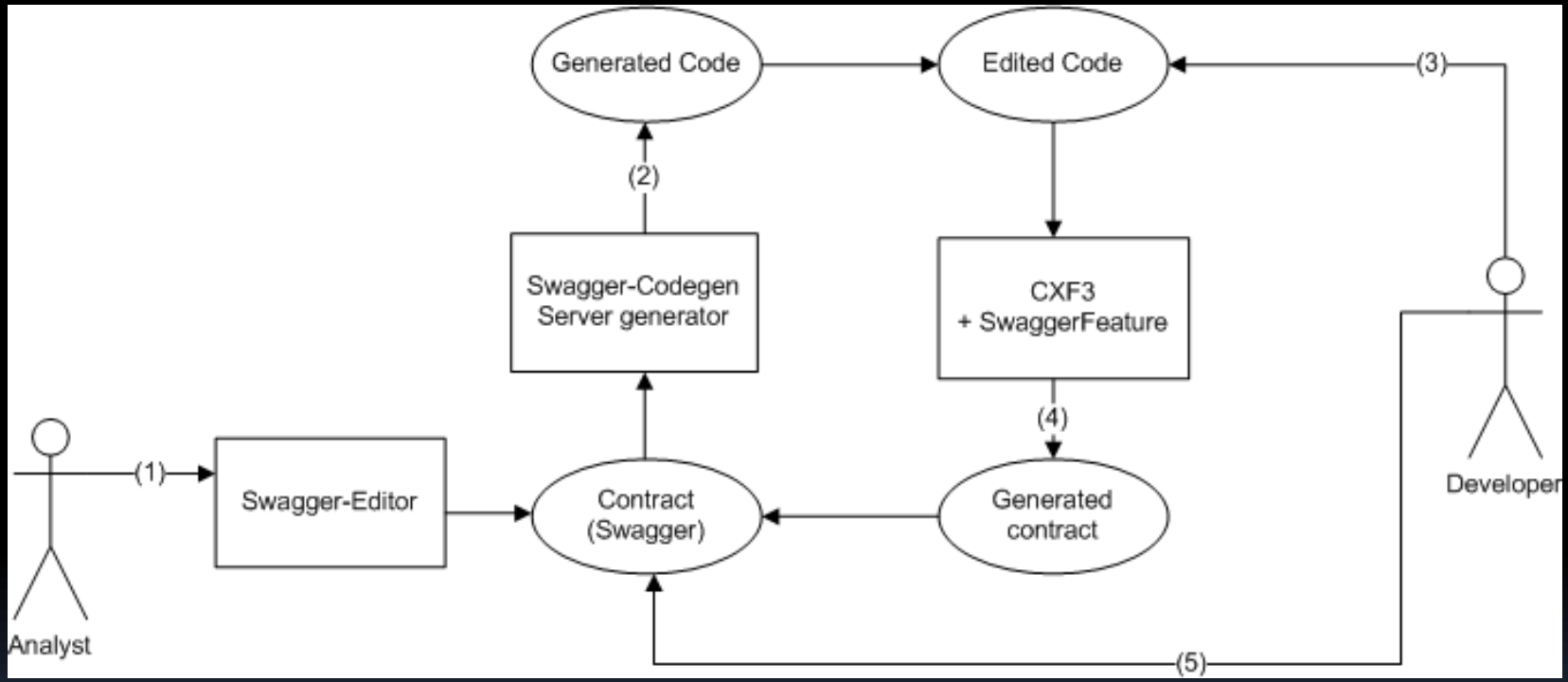
Agenda

- Generate based on contract
- Extend using code first
- Freeze the contract
- Use the REST API
  - Generate client code (Java/Javascript)
  - Access with a browser using a UI
  - View/Share as HTML/PDF
- Customize the code generator

