[개정판] Spring Boot를 이용한 RESTful API 개발

{ REST API } + Spring Boot

프로필

Dowon Lee

수강생 19K 강의평점 4.8(1K)

멘토링 ON

멘토링 설정

- 홈
- 강의
- 로드맵
- 수강평
- 게시글
- 블로그

수정하기

이 되었던 때가 있었습니
"IT 엔지니어"라고 적고

을 가지고 있습니다. 누구
사람입니다. 개발자로써, 강
지만, 그래도, 남들보다 조

tecture, Blockchain,

강의 전체 6

[개정판] 웹 애플리케이션 개발을 위한 IntelliJ IDEA 설정

학습중

0강 / 25강 (0%)

818명

멀티OS 사용을 위한 가상화 환경 구축 가이드 (Docker + Kubernetes)

학습중

0강 / 16강 (0%)

924명

Jenkins를 이용한 CI/CD Pipeline 구축

학습중

81강 / 83강 (98%)

독점 2709명

Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)

학습중

156강 / 156강 (100%)

독점 5531명

Spring Boot를 이용한 RESTful Web Services 개발

학습중

3강 / 48강 (6%)

독점 3862명

[구버전] 웹 애플리케이션 개발을 위한 IntelliJ IDEA 설정 (2020 ver.)

학습중

14강 / 14강 (100%)

4813명

목차

0. Section

Web Service와 Web Application

- Web Service 개요
- Web Application 개요
- SOAP vs REST

Web Service & Web Application

- Web Service

  "A *service* offered by an electronic *device* to another electronic *device*, *communicating* with each other via the *World Wide Web*"

  네트워크 상에서 서로 다른 종류의 컴퓨터들 간에 상호작용하기 위한 소프트웨어 시스템

  port over a network, *serving web documents* (HTML, JSON, XML, images), and *creating web applications services*, which serve in solving specific domain problems over the *Web* (WWW, Internet, HTTP)"

- 3 Keys

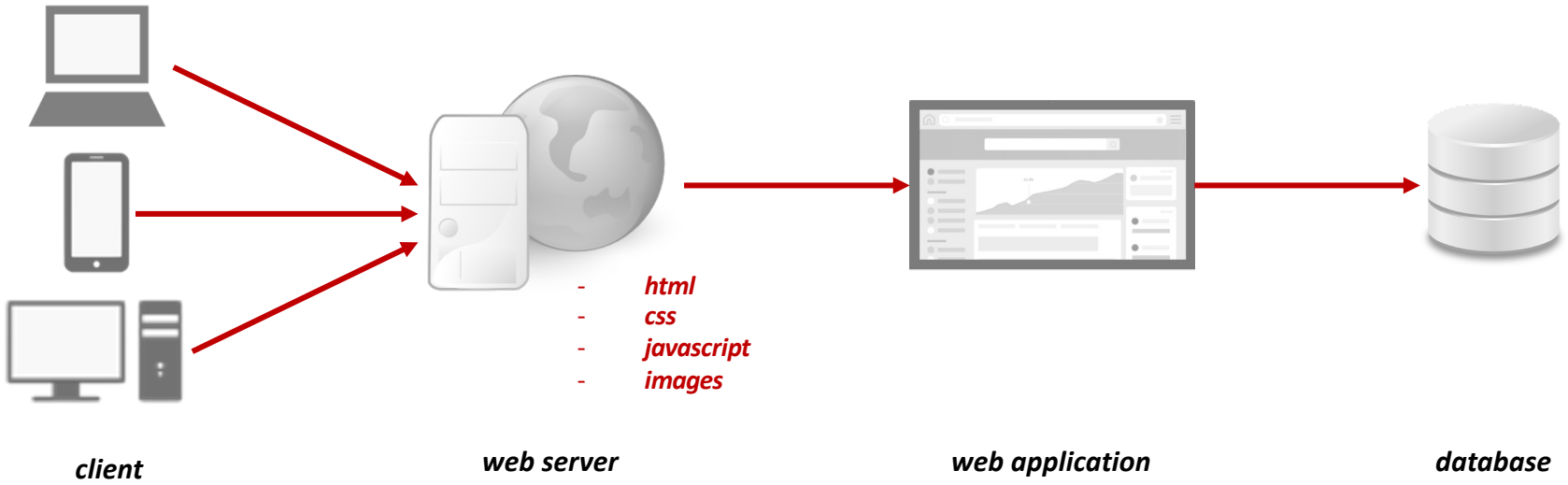
  - Designed for machine-to-machine (or application-to-application) interaction
  - Not platform dependent
  - Allow communication over a network

