ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Spring Workshop

3 - Web

Part 1: Basics

Agenda

- Common Annotations & Classes
- DTOs & Records & Mapping
- BeanValidation
- Exception Handling
- WebTests

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Common Annotations (already in use)

- @RestController
- @RequestMapping
  - URL
  - Shortcuts for different Http Methods
- @PathVariable
  - Used for Url Matching

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Common Annotations (new)

- @RequestBody
  - Maps the complete body into an object
- @RequestPart
  - Multipart Form
  - e.g. file upload
  - also support type conversion
- @RequestParam
  - Query Strings
  - Form Data

Common Annotations Classes (new)

- ResponseEntity
  - Allows to set headers and status
  - e.g. Content-Disposition for download
  - superior @ResponseBody (included in @RestController)
- HttpEntity
  - Like ResponseEntity but for @RequestBody

BeanValidation

- Standard with built-in validators
- https://jakarta.ee/specifications/bean-validation/3.0/apidocs/
- Examples
  - @NotNull
  - @NotBlank
  - @NotEmpty
  - @Min
  - @Past
  - @Pattern
- @Valid: Recursive Application

BeanValidation Example 1/2

```
public class Customer {

  private Long id;
  private String firstname;

  @NotBlank
  private String lastname;

  private Boolean signedGdpr;
}
```

BeanValidation Example 2/2

```
public class CustomersController {

  @PostMapping

  public void add(@RequestBody @Valid Customer customer) {

    this.repository.add(customer);

  }

}
```

DTOs

- Data Transfer Object
- Heavily used between
  - Modul Boundaries
  - Endpoints
- Can be constructed via record
- Support BeanValidation

DTOs

```
public record AddCustomer(
  @NotBlank String firstname,
  @NotBlank String lastname
) {}

public class CustomersRepository {
  public Instant add(AddCustomer addCustomer) {
    var customer = new Customer(
      this.id++,
      addCustomer.firstname(),
      addCustomer.lastname(),
      showGdpr
    );
    customers.add(customer);
    return LocalDate.now().atStartOfDay().toInstant(ZoneOffset.UTC);
  }
}
```

DTOs

Validate on Controllers

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Exception Handling

- ResponseStatusException
  - Individually
  - Maps an Exception to an HttpStatus
  - Is Applied per method (endpoint)
  - Spring internally does the mapping
- @ControllerAdvice
  - Globally
  - Returns ReponseEntity
- Problem Details for HTTP Apis
  - https://www.rfc-editor.org/rfc/rfc7807.html

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

ControllerAdvice

```
@ControllerAdvice

public class GlobalControllerAdvice extends ResponseEntityExceptionHandler {

  @ExceptionHandler(RuntimeException.class)

  public ProblemDetail handleIdNotFound() {

    return ProblemDetail.forStatusAndDetail(HttpStatus.BAD_REQUEST, "something went wrong");

  }

}
```

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

WebTests with WebTestClient

- @AutoConfigureMockMvc starts server
- Allows us to send HTTP Request to Controllers
- WebTestClient
  - more powerful than MockMvc
  - Fluent
  - JSONPath for assertions: https://jsonpath.com/

Example WebTest

```
@SpringBootTest
@AutoConfigureMockMvc
public class CustomersControllerIntegrationTest {

  @Autowired
  WebTestClient webTestClient;

  @Test
  void testRepo() {
    webTestClient.get().uri("/api/customers").exchange().expectStatus().isOk();
  }
}
```

Example WebTest

```
public class CustomersControllerIntegrationTest {
  @Test
  void testAddCustomer() {
    AddCustomer addCustomer = new AddCustomer("Franz", "Maier");
    webTestClient
      .post()
      .uri("/api/customers")
      .bodyValue(addCustomer)
      .exchange()
      .expectStatus()
      .isOk();

    webTestClient
      .get()
      .uri("/api/customers")
      .exchange()
      .expectBody()
      .jsonPath("[0].firstname")
      .isEqualTo("Franz");
  }
}
```