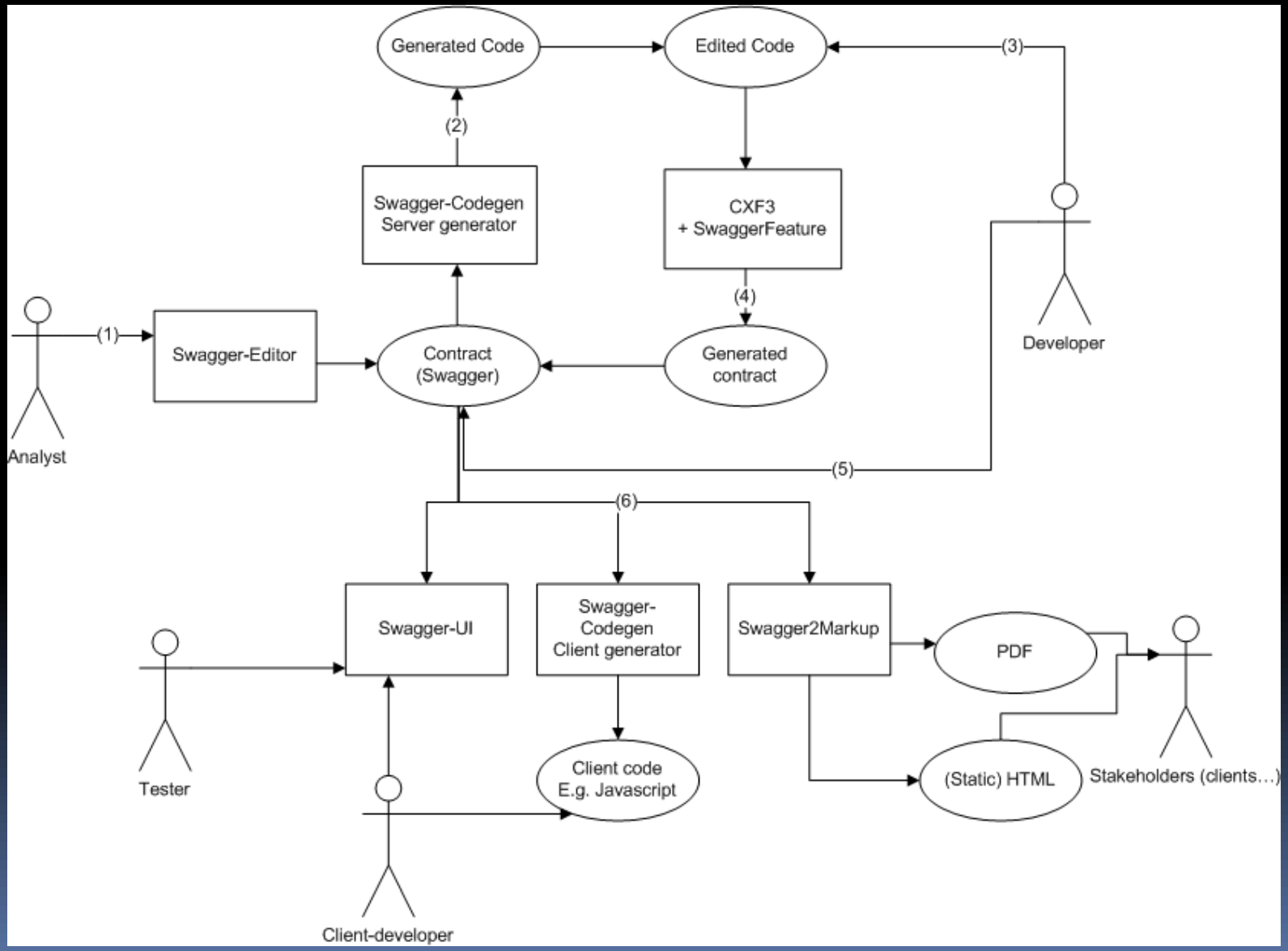
Toolchain

- Apache CXF 3
  - + SwaggerFeature
  - + Spring integration/Spring Boot integration
- Swagger-Tools
  - Swagger-Editor
  - Swagger-Codegen
  - Swagger-UI
  - Swagger2Markup