Web Service & Web Application

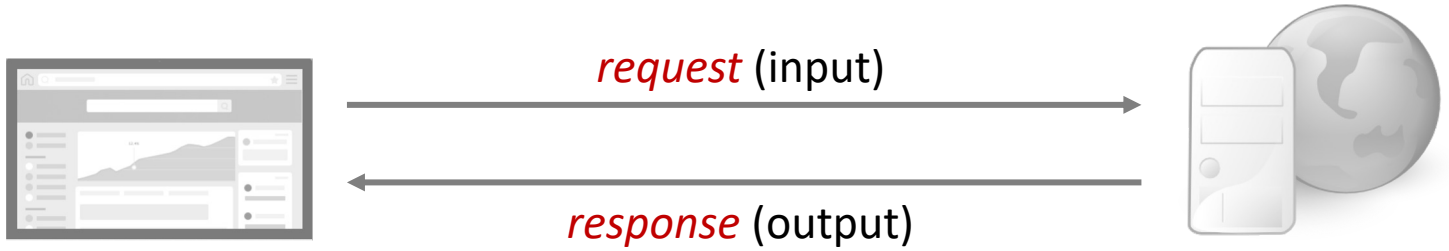
- Web Application

  "An application *program* that is stored on a *remote server* and delivered over the *Internet* through a *browser* interface"

  "Common web applications include *webmail*, online *retail sales*, online *banking*, and online *auctions*"



Web Service



- Service Definition
  - Request/Response Format
  - Request Structure
  - Response Structure
  - Endpoint (URL ...)

- Data exchange 2 formats

</> {...}

- https://www.json.org/json-ko.html

JSON 개요

العربية Б ъ л г а р с к □ 中文 Český Dansk Nederlands English Esperanto Français Deutsch Ε λ λ η ν ι κ ά עברית Magyar Indonesia Italiano
日本 한국어 فارسی Polski Português Română Р у с с к □ и С р п с к о - х р в а т с к □ Slovenščina Español Svenska Türkçe У к р а ї н с ь к а Tiếng Việt

ECMA-404 The JSON Data Interchange Standard.

JSON (JavaScript Object Notation)은 경량의 DATA-교환 형식이다. 이 형식은 사람이 읽고 쓰기에 용이하며, 기계가 분석하고 생성함에도 용이하다. JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999의 일부에 토대를 두고 있다. JSON은 완벽하게 언어로 부터 독립적이지만 C-family 언어 - C, C++, C#, Java, JavaScript, Perl, Python 그외 다수 - 의 프로그래머들에게 친숙한 관습을 사용하는 텍스트 형식이다. 이러한 속성들이 JSON을 이상적인 DATA-교환 언어로 만들고 있다.

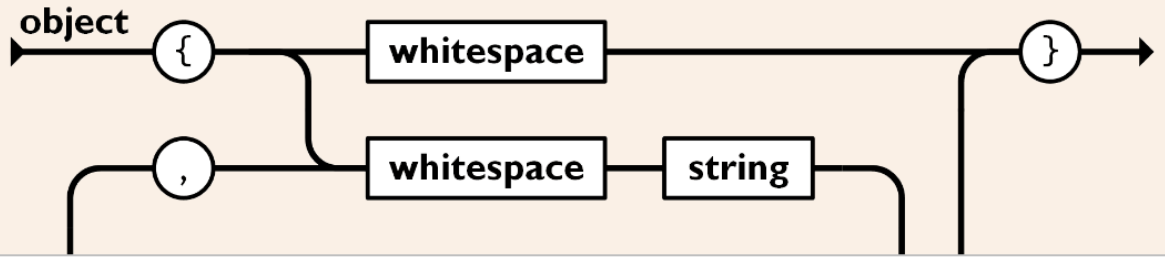
JSON은 두개의 구조를 기본으로 두고 있다:

name/value 형태의 쌍으로 collection 타입. 다양한 언어들에서, 이는 *object*, record, struct(구조체), dictionary, hash table, 키가 있는 list, 또는 연상배열로서 실현 되었다.
값들의 순서화된 리스트. 대부분의 언어들에서, 이는 *array*, vector, list, 또는 sequence로서 실현 되었다.