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Example WebTest

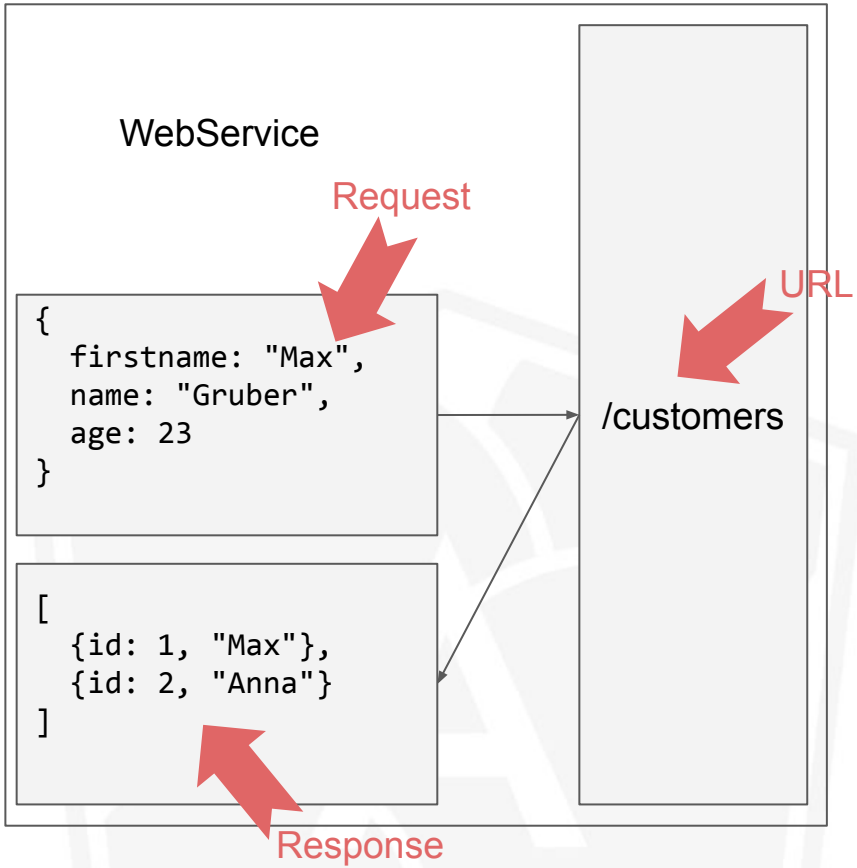
```
@SpringBootTest
@AutoConfigureMockMvc
public class CustomersControllerIntegrationTest {
  @Test
  void testAddInvalidCustomer() {
    AddCustomer addCustomer = new AddCustomer("Franz", "");
    webTestClient
      .post()
      .uri("/api/customers")
      .bodyValue(addCustomer)
      .exchange()
      .expectStatus()
      .isBadRequest();
  }
}
```