Contract first, then code, then contract …



Complete Process flow



Contract

- WADL (XML-based)
  - By w3c, Last update 2009
- Swagger (Json/Yaml-based)
  - By Linux foundation
  - Version 1.0 – 2011 (Wordnik)
  - Version 1.2 - 2014
  - Version 2.0 – 2015 / transferred to Linux foundation / Open-API initiative
  - Next version: 3.0 (beta released 2017-03-01)
- Others: Blueprint, RAML, …

Open API / Swagger

- A language-agnostic interface to REST APIs
- allows to discover and understand the capabilities of a service
- Supported Formats: JSON/YAML

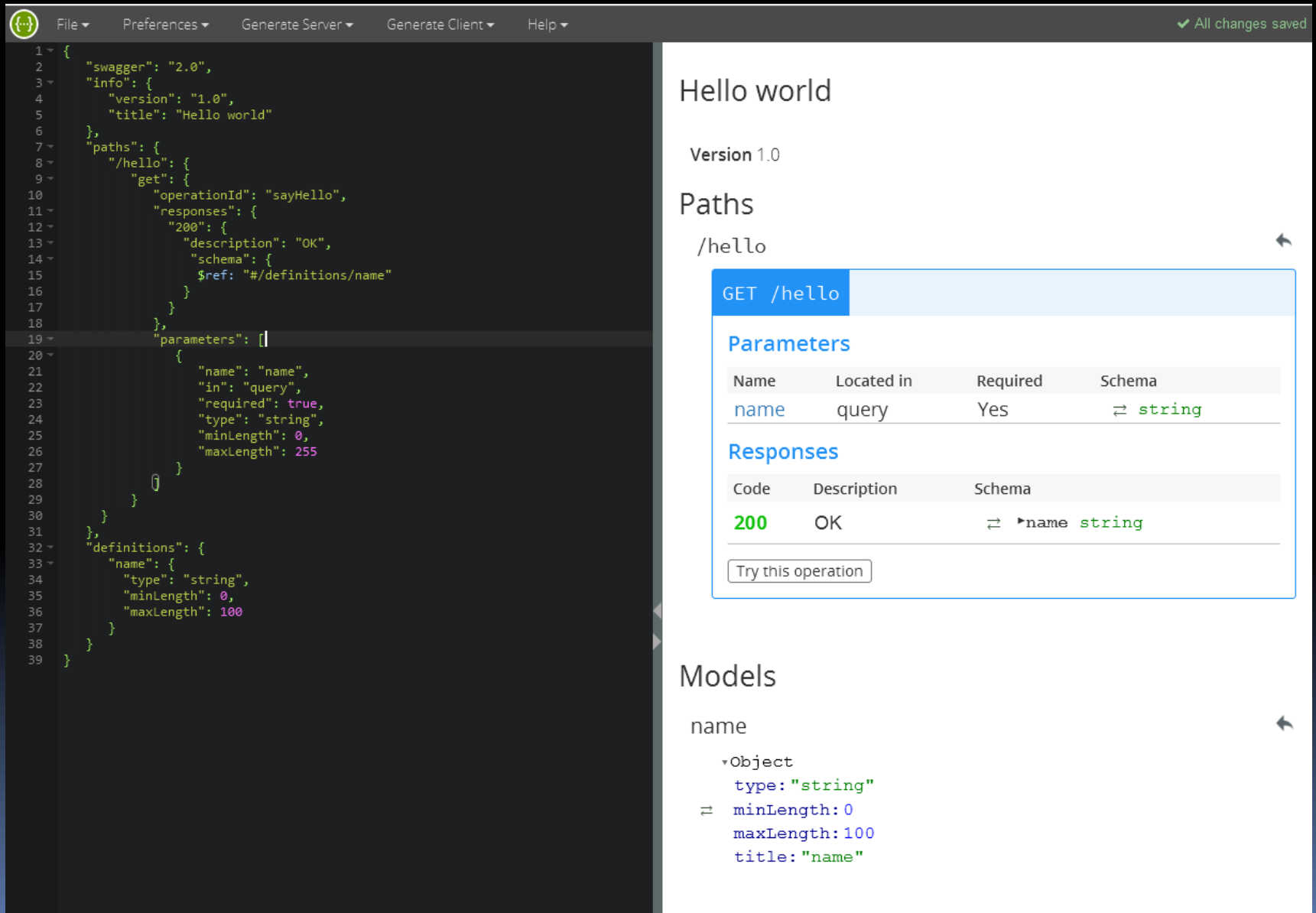
https://github.com/OAI/OpenAPI-Specification