이러한 것들은 보편적인 DATA 구조이다. 사실상 모든 현대의 프로그래밍 언어들은 어떠한 형태로든 이것들을 지원한다. 프로그래밍 언어들을 이용하여 호환성 있는 DATA 형식이 이러한 구조들을 근간에 두고 있는 것은 당연하다.

JSON 에서, 이러한 형식들을 가져간다:

*object*는 name/value 쌍들의 비순서화된 SET이다. object는 **{** *좌 중괄호*로 시작하고 **}** *우 중괄호*로 끝내어 표현한다. 각 name 뒤에 **:** *colon*을 붙이고 **,** *comma*로 name/value 쌍들 간을 구분한다.




```
json
    element

value
    object
    array
    string
    number
    "true"
    "false"
    "null"

object
    '{' ws '}'
    '{' members '}'

members
    member
    member ',' members

member
    ws string ws ':' element

array
    '[' ws ']'
    '[' elements ']'

elements
```

Web Service

request (input)

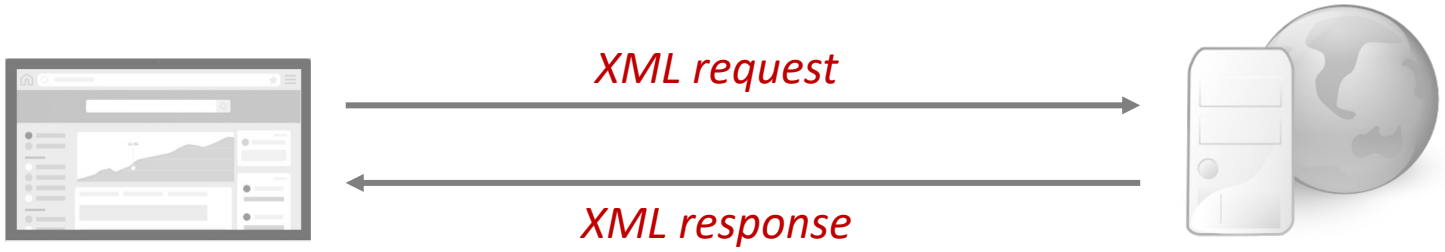
response (output)

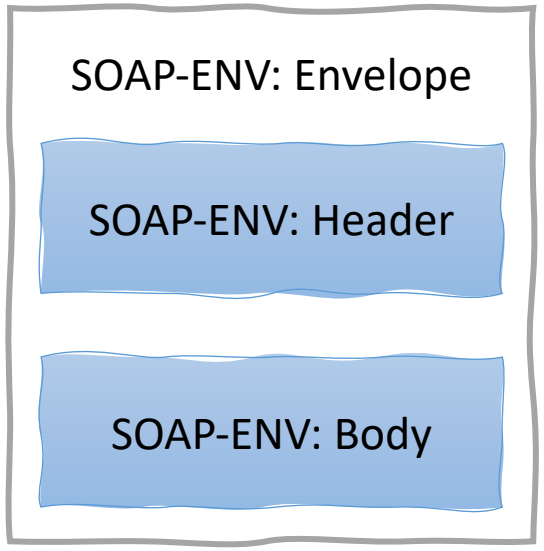
Service consumer
or
Client

Service provider
or
Server

SOAP

- SOAP (Simple Object Access Protocol)