Part 2: OpenAPI

Agenda

- Swagger / OpenAPI
- UI
- Code Generator
- ShowCase on Customers
- API First

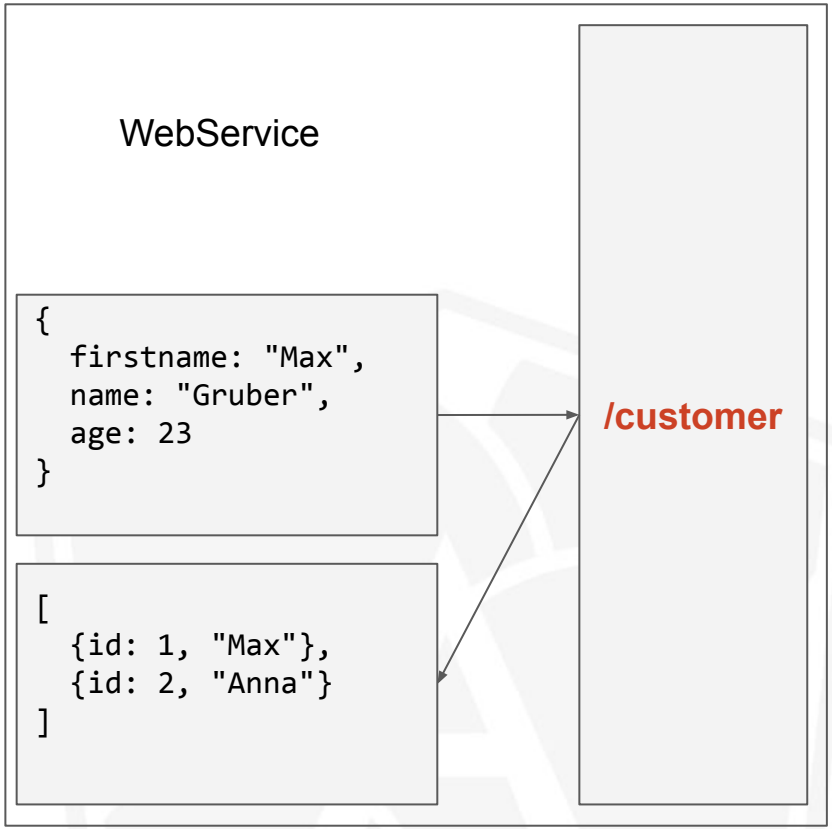
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE



Client / Javascript

```
function addCustomer(firstname, name, age) {
  return fetch(
    '/customers',
    {firstname, name, age}
  );
}
```

