

Add Some Swagger to Your REST API

“The World's Most Popular API Framework” – swagger.io



SWAGGER

Kevin Johnson

 @AmplitudeDesign

 khjohnson1972

www.amplitudedesign.com

Freelance Developer Since 2002

Started in Development in 1997

Live In Madison, WI

<https://joind.in/talk/42578>



What is ***SWAGGER***





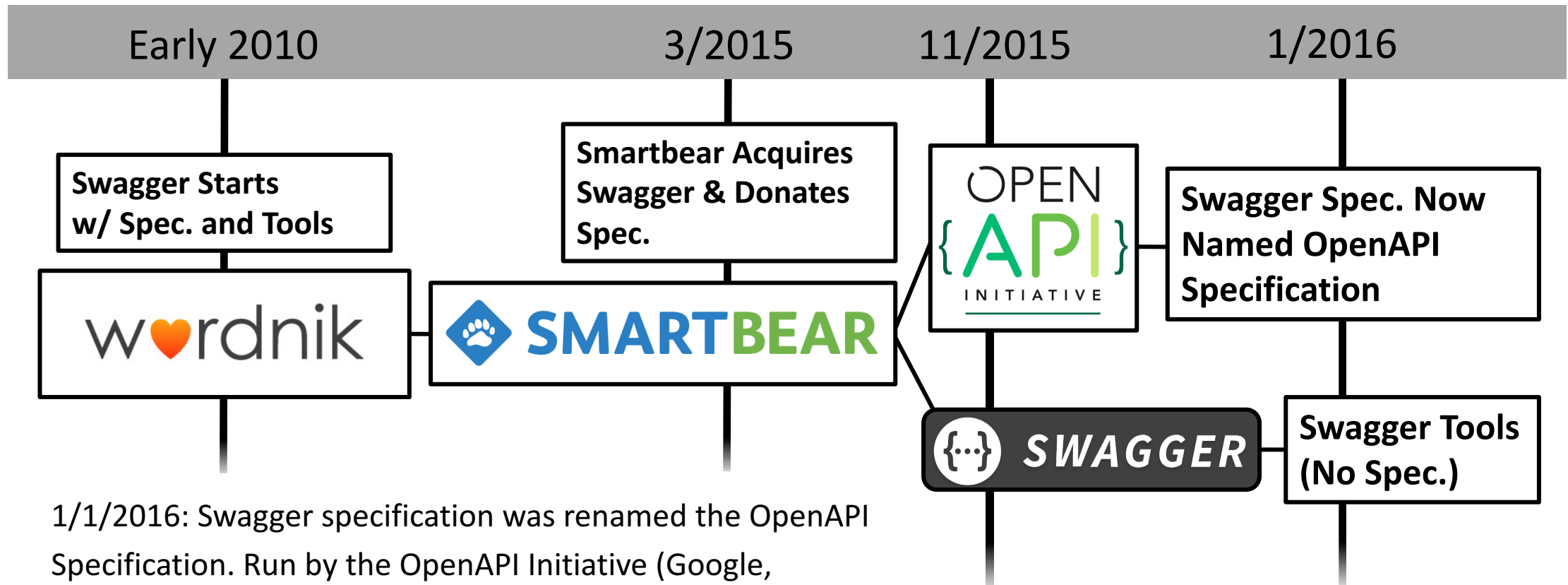
SWAGGER

1. WHY? 2. WHAT? 3. HOW?

DISCLAIMER:

- I will be scratching the surface.
- I will not speak about the subtleties of an API.
- This talk is merely an introduction.
- I am still learning myself...
- I have been working with Swagger for two years to document APIs.
- I AM NOT A DESIGNER. ANY OF MY WORK YOU SEE HERE IS POOR DESIGN.

Intro **SWAGGER**



1/1/2016: Swagger specification was renamed the OpenAPI Specification. Run by the OpenAPI Initiative (Google, Microsoft, Linux Foundation: <https://openapis.org>)

Why Use *SWAGGER*



“An API is only as good as its documentation.”

Undocumented **API** calls might as well **not exist**.

APIs are **hard to test**, especially if you're not part of the development team.

An API isn't tied directly to a user interface.

No documentation, no way of sharing your API.

Why Use **SWAGGER**

- The underlying **OpenAPI Specification** is **language-agnostic**.
- As your API grows, along with your codebase, it becomes **hard to recollect details**.
- Swagger **can be used independently** from your code base.
- Swagger is **lower risk, high reward**.
- For the next developer...
- 5,000+ Github Repos, 2,500+ Contributors, 25+ Programming Languages, 10M+ Downloads

Why Use **SWAGGER**

- The Swagger UI framework allows both developers and non-developers to **interact with the API in a sandbox UI that gives clear insight into how the API responds to parameters and options.** Swagger may utilize both JSON and XML. (<https://github.com/OAI/OpenAPI-Specification/wiki>)
- **Swagger-Codegen contains a template-driven engine to generate documentation, API clients and server stubs** in different languages by parsing the OpenAPI/Swagger definition. (<https://github.com/swagger-api/swagger-codegen>)

