Spring Framework 6

Beginner to Guru

Overview of OpenAPI

What is OpenAPI?

- OpenAPI is a specification standard to describe RESTful APIs
- OpenAPI is a structured document written in JSON or YAML which adheres to the specification schema
- OpenAPI is used to:
  - Define API Endpoints
  - Authentication Methods
  - Parameters
  - Expected request and response bodies

OpenAPI History

- OpenAPI's origin starts with the Swagger Specification in 2010
- In 2015, Swagger was acquired by SmartBear Software who renamed Swagger with version 2.0 to OpenAPI and donated it to the Linux Foundation
- SmartBear formed an organization called the OpenAPI Initiative which includes member companies Apigee, IBM, Capital One, Google, Microsoft, PayPal and Intuit
- In February 2021 OpenAPI released version 3.1.0, which included alignment with JSON schema and Webhook support
- Often abbreviated as OA3 (OpenAPI 3.x)

OpenAPI Today

- Today OpenAPI is has wide adoption in the industry
- OpenAPI has become the de facto standard for defining APIs
- OpenAPI is the source for rich user documentation
- OpenAPI Generator can be used to generate client and server stubs for all popular programming languages
- OpenAPI has a very large and robust open source community

OpenAPI Generator Clients

ActionScript, Ada, Apex, Bash, C, C# (.net 2.0, 3.5 or later, .NET Standard 1.3 - 2.1, .NET Core 3.1, .NET 5.0. Libraries: RestSharp, GenericHost, HttpClient), C++ (Arduino, cpp-restsdk, Qt5, Tizen, Unreal Engine 4), Clojure, Crystal, Dart, Elixir, Elm, Eiffel, Erlang, Go, Groovy, Haskell (http-client, Servant), Java (Apache HttpClient, Jersey1.x, Jersey2.x, OkHttp, Retrofit1.x, Retrofit2.x, Feign, RestTemplate, RESTEasy, Vertx, Google API Client Library for Java, Rest-assured, Spring 5 Web Client, MicroProfile Rest Client, Helidon), k6, Kotlin, Lua, Nim, Node.js/JavaScript (ES5, ES6, AngularJS with Google Closure Compiler annotations, Flow types, Apollo GraphQL DataStore), Objective-C, OCaml, Perl, PHP, PowerShell, Python, R, Ruby, Rust (hyper, reqwest, rust-server), Scala (akka, http4s, scalaz, sttp, swagger-async-httpclient), Swift (2.x, 3.x, 4.x, 5.x), Typescript (AngularJS, Angular (2.x - 13.x), Aurelia, Axios, Fetch, Inversify, jQuery, Nestjs, Node, redux-query, Rxjs)

OpenAPI Generator Server Stubs

Ada, C# (ASP.NET Core, Azure Functions), C++ (Pistache, Restbed, Qt5 QHTTPEngine), Erlang, F# (Giraffe), Go (net/http, Gin, Echo), Haskell (Servant, Yesod), Java (MSF4J, Spring, Undertow, JAX-RS: CDI, CXF, Inflector, Jersey, RestEasy, Play Framework, PKMST, Vert.x, Apache Camel, Helidon), Kotlin (Spring Boot, Ktor, Vertx), PHP (Laravel, Lumen, Mezzio (fka Zend Expressive), Slim, Silex, Symfony), Python (FastAPI, Flask), NodeJS, Ruby (Sinatra, Rails5), Rust (rust-server), Scala (Akka, Finch, Lagom, Play, Scalatra)

Code First Development With OpenAPI

- Most programming languages have libraries to generate the OpenAPI specification from the source code
- Good for initial spec generation for existing projects
- Specification will always match what the source code has defined
- Source code can be decorated to improve completeness of generated specification

Problems with Code First Development

- Generated OA3 Specification is typically minimal
- Source code decorations to improve generated code creates clutter in source code
- Breaking API changes are hard to detect because spec always matches source code
- Lack of single source of truth for what the API should do
  - Is it the source code?
  - A Jira ticket, an email, a wiki page???

Specification First Development

- In Spec First development, the OA3 Specification is created first
- The OA3 Spec becomes the contract to define what is expected of the API
  - Single source of truth to define the API
- Enables parallel development - the producer and consumer can each develop to the spec
- The produced specification is much richer in content than a generated specification

Best Practices

- Write OpenAPI specification first
- Provide rich meta-data in specification
- Use JSON schema features to define expected request and response payloads
- Define properties that are required
- Define constraints - ie min/max size, date formats, etc
- Use OpenAPI Generator to generate client, server stubs, and data models
- Use tools to verify adherence to specification in integration tests

SPRING FRAMEWORK

GURU