Contract editors

- Swagger Editor
  - by SmartBear
- Eclipse SwagEdit
  - By RepreZen API Studio

- Commercial Tools:
  - Restlet Studio
  - RepreZen API Studio

Swagger-Editor (Json)



Swagger-Codegen

- by SmartBear (Apache License)
- Version 2.2.2
- Java program
  - + Mustache templating
- Generates server + client code

https://github.com/swagger-api/swagger-codegen

Swagger-Codegen Supported Languages

- Client and/or Server available for:
- Android, Asp.net, AsyncScala, Bash, Clojure, ConfluenceWiki, C++, C#, Dart, Elixir, Erlang, Finch, Flash, FlaskConnexion, Go, Groovy, Haskell, Java, CXF, JavaInflector, Jaxrs-Spec, Jersey, JavaMSF4J, Resteasy, Javascript, AngularJS, Jmeter, Lumen, NancyFX, NodeJS, ObjC, Perl, Php, Python, Qt5C++, Rails, Ruby, Scala, Scalatra, Silex, Sinatra, Slim (PHP), Spring, Static Html, Swagger, Yaml, Swift, Tizen, TypescriptAngular, Undertow, …

- And many more libraries (e.g. Java Jersey1, ok-http-gson, …)

Generate the server stub Swagger-Codegen

- Supported server stubs for Java:
  - Java JAX RS
    - 2.0 spec
    - Jersey
    - CXF (Spring/CDI)
    - Resteasy
  - Spring MVC
  - Spring Boot

CXF server stub Features

- Since 2.2.2 (released 2017-03-01)
- Generates Interface/Implementation/Models
- Generate a complete web application
  - Context.xml / ApplicationContext.xml
  - Web.xml
  - Jboss: jboss-web.xml
- Spring Boot support
  - Run as Spring-Boot Application
  - Run using Spring Integrationtests using random ports

CXF server stub Supported Features

- Spring application
  - Swagger generator
  - WADL generator
  - BeanValidation annotations
  - Activate automatic BeanValidation (1.1)
  - Compression (gzip)
  - Logging
  - Integration-Tests (Spring Boot)

Swagger-Codegen CLI

- io.swagger.codegen.Codegen
  -i hello_world.json
  -l jaxrs-cxf
  -o c:\hello_world_project
  -c hello_world_config.json

Demo

- Create hello world service
- Generate CXF server stub (with Spring support enabled)
- Run using Spring Boot
- Run Junit-Tests
- Deploy to application server (Jboss EAP 7)

Further development life cycle

- Extend the API
  - code first
- Freeze the contract
- Use the API
  - Frontend development

Extend the application

- Modify your API:
  - Add new services (use JAXRS-annotations)
  - Use Swagger annotations
  - Use BeanValidation annotations
- Generated Swagger spec gets updated automatically

Extend the application Swagger annotations

- Service:
  - @Api – activate Swagger for api
- Operations:
  - @ApiOperation - description
  - @ApiResponse – error codes + return types
- Model:
  - @ApiModel - description
  - @ApiModelProperty - description

Freeze your API:

- Generate only interfaces/models
- Prevent accidential changes of the API
- Integrate in build job (Maven / Gradle)
- Specify which sources should NOT be generated during build using .swagger-codegen-ignore