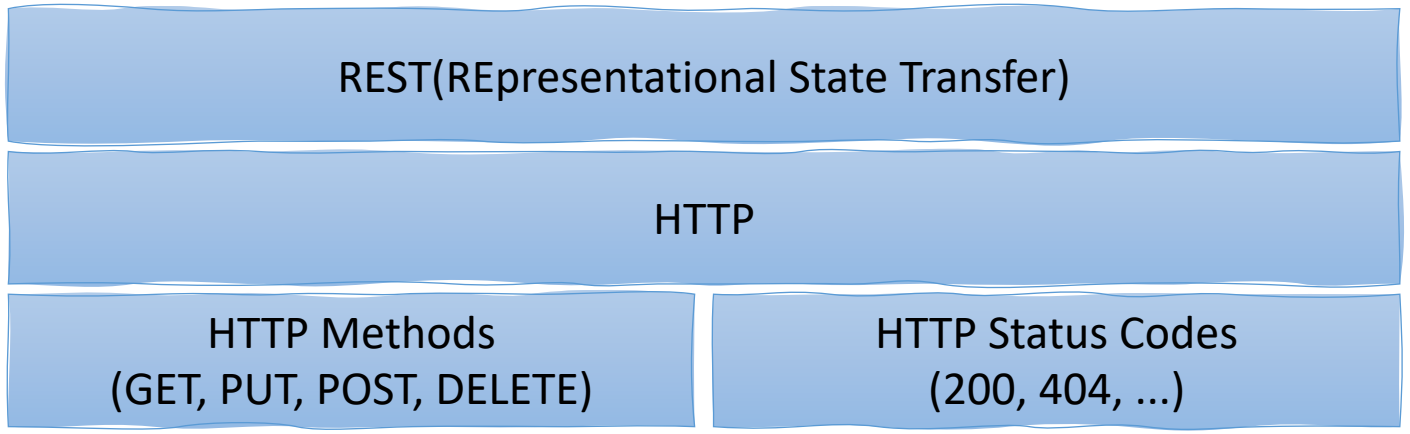



```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:c="http://www.acmeOrders.com/OrderService"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <c:OrderMessage>
            <localElement>
                <FirstName>John</FirstName>
                <LastName>Smith</LastName>
                <Street>High Street</Street>
                <City>London</City>
                <ZipCode>W1A1AA</ZipCode>
                <PartNumber>ABC1234</PartNumber>
                <Quantity>1</Quantity>
            </localElement>
        </c:OrderMessage>
    </soap:Body>
</soap:Envelope>
```

RESTful

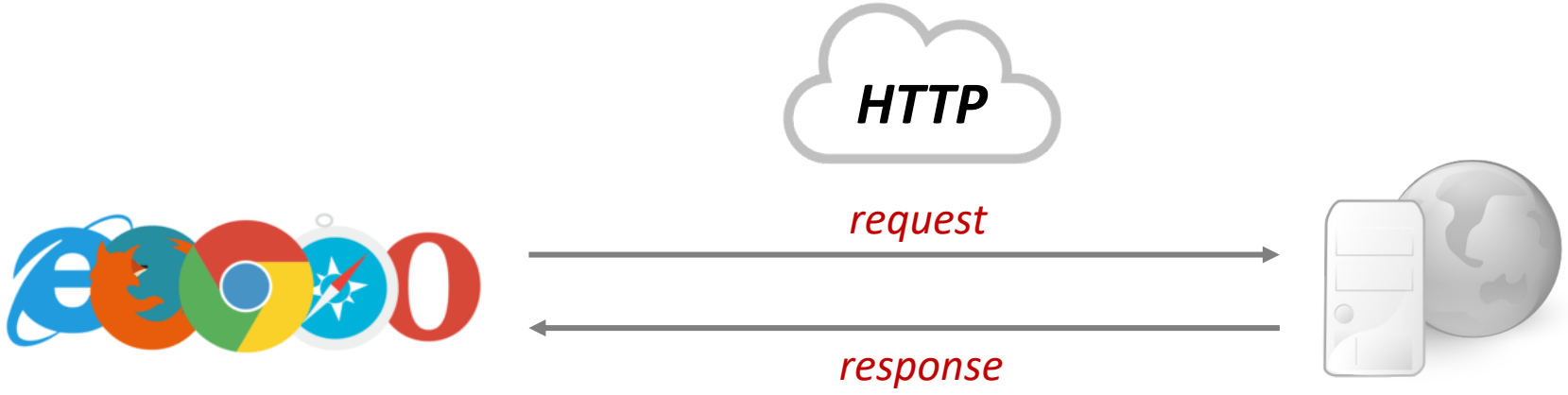
- REST (REpresentational State Transfer)
  - Resource의 Representation에 의한 상태 전달
  - HTTP Method를 통해 Resource를 처리하기 위한 아키텍쳐