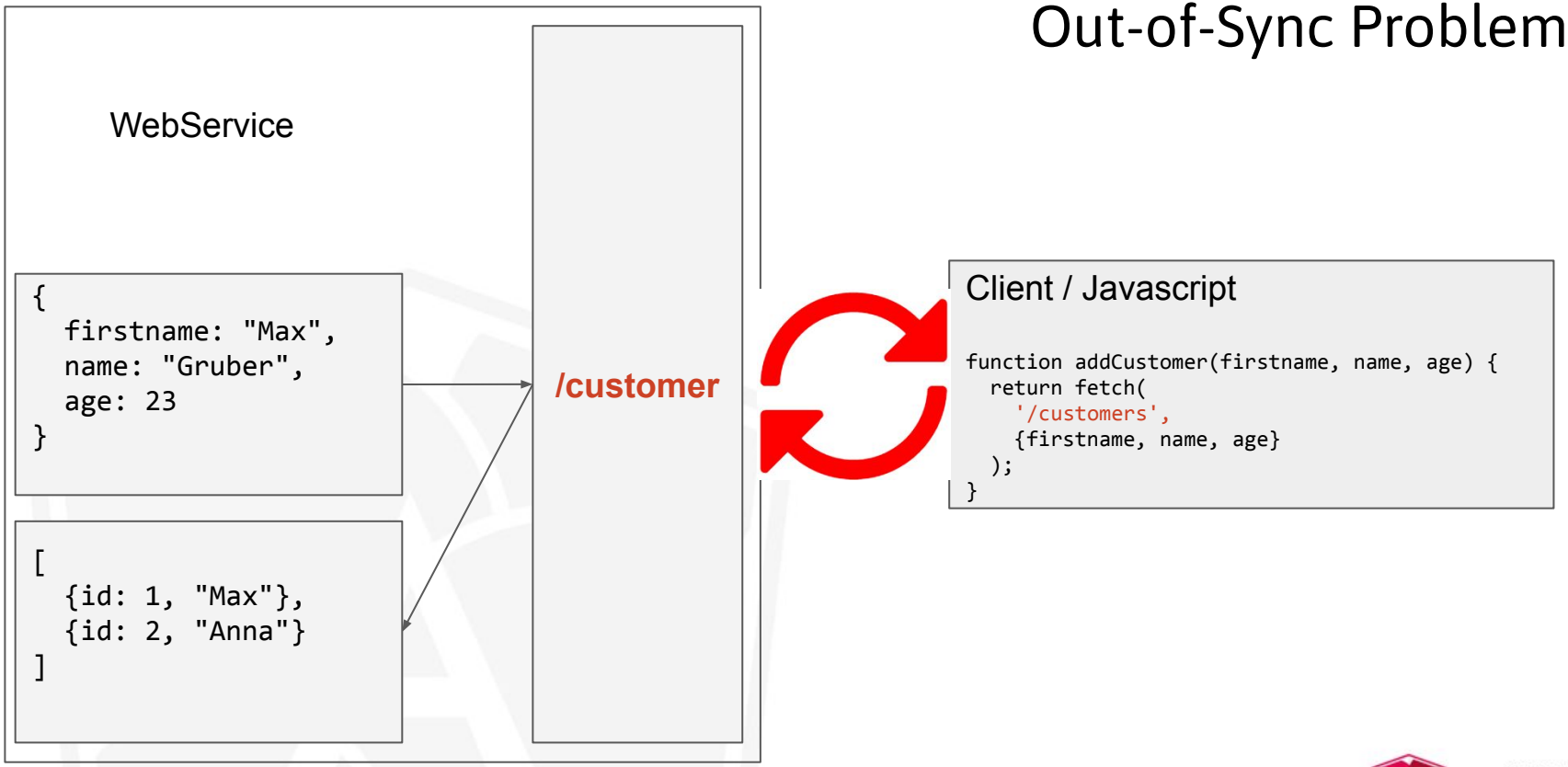
Out-of-Sync Problems



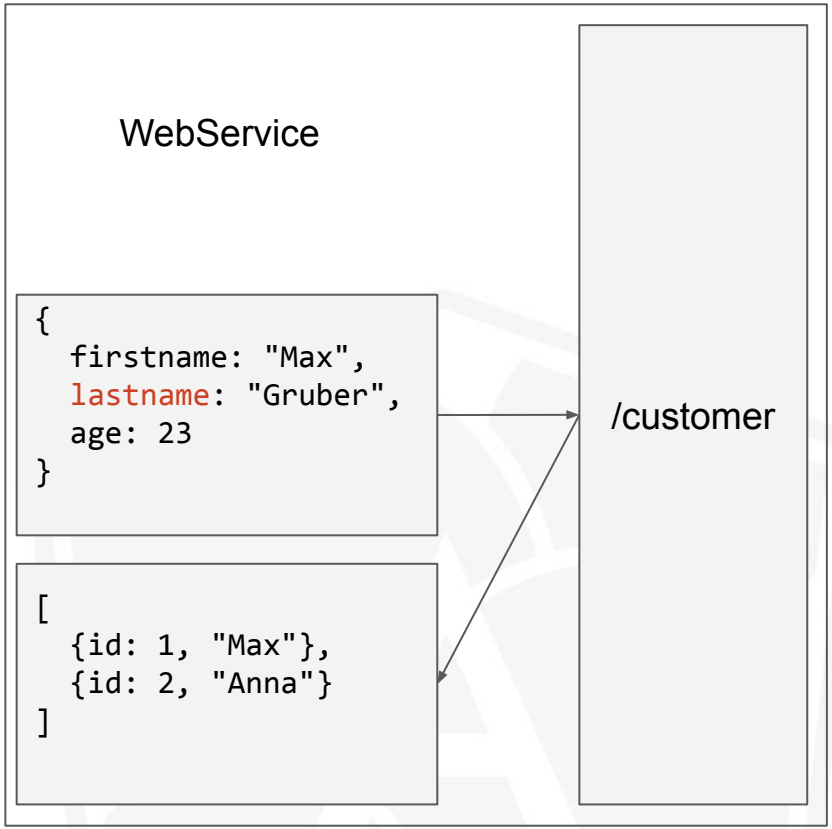
Client / Javascript

```
function addCustomer(firstname, name, age) {
  return fetch(
    '/customers',
    {firstname, name, age}
  );
}
```