- https://github.com/swagger-api/swagger-codegen/tree/master/modules/swagger-codegen-maven-plugin

Freeze your API: maven configuration

```
<!-- re-generate interface/model -->
<plugin>
      <groupId>io.swagger</groupId>
      <artifactId>swagger-codegen-maven-plugin</artifactId>
      <version>2.2.2</version>
      <executions>
          <execution>
              <goals>
                  <goal>generate</goal>
              </goals>
              <configuration>
                  <inputSpec>${project.basedir}/src/main/resources/api/hello_world.json</inputSpec>
                  <language>jaxrs-cxf</language>
                  <sourceFolder>src/gen/java</sourceFolder>
                  <configOptions>
                      <useBeanValidation>true</useBeanValidation>
                  </configOptions>
                  <addCompileSourceRoot>false</addCompileSourceRoot>
                  <ignoreFileOverride>${project.basedir}/.swagger-codegen-ignore</ignoreFileOverride>
                  <output>${project.basedir}</output>
              </configuration>
          </execution>
      </executions>
  </plugin>
```

Demo

- Extend hello world service (+ BeanValidation)
- Access updated specs
- Freeze the contract

Use your API

- Generate client stubs
  - Swagger-Codegen
- Access your API using a browser
  - Swagger-UI
- Generate HTML/PDF documentation
  - Swagger2Markup

Why generate client code?

- No more manual api calls
- Ensure consistency of your client code with the API!
  - Makes code completion possible!
- Allows developers to read description for your operations and models in the IDE
- You get compilation errors if the API breaks with newer versions!