- RESTful
  - REST API를 제공하는 웹 서비스



MAKE BEST USE OF HTTP

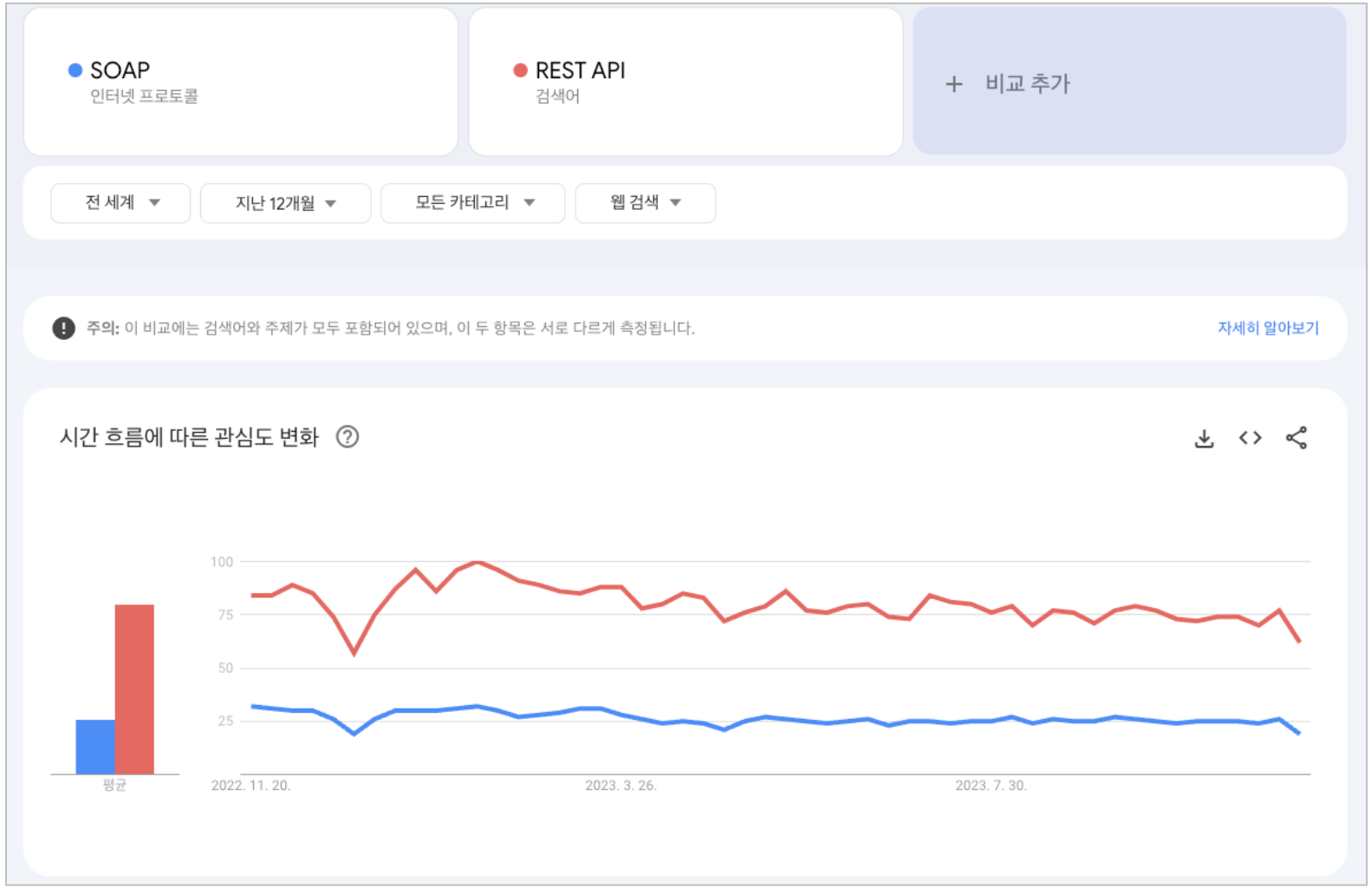
RESTful



- Resource
  - URI (Uniform Resource Identifier), 인터넷 자원을 나타내는 유일한 주소
  - XML, HTML, JSON
- Endpoint
  - API를 통해 서버가 제공하는 리소스에 접근하기 위해서 제공 되는 주소