What is **SWAGGER**

Tools @ <http://swagger.io> or <https://github.com/swagger-api>:

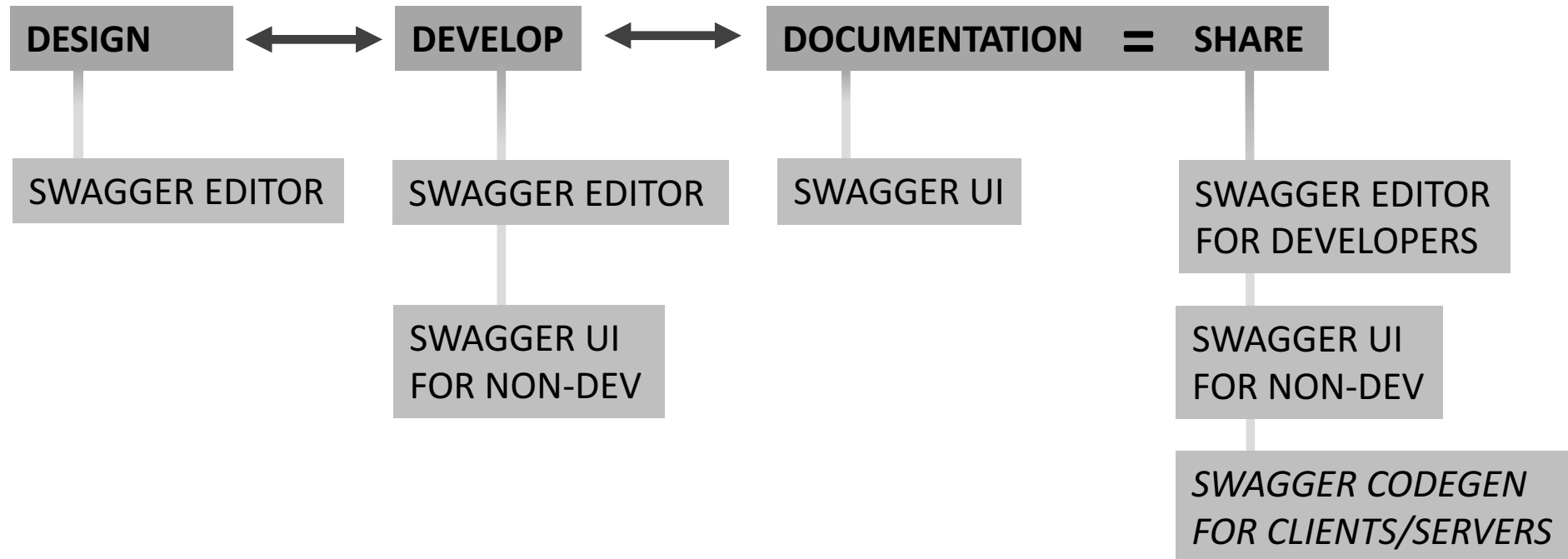
1. **Swagger Editor** for **development** and **validation** of Swagger files.
2. **Swagger Codegen** for generating clients (consumers), servers and documentation. Commonly used in the command line. Can be used within Swagger Editor.
3. **Swagger UI** for distributing your documentation.
4. *There are other commercial and non-commercial projects.*

Swagger maintained tools use Angular and handlebars.js for templating.

Swagger Editor includes working examples to help get you started. See, File > Open Example.

What is **SWAGGER**

COVERS MOST ASPECTS OF API DEVELOPMENT



Before We Dive In



SWAGGER

SUPPORTS:

METHOD:

POST

GET

PUT

PATCH

DELETE

ACTION:

Create

Read

Update/Replace

Update/Modify

Delete

AUTHENTICATION:

Basic

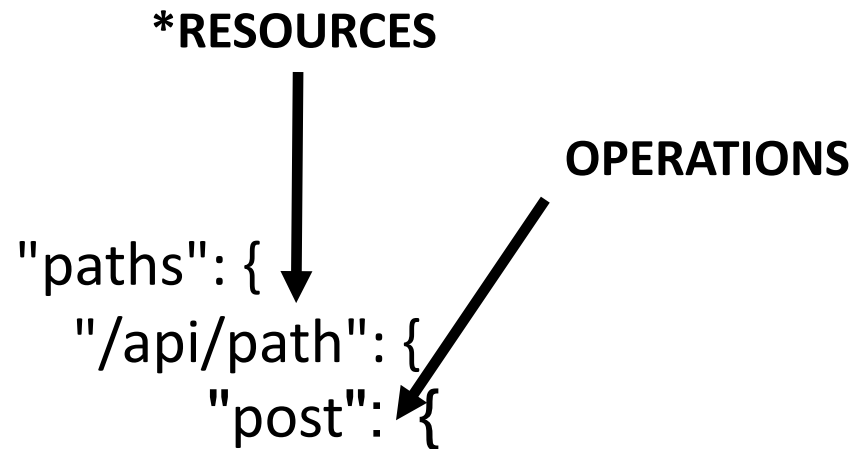
apiKey

oauth2

[More info @RestAPITutorial](#)

Before We Dive In **SWAGGER**

PATH/RESOURCE DEFINITIONS:



* Resources (paths/routes) can use the notation `/users/{id}`, which I won't show.

