



ECLIPSE
2021 CON

Automated testing of OpenAPI-described RESTful
microservices utilizing open source tools

About me

- 15+ years of experience working on a variety of software projects with clients from the USA, UK and EU, across many diverse industries
- I provide Software Engineering, Integration, Automation and IT Consulting services through my personal services company
- For more information, please see: <https://ideas.into.software/>

Objective

Automatically test your OpenAPI-described RESTful microservices with open source tools, i.e.:

- Automatically generate your REST API test scripts, along with test data, based on automatically generated OpenAPI spec
- Automatically test your REST API via a combination of such automatically generated smoke tests (<https://k6.io/docs/test-types/smoke-testing/>), load tests (<https://k6.io/docs/test-types/load-testing/>), as well as scenario tests (<https://k6.io/docs/testing-guides/api-load-testing/#user-flow-scenarios-testing>)
- Run such REST API tests continuously as part of CI/CD pipeline

Tools

- OpenAPI Generator (<https://openapi-generator.tech/>)
- Swagger / OpenAPI Specification (<https://swagger.io/specification/>)
- K6 (<https://k6.io>)
- Other (part of the technology stack of the example project): OSGi Release 7 Core and Enterprise specifications implementations, including OSGi Configurator, OSGi Declarative Services, OSGi HTTP Whiteboard and OSGi JAX-RS Whiteboard; Gecko.IO; Pac4J; EMF; Apache Felix; MongoDB; Keycloak; bnd; Gradle; Docker; Docker Compose; Java 11

Example project

- Example project demonstrating all that is showcased in this presentation is available at: <https://gitlab.com/oss-contrib/EclipseCon2021/example-project>
- This example project relies on my yet unreleased `k6` OpenAPI Generator contributions
- Please see project README for more details

Problems with existing tooling

- No ordering guarantee – but requests must be ordered for anything but most trivial tests
- No example data – often, randomly generated data cannot be used to get predictable results
- Drift – keeping REST API and REST API tests in sync
- Generated tests are only a skeleton
- Impossible or very difficult to run REST API tests as part of CI/CD pipeline
- Requires additional tech stack knowledge – lack of tech stack knowledge introduces bugs very easily

Tools not up to task

- Curl
- RESTer
- Dredd
- Swagger Test Templates / oatts
- SoapUI
- .. similar

Tools not up to task (c.d.)

None of the tools evaluated meet the requirements and have at least one or several of the following problems:

- not compatible with OAS 3 (latest OpenAPI spec)
- do not support OAuth2.0 authentication
- do not allow automatic generation of tests based on OpenAPI spec
- do not support variety of test types (smoke tests, load tests, scenario tests)
- do not provide the ability to measure SLA/SLO metrics
- not open source / have very expensive license

Solution

- `k6` OpenAPI Generator – enhancements
 - Parameter example data extraction
 - Request body example data extraction
 - Request grouping and ordering (`x-k6-openapi-operation-grouping` custom OpenAPI extension)
 - Response visibility (`x-k6-openapi-operation-response` custom OpenAPI extension)
 - Request data extraction for chaining requests (`x-k6-openapi-operation-dataextract` custom OpenAPI extension)
- `k6` OpenAPI Generator – customized templates

Solution (c.d.)

- Allows to generate REST API test scripts which are ready to run, out-of-the-box
- REST API test scripts can be generated automatically based on automatically generated OpenAPI spec
- REST API tests can be run continuously as part of CI/CD pipeline
- Allows for full round-trip development flow and ensures REST API tests are always up to date
- Since tests are generated automatically, no additional knowledge is required – developer working on REST API can focus only on REST API code

Flow: until now

1. Edit REST API code – i.e. add / update REST API methods / method parameters / examples
2. Manually update REST API test script(s) – i.e. curl calls / REStEr test definitions / similar
3. Manually update example data / payloads used for REST API calls
4. Manually / half-automatically (re) run REST API test(s) against local / development environment
5. Commit
6. Manually / half-automatically (re) run REST API test(s) against staging environment

Flow: with new toys

1. Edit REST API code – i.e. add / update REST API methods / method parameters / examples
2. Automatically (re) generate
 - (a) OpenAPI spec
 - (b) REST API test script(s)
 - (c) Example data / payloads used for REST API calls
3. Automatically (re) run REST API test(s) against local / development environment
4. Commit
5. Automatically (re) run REST API test(s) against staging environment / run as part of CI/CD pipeline

Run as part of CI/CD pipeline

- Prerequisites
 - Continuous build
 - Continuous deployment
 - For Jenkins – custom Docker Agent image with OpenAPI generator and K6 tooling

Run as part of CI/CD pipeline (c.d.)

- Then, simply add as an additional, post-continuous deployment stage, where:
 - OpenAPI spec is generated off your latest REST API code, then
 - REST API test scripts are generated automatically off that spec, then
 - REST API tests are run against latest deployed version of your REST API
- Results of test runs
 - Are visible in logs
 - Can be visualized via K6 Cloud (<https://k6.io/cloud/>)
 - Can be visualized via custom plugin / library

More information: OpenAPI Generator

- OpenAPI Generator (<https://openapi-generator.tech/>)
- [K6 Generator] Support for extracting examples defined at parameter level of Swagger/OpenAPI specification, plus minor fixes #9750 (<https://github.com/OpenAPITools/openapi-generator/pull/9750>)
- [K6 Generator] Further K6 OpenAPI generator enhancements (request body example data extraction, support for generating scenario tests and load tests out of the box, and much more) #10614 (<https://github.com/OpenAPITools/openapi-generator/pull/10614>)

More information: Swagger / OpenAPI

- Swagger / OpenAPI Specification (<https://swagger.io/specification/>)
- Adding Examples (<https://swagger.io/docs/specification/adding-examples/>)
- OpenAPI Extensions (<https://swagger.io/docs/specification/openapi-extensions/>)

More information: K6

- K6 Cloud (<https://k6.io/cloud/>)
- Smoke testing (<https://k6.io/docs/test-types/smoke-testing/>)
- Load testing (<https://k6.io/docs/test-types/load-testing/>)
- User flow (scenarios) testing (<https://k6.io/docs/testing-guides/api-load-testing/#user-flow-scenarios-testing>)
- How to Load Test OAuth secured APIs with k6? (<https://k6.io/blog/how-to-load-test-oauth-secured-apis-with-k6/>)
- Load Testing Your API with Swagger/OpenAPI and k6 (<https://k6.io/blog/load-testing-your-api-with-swagger-openapi-and-k6/>)
- Load testing with Jenkins (<https://k6.io/blog/integrating-load-testing-with-jenkins/>)

Thank you!

Join the conversation:

 [@EclipseCon](https://twitter.com/EclipseCon) | [#EclipseCon](https://twitter.com/EclipseCon)



ECLIPSE
2021 CON