Document your REST-API and generate client-code using Swagger and Spring

@

In the next few minutes you‘ll learn how to...

- provide documentation for your REST-API
- access your API with a browser using Swagger UI
- generate client code for accessing your API
- get a nice picture of your API services & models

Swagger

- A language-agnostic interface to REST APIs
- allows to discover and understand the capabilities of a service
- Swagger removes the guesswork in calling the service
- Swagger renders the API docs in JSON syntax

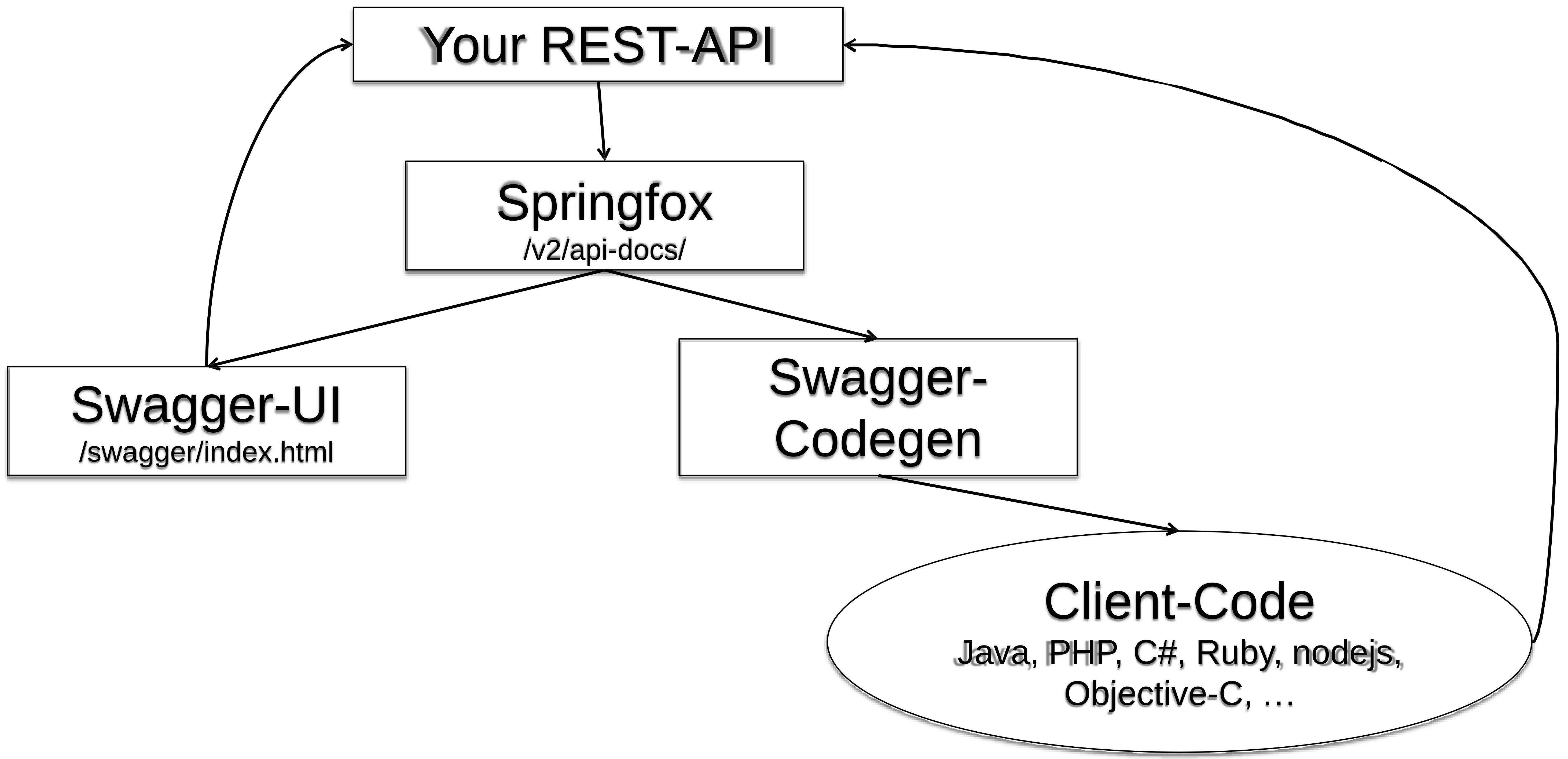
“A standardized way to describe an API in JSON syntax”