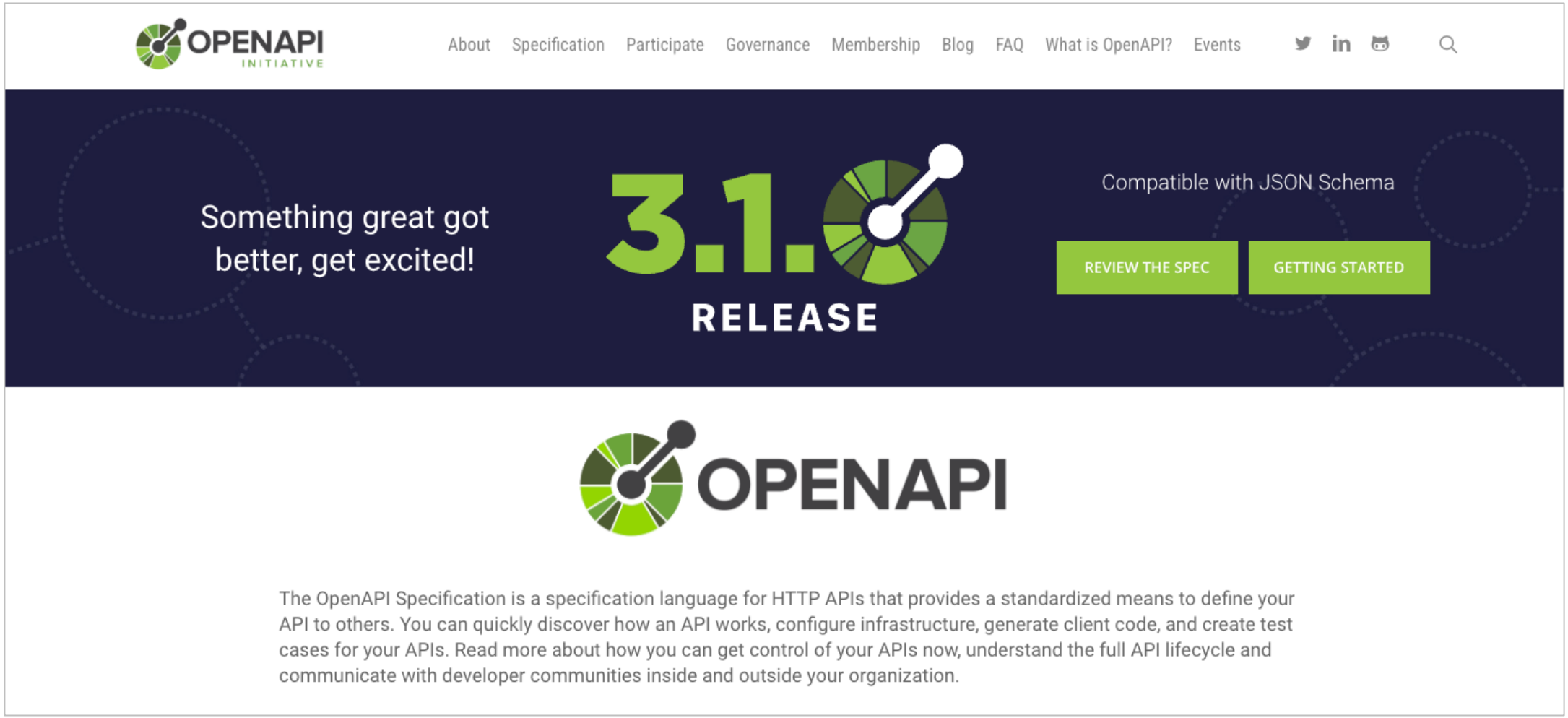
SOAP vs RESTful

- Restrictions vs Architectural Approach
- Data Exchange Format
- Service Definition
- Transport
- Ease of implementation



OpenAPI

- https://www.openapis.org/



OpenAPI

Swagger
Supported by SMARTBEAR

Why Swagger? Tools Resources

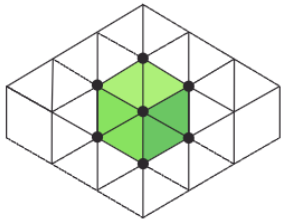
Sign In Try SwaggerHub

API Tools for Individuals, Teams, and Enterprise

SwaggerHub

The design and documentation platform for teams and individuals working with the OpenAPI Specification.