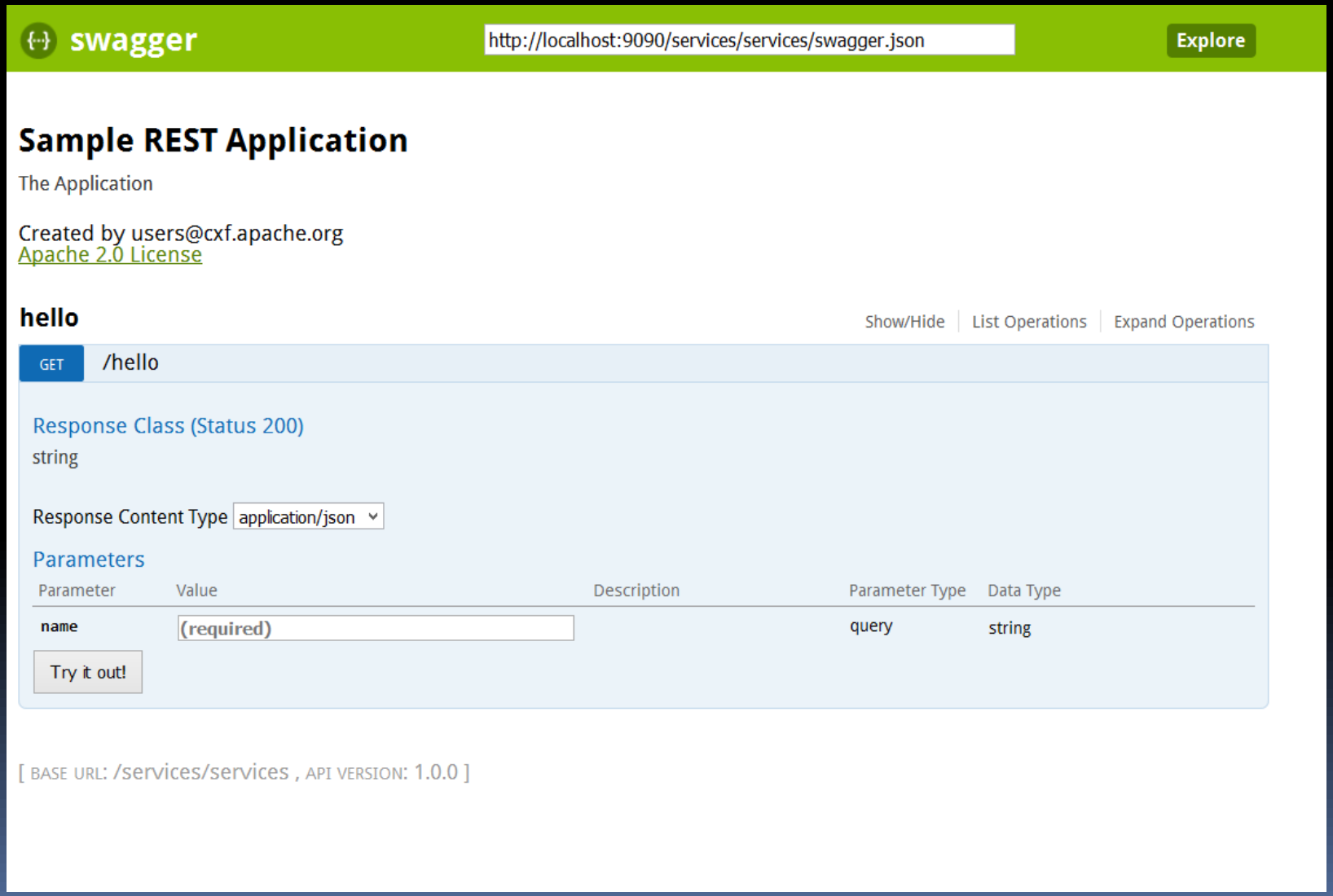
Swagger-UI

- By SmartBear
- Access your API with a browser
- Javascript application
- Can access the generated Swagger spec – always consistent with your code

- Integration options:
  - Copy into your webapp
  - load as Web-Jar

https://github.com/swagger-api/swagger-ui

Swagger-UI


swagger
http://localhost:9090/services/services/swagger.json
Explore
Sample REST Application
The Application
Created by users@cxf.apache.org
Apache 2.0 License
hello
Show/Hide
List Operations
Expand Operations
GET
/hello
Response Class (Status 200)
string
Response Content Type
application/json
Parameters
Parameter
Value
Description
Parameter Type
Data Type
name
(required)
query
string
Try it out!
[BASE URL: /services/services , API VERSION: 1.0.0]