Out-of-Sync Problems



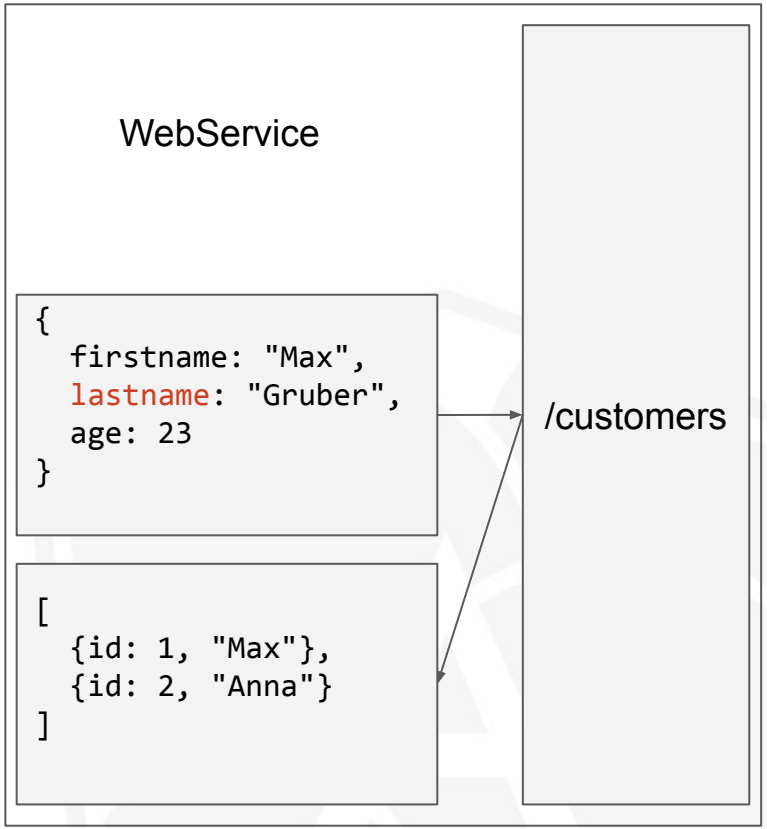
Out-of-Sync Problems

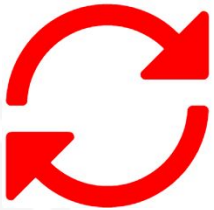


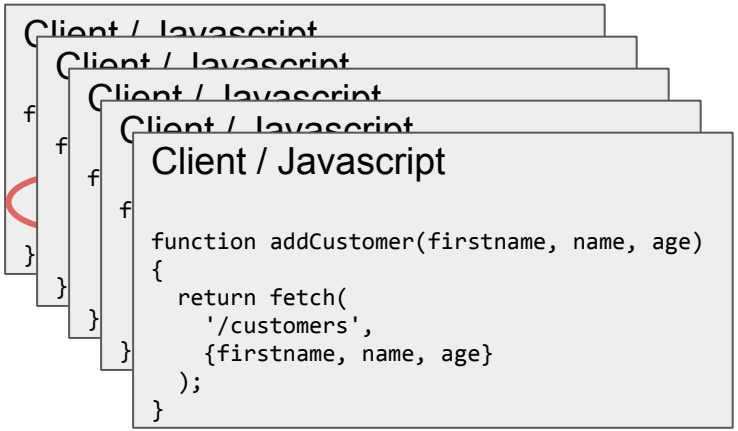
Client / Javascript

```
function addCustomer(firstname, name, age) {
  return fetch(
    '/customers',
    {firstname, name, age}
  );
}
```

Out-of-Sync Problems
(Microservices Version)







Ruby Client | Python Client | C# Client

PHP Client | Java Client | Objective-C Client

OpenAPI

Solution for developing, documenting and maintaining Web APIs

- Elements
  - URL Paths
  - Response Types
  - Request Types
- Code duplication
- On-Going synchronisation efforts
- Resulting bugs

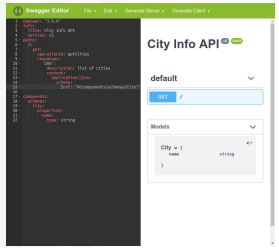
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