Source: https://github.com/swagger-api/swagger-spec

Technology stack

- Spring
  - Spring Boot for fast start
  - Also possible for Spring REST applications
- Springfox
  - For providing the dynamic api-docs
- Swagger UI
  - For accessing the api using a browser
- Swagger Codegen
  - For generating client code stubs

Big Picture


Your REST-API
Springfox
/v2/api-docs/
Swagger-UI
/swagger/index.html
Swagger-Codegen
Client-Code
Java, PHP, C#, Ruby, nodejs, Objective-C, …

Springfox

- Provide complete api-docs for every @RESTController
  - Services
  - Supported Verbs (GET/POST/…)
  - Request parameters/body
  - Response codes + body
- Many customization options (hide attributes, custom data types, …)

Swagger UI

- Javascript application for accessing your complete REST API using a browser
- Lists all services directly from the (dynamic) api-docs
- Always consistent with your API!

Swagger Codegen

- Client code stub generator (commandline)
- Generates client stubs (customizable!)
- Supported languages:
  - Java, C#, Dart, Flash, Groovy, JaxRS, NodeJS, Objective-C, Perl, PHP, Python, Ruby, Scala, ….

Why generate client code?

- Ensure consistency of your client code with the API!
- Makes code completion possible!
  - For service methods & model classes
- 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 Codegenerator

- Code generation templates:
  - DefaultCodegenerator.generate():
    - Scans Model Properties first
    - Then compliles the Mustache templates
  - Language specifics:
    - io.swagger.codegen.languages.*
    - Mustache files

Wrapup

- Springfox: provide api-docs
  - Completely dynamic & Always consistent
- Swagger-UI: access api-docs using browser
  - make your api-docs easily acessible for testers/developers/…
- Swagger-Codegen: generate client stubs
  - Get code completion
  - Keep your clients in sync
- Swagger.Ed: display your API graphically

Enable Swagger/Springfox

- Add dependencies
- Add @EnableSwagger2 to Application
- Run your spring application

Springfox annotations

- Controller:
  - @Api
- Operations:
  - @ApiOperation – describe your service
  - @ApiResponse – Define error codes
- Model:
  - @ApiModel(„description“)
  - @ApiModelProperty: description + required-flag
  - @JsonIgnore

Swagger-UI

- Ship together with your REST-service
- Makes accessing your services easy
- Protect the Swagger-UI if applicable

Use Swagger Codegen

- Call using the commandline
- Integrate with your build environment (e.g. using Ant)

Swagger.Ed

- Chrome Plugin for visually displaying you API
- Allows users to graphically explore through your services & models
- Nice for „Big picture“ of your API

Who is using Swagger?

- Paypal
- Microsoft
- Amazon AWS API gateway:
  - http://swagger.io/getting-started-with-the-amazon-swagger-importer/
  - You can import Swagger definitions here
  - You can manage your services using Amazon AWS then

Links & Resources

- Swagger.io
  http://swagger.io
- Swagger.io integrations
  http://swagger.io/open-source-integrations/

- Springfox
  http://springfox.github.io/springfox/

Links & Resources

- Swagger UI
  https://github.com/swagger-api/swagger-ui
- Swagger Codegen
  https://github.com/swagger-api/swagger-codegen
- Chrome Plugin Swagger.ed
  https://github.com/chefArchitect/apispots-browser-swaggered