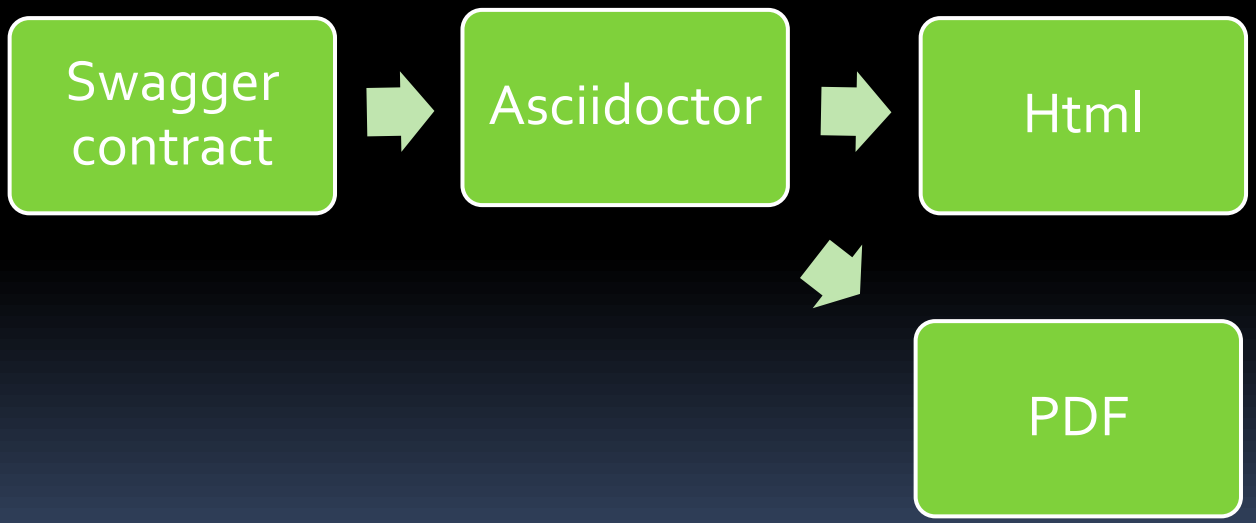
Swagger2Markup

- By Robert Winkler (github)
- Render your API in HTML/PDFs
- Uses Markdown/Asciidoctor files
- Completely customizable

- Integration options:
  - Run as Program/Unittest
  - Maven

https://github.com/Swagger2Markup/swagger2markup

Swagger2Markup



Demo

- Client stub generator:
  - Java
- Swagger-UI
- Swagger2Markup

Customize the generator

- Generator implemented in Java (one class for each language)
- Mustache-files
  - api.mustache
  - apiServiceImpl.mustache
  - pojo.mustache
  - api_test.mustache
  - …

Customize the generator

- Use –t flag for your own templates
- Customize only the templates you need
- Examples:
  - Add Maven profile for deployment
  - Add logger declaration
  - Customize generated unit tests
  - …

Customize the generator

- Customize Codegen Languages
  - Extend Language class
  - Add it to io.swagger.codegen.CodegenConfig
    - swagger-codegen\src\main\resources\META-INF\services\io.swagger.codegen.CodegenConfig
  - Copy language templates

WADL?
From WADL to Swagger

- Use wadl2java to generate server stub
  - BeanValidation: krasa-jaxb-tools
- Activate CXF3 SwaggerFeature
- Use generated Swagger-file
  - Will include BeanValidation annotations for models

Swagger-Codegen – What‘s next for CXF

- Builder pattern for models:
  - return new Response().name(“Foo").first(“Test”);
- Cascaded BeanValidation: @Valid
- Add snippets for Multipart
- Optionally use JaxRs-Response instead of return datatypes
- ???

Wrapup

- Generate based on contract
  - Swagger-Codegen server stubs
- Extend using code first
  - CXF 3 Swagger Feature
- Freeze using contract
  - Swagger-Codegen build integration (mvn/gradle/cmd)
- Use your application
  - Generate client code (Swagger-Codegen)
  - Use in browser (Swagger-UI)
  - View/Share as HTML/PDF (Swagger2Markup)
- Customize the code generator

Contribute to the projects

- Open API
  - Join the technical developer community
- Swagger-Codegen
  - Java / JMustache
- Swagger-UI
  - Javascript
- Swagger-Editor
  - Javascript
- Swagger2Markup
  - Java/Asciidoctor

Thank you for your attention!

- Demo-Code: http://github.com/jfiala/swagger-cxf-demo
- Contact:
  - @johannes_fiala

Q & A / 1

- Which framework is used for the REST invocation?
  - The framework used for the invocation is depending on the language (e.g. java currently jersey is used, cxf-client uses cxf etc.)
- Which datatypes can be used?
  - Swagger is language-agnostic; to quickly find out the mappings for the language used, you can use e.g. the SwaggerFeature of CXF server
- How long does it take to write/customize a language generator
  - Depending on your Java/mustache skills you should be able to customize within a few hours, setting up a new language will take somewhat longer.

Q & A / 2

- Why do I have to pay for Swagger-Hub services
  - Swagger-Codegen is a Apache licensed open source project and open to contributions, it is free of charge.
    The services offered by SwaggerHub are provided by SmartBear for running your project and provide tools for collaboration and integration.

Links & Resources

- Swagger Editor
  - http://editor.swagger.io/
- Swagger Codegen
  - https://github.com/swagger-api/swagger-codegen
- Swagger UI
  - https://github.com/swagger-api/swagger-ui
- CXF
  - http://cxf.apache.org/