Create Free Account



Swagger Editor

API editor for designing APIs with the OpenAPI Specification

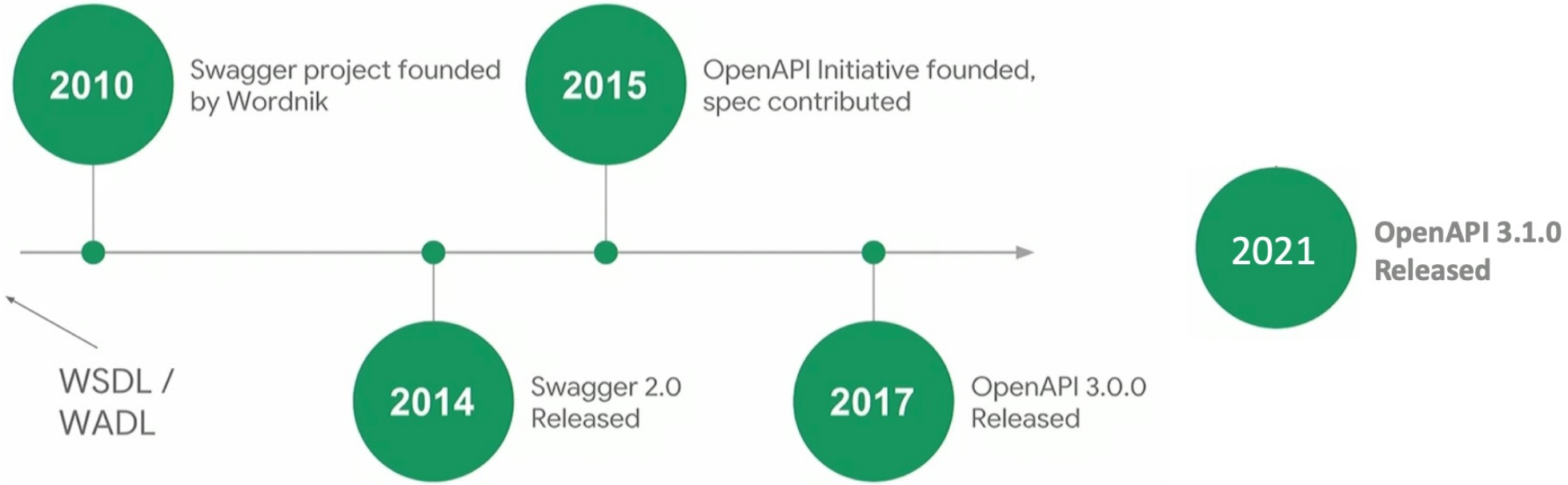
Swagger UI

Visualize OpenAPI Specification definitions in an interactive UI

Swagger Codegen

Generate server stubs and client SDKs from OpenAPI Specification definitions

OpenAPI

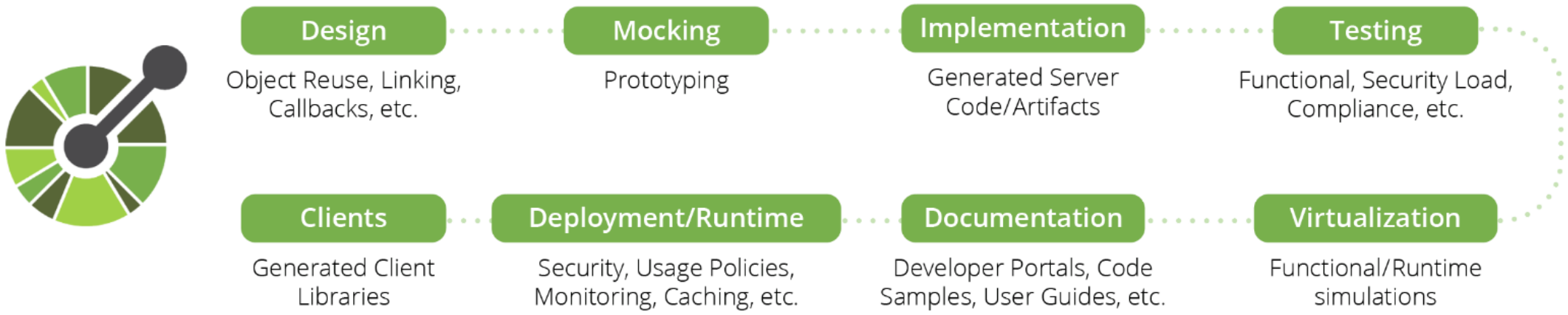


- https://spec.openapis.org/oas/v3.1.0
- https://www.openapis.org/blog/2021/02/16/migrating-from-openapi-3-0-to-3-1-0

OpenAPI – Lifecycle

1. Planning and Designing the API
2. Developing the API
3. Testing the API
4. Deploying the API
5. Retiring the API

API Lifecycle with OpenAPI



OpenAPI – Lifecycle

API Design

- OpenAPI guides API design
  - JSON → JSON Schema
  - Signatures
  - Don’t vary response models from parameters!
  - Collections and entities are a good fit
  - Versioning
  - Defaults and pagination