The Ecosystem

Documentation

Specification
v2 = Swagger Spec.
v3 = OpenAPI

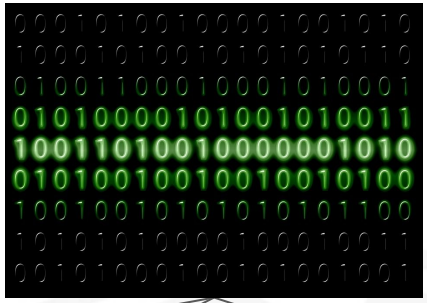
Tools



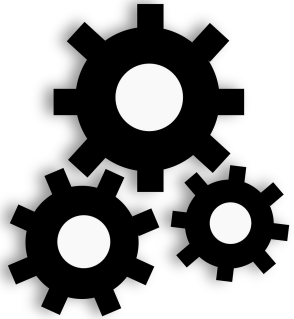
Editor

Code First

API First
(Code Generators)



Tools



spring

Microsoft® .NET

others

SpringDoc

Swashbuckle

Client

and many more

Server Stubs

spring

ASP.NET Core

Flask
web development, one drop at a time

and many more

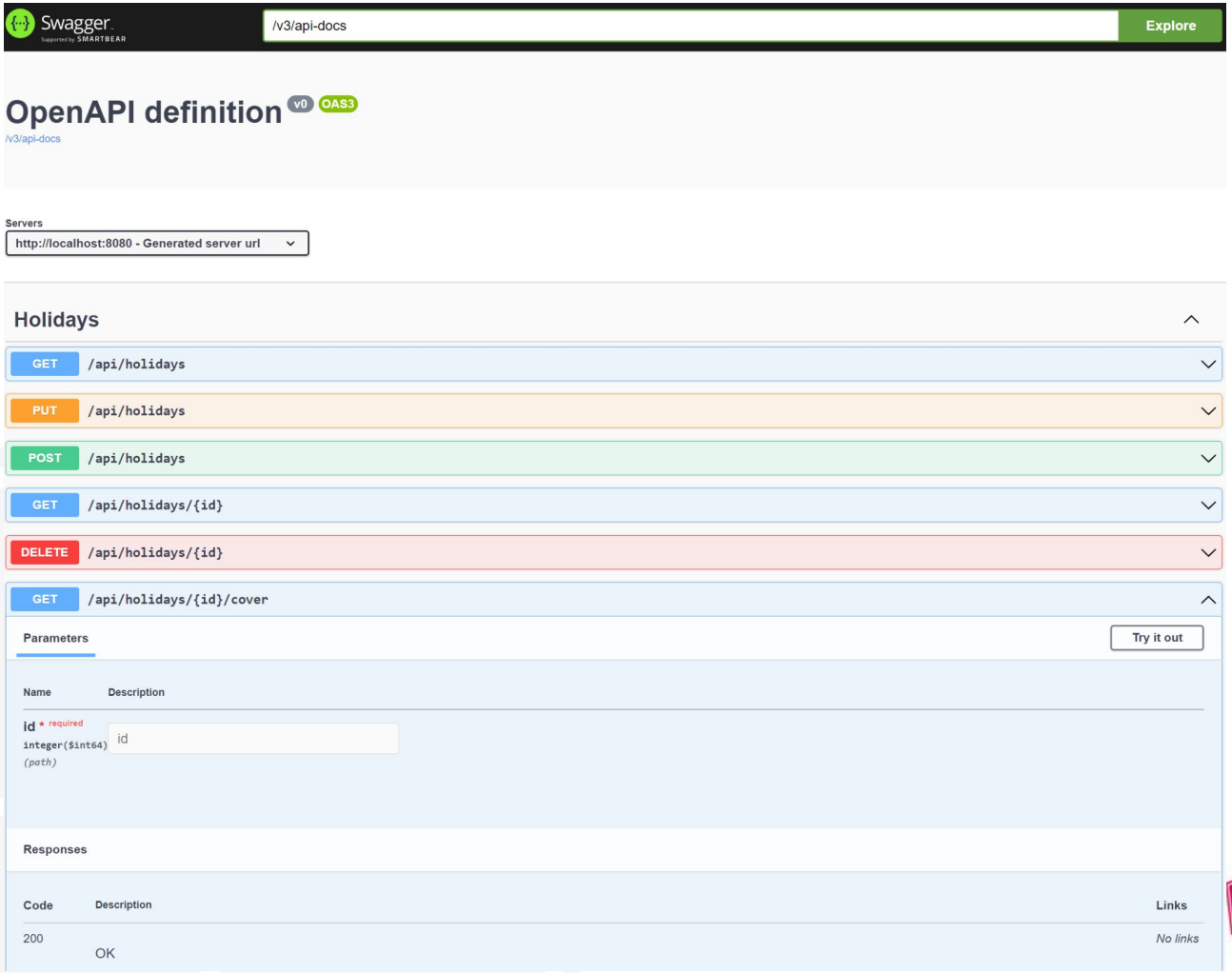
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Dependencies