How Can You Use **SWAGGER**

APPROACH 1:

TOP DOWN OR DESIGN FIRST APPROACH:

Start building a Swagger file with the Swagger Editor, typically swagger.json or swagger.yml.

- It's human readable, and uses the : <http://json-schema.org>
- Uses the \$ref for linking to external definitions (do not repeat yourself), if you want to break up your swagger file.
- You can reference external files using \$ref: http:\\\\www.url.com/path-to-file

How Can You Use ***SWAGGER***

APPROACH 2:

BOTTOM UP OR CODE FIRST APPROACH:

You have an existing API or are writing your code first.

Write your Swagger file to match your existing API

How Can You Use **SWAGGER**

APPROACH 3:

**BOTTOM UP OR CODE FIRST APPROACH WITH DOCTRINE ANNOTATIONS
AND [ZIRCOTE/SWAGGER-PHP](https://github.com/zircote/swagger-php):**

- Uses doctrine annotations to construct the Swagger file.
- Uses composer installed bin and Doctrine Annotations.
- See /Examples/petstore.swagger.io in project.

```
$: php ../../bin/swagger -o swagger.json -b swagger-v2.php
```

Swagger UI



SWAGGER

- Depends upon node for building from source during development.
- Customizing Swagger UI.
- npm install
- npm run build
- "src/ files are written to dist/"
- * nice example <https://api.tfl.gov.uk/>

Other Tools/Options

- **RAML – Licensed IDE (seems more robust then swagger-editor) – YAML for definitions** – “Because of the way RAML is designed, it can support REST API documentation in addition to documentation for APIs that don’t precisely or strictly adhere to REST standards such as those utilizing other design architectures like [SOAP](#) and RPC. This makes it a wonderful documentation language for languages that aren’t classically designed, or are designed to take advantage of the extensive modified frameworks available to API developers.”
- **API Blueprint (“dwarfed in comparison to RAML and Swagger”)**
- **Apiary.io** (supports [API Blueprint](#) and [OpenAPI](#) (Swagger) formats)
- <https://www.mashery.com/api-programs>
- <https://www.3scale.net/api-management/interactive-api-documentation/> (SASS that can use swagger files)
- Npm bootstrap w/ bootstrap-swagger – generate html documentation from swagger file. (bootstrap swagger swagger.json bootstrap/)

Other Tools/Options

- <https://gelato.io/> (import your Swagger files)
- <https://www.dreamfactory.com>
- <https://apigee.com/about/developers> (Support OpenAPI/Swagger, and can transform SOAP to REST)
- Apigility.org - Has an adaptor for Swagger documentation. Zend Framework includes UI to create APIs using JSON (HAL). Works with REST and RPC.
- Swagger-php – **blended approach** (create swagger files using composer bin and Doctrine Annotations, see Examples in project) - `php ../../bin/swagger -o swagger.json -b swagger-v2.php`
- [Reprezen](#) - An Eclipse based IDE for API Design/Development
- SOAP WSDL (Swagger On Another Planet)
- SmartBear - <https://smartbear.com/product/swaggerhub/overview/>
- Postman, etc...

Is Swagger Just WSDL for the Web?

- Maybe, but way more useful.
- When would this not work?
- What are others using or doing?

Pros/Cons to Swagger, RAML, API Blueprint

From <http://nordicapis.com/top-specification-formats-for-rest-apis/>

Swagger (OpenAPI)

Pros: Heavily adopted, large community of users and supporters, greater support for multiple languages.

Cons: Lacks advanced constructs for metadata

RAML

Pros: Supports advanced constructs, decent adoption, human readable format, high industry backing (non-traditional REST API structures)

Cons: Lacks code-level tooling, still unproven long-term

API Blueprint

Pros: Easy to understand, simple to write

Cons: Low adoption, lacks advanced constructs in general, complex installation

Installing Tools



SWAGGER

```
wget https://github.com/swagger-api/swagger-editor/releases/download/v2.10.4/swagger-editor.zip
```

```
unzip swagger-editor.zip
```

Resources/Tuts



SWAGGER

- **MY EXAMPLE swagger.json**

<https://github.com/khjohnson1972/SwaggerExample>

- **GREAT 7 PART TUTORIAL**

<https://apihandyman.io/writing-openapi-swagger-specification-tutorial-part-1-introduction/>

- <https://app.swaggerhub.com/help/tutorials/writing-swagger-definitions>

Kevin Johnson

 @AmplitudeDesign

 khjohnson1972

www.amplitudedesign.com

Freelance Developer Since 2002

Started in Development in 1997

Live In Madison, WI

<https://joind.in/talk/42578>