OpenAPI Basics
○○○○○○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○○○

Questions?
○○

Property Based API Testing

EDD Workshop Session

Patrick Pfeifer   Johannes Brügmann

March 6, 2025

OpenAPI Basics
○○○○○○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
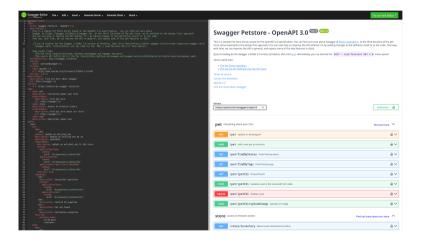○○○

Questions?
○○

Section 1

OpenAPI Basics

OpenApi: About

- API description language on top of JSON schema
- intended for REST and JSON RPC APIs with `JSON` or `XML` payloads
- language independent API specification written in `JSON` or `YML`
- plenty of generators and tools for individual language bindings
- client and server adapter generation for models and api

OpenApi: Building Blocks

- General info block (title, license, info, server, tags, …)
- Operations (HTTP-verb + Path, request and responses, media-types)
- Schemata (named and anonymous models, parameters)
- Security (OAuth 2.0, HTTP BasicAuth, …)

OpenAPI Basics
●●●○●○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○○○

Questions?
○○

OpenApi: Swagger Editor

OpenAPI Basics
○○○○●○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○○○

Questions?
○○

Vorbereitung

ERNICommunity GitHub Projekt auschecken:
https://github.com/ERNICommunity/openapi-workshop

A1: Modelliere in OpenAPI

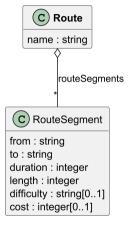**Figure 1:** Klassendiagramm für das Modell "Route".

OpenAPI Basics
○○○○○○○●○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○○○

Questions?
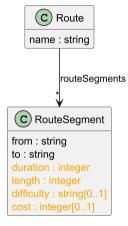○○

A1: Ausgangslage

Figure 2: Fehlende "Route".

A1: 4 "Route"-Attribute hinzufügen

- duration: integer
- length: integer
- difficulty: string[0..1]
- cost: [0..1]

OpenAPI Basics
〇〇〇〇〇〇〇〇●

Property-Based-Testing Basics
〇〇〇〇

Applying PBT to OpenAPI /-Generator
〇〇〇

Questions?
〇〇

Section 2

Property-Based-Testing Basics

PBT: General

Figure 3: Testing Dilemma: "testing can only find bugs, not prove their absence" (E.W. Dijkstra in [1]).

PBT: Basics

$$\forall i \in I : P(i) = \top, \qquad P : I \to \{\top, \bot\} \tag{1}$$

- I: domain of valid input values ("Eingabedomäne")
- P: property to prove; candidates include:
  - invariants
  - round-trips
  - symmetry
  - commutativity
  - idempotency
  - induction

[2]

A2: Implementiere PBT `isValid()`

- using the Arbitraries class, write a `jqwik` "**?**" function to define an "Arbitrary" that generates input data ("Testdatengenerator")
- validate all inputs with `openapi4j`
- use model `Route` from previous exercise

OpenAPI Basics
○○○○○○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
●○○

Questions?
○○

Section 3

Applying PBT to OpenAPI /-Generator

OpenAPI Basics
○○○○○○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○●○

Questions?
○○

Folie 3

A3: Modelliere in OpenApi und prüfe mit PBT

Figure 4: Klassendiagramm für das polymorphe Modell "RouteSegment".

OpenAPI Basics
○○○○○○○○○

Property-Based-Testing Basics
○○○○

Applying PBT to OpenAPI /-Generator
○○○

Questions?
●○

Section 4

Questions?

References

[1]   P. Godefroid and K. Sen, "Combining model checking and testing," *Handbook of Model Checking*, pp. 613–649, 2018, Available: https://patricegodefroid.github.io/public_psfiles/mchandbook2018.pdf

[2]   S. Sahayam, "Property-based testing patterns." Jun. 2016. Available: https://blog.ssanj.net/posts/2016-06-26-property-based-testing-patterns.html