- SpringDoc
  - org.springdoc:springdoc-openapi-starter-webmvc-ui:2.0.0-RC1
  - Successor of SpringFox
  - Lots of Customisations
- openapi-generator-cli
  - NPM Package or Gradle Plugin
  - Runs Java internally

Important Annotations

- @Tag
  - Defines Name of generated Angular Service
- @Operation
  - Defines Name of Method



API First

- Specification in YML
- Necessity to generate Server Stubs (next to clients)
- Generation via Script or Gradle Task

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

OpenAPI Spec - Part 1: Header

```yaml
openapi: 3.0.1
info:
  title: OpenAPI definition
  version: v0
servers:
  - url: http://localhost:8080
    description: Generated server url
```

OpenAPI Spec - Part 2: Paths

```yaml
paths:
  /api/booking:
    get:
      tags:
        - Customers
      operationId: findAll
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Customer
```

OpenAPI Spec - Part 3: Schemas

```yaml
components:
  schemas:
    Customer:
      required:
        - id
      type: object
      properties:
        id:
          type: integer
          format: int64
        birthday:
          type: string
          format: date
```

Gradle Plugin

```
plugins {
  id 'org.springframework.boot' version '3.1.1'
  id 'io.spring.dependency-management' version '1.1.0'
  id 'java'
  id 'org.openapi.generator' version '6.6.0'
}
```

Gradle Configuration

```
openApiGenerate {
  generatorName.set('spring')

  inputSpec.set("$rootDir/api/customers.yml")
  outputDir.set("$rootDir/generated")

  additionalProperties.set([
    interfaceOnly: true,
    apiPackage: 'com.softarc.eternal.web.api',
    modelPackage: 'com.softarc.eternal.web.model',
    useSpringBoot3: true
  ])
}
```

API First - Further Reading

- OpenAPI Specification
  - https://swagger.io/specification/
- Spring Generator
  - https://openapi-generator.tech/docs/generators/spring/
- Gradle Plugin
  - https://github.com/OpenAPITools/openapi-generator/tree/master/modules/openapi-generator-gradle-plugin

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Part 3: Upload & Download

Controller for File Uploads

```
@RequestMapping("/api/customers")
@RestController
@Tag(name = "Customers")
public class CustomersController {

  @PostMapping(consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
  @Operation(operationId = "add")
  public boolean add(
    @RequestPart AddCustomer addCustomer,
    @RequestPart MultipartFile profile
  ) throws IOException {

    var filename = profile.getOriginalFilename();
    var path = Path.of("", "filestore", filename);
    this.repository.add(addCustomer, filename);
    profile.transferTo(Path.of("", "filestore", filename));
    return true;
  }
}
```

Angular Code for Upload

```
/* Legacy Code without OpenAPI */
const { name, description } = holiday;
if (holiday.cover) {
  formData.append('cover', holiday.cover);
}

const blob = new Blob([JSON.stringify({ name, description })], {
  type: 'application/json',
});

formData.append('holidayDto', blob);
assertDefined(holiday.cover);

const saveRequest: SaveRequest = {
  cover: holiday.cover,
  holidayDto: blob as HolidayDto,
};
this.#httpClient.post('http://localhost:8080/api/holidays', formData);
```

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE