

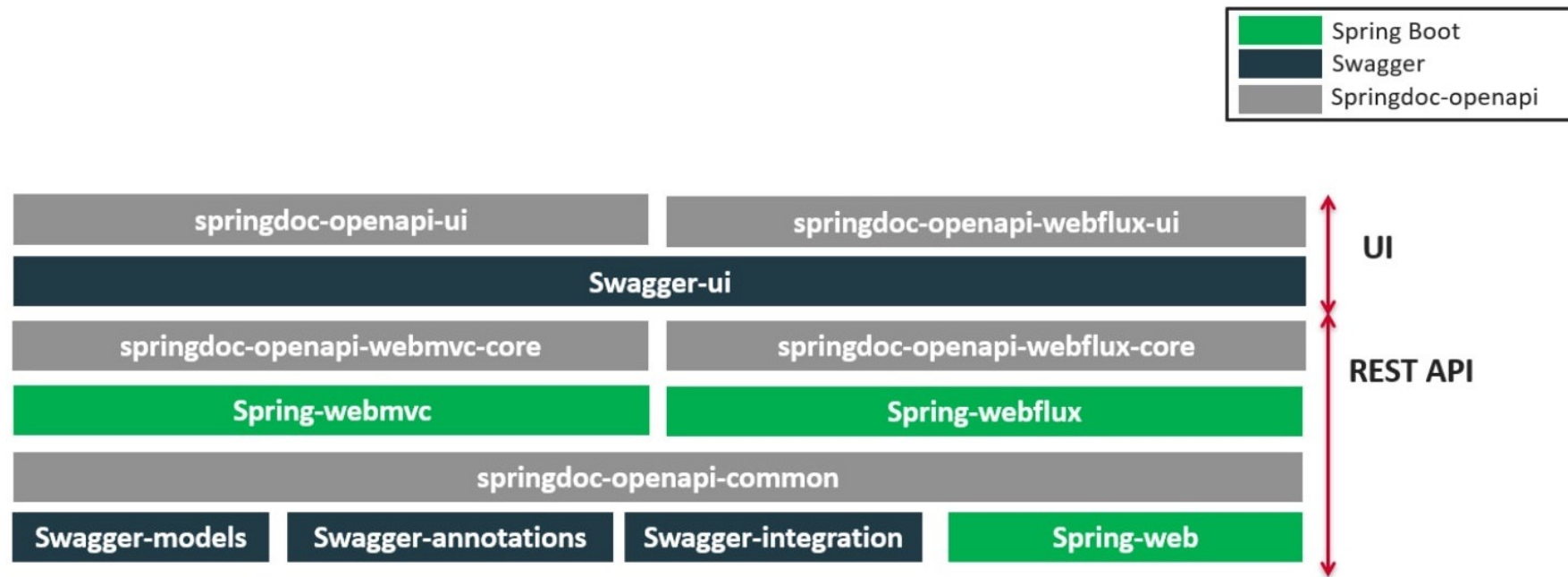
OpenAPI 3.0

INTERNET SOFTVERSKJE ARHITEKTURE, FAKULTET TEHNIČNIH NAUKA,
NOVI SAD

Šta je OpenAPI?

- ▶ OpenAPI specifikacija je definiše standardni interfejs za REST API-je, nezavisno od programskog jezika
- ▶ OpenAPI fajl omogućava opis:
 - ▶ Dostupnih endpointa i operacija za svaki endpoint (npr. GET /users)
 - ▶ Ulaznih i izlaznih parametara za svaku operaciju
 - ▶ Metoda autentifikacije
 - ▶ Licence, uslova korišćenja endpoint-a, informacija o vlasniku i druge informacije
- ▶ Specifikacija može biti napisana u json ili yaml formatu

OpenAPI i SpringBoot komponente



Integracija sa SpringBoot-om

- ▶ Dodati u pom.xml
- ▶

```
<dependency>  
  <groupId>org.springdoc</groupId>  
  <artifactId>  
    springdoc-openapi-ui  
  </artifactId>  
  <version>1.6.4</version>  
</dependency>
```


openapi.json

- ▶ Nakon pokretanja aplikacije, openapi.json document je potom dostupan na stranici: <http://localhost:8080/v3/api-docs/> a grafički prikaz endpoint-a je dostupan na adresi: <http://localhost:8080/swagger-ui/index.html#/>
- ▶ Ovako izgenerisan fajl će sadržati podrazumevani opis svih API-ja: tip metode, putanju, parametre putanje i podrazumevani 200 OK odgovor.
- ▶ Dodatne informacije o vlasniku, uslovima korišćenja, licenci, serverima i sl. je moguće podesiti kroz @Configuration klasu, pomoću OpenAPI() instance.

Rest example

v1.0

OAS3

[/v3/api-docs](#)

Rest example with OpenAPI documentation

[Terms of service](#)

[FTN - Website](#)

[Send email to FTN](#)

[License](#)

Servers

[http://localhost:8080/ - development server](#)

Greeting controller

The greeting API



GET **/api/greetings/{id}** Get greeting by id



PUT **/api/greetings/{id}** Update an existing greeting



DELETE **/api/greetings/{id}** Deletes a greeting



PATCH **/api/greetings/{id}** Update fields of an existing greeting



GET **/api/greetings** Get all greetings



POST **/api/greetings** Create new greeting



GET **/api/greetings/search** Searches greetings by containing text

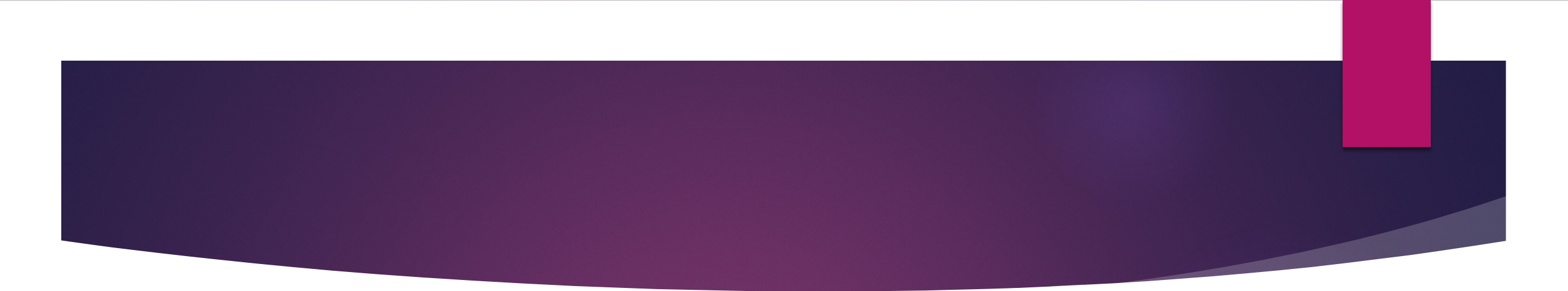


Objekti putanje

- Moraju počinjati sa /
- Dozvoljeno je pravljenje šablona za URL-ove, ali prednost prilikom mapiranja imaju konkretni URL-ovi
- Nije dozvoljena ambivalencija: šabloni iste hijerarhije ali drugog naziva

Formiranje putanje

- ▶ PathItem objekti – formiraju se na osnovu anotacija metoda u kontroleru
- ▶ @Operation – služi za definisanje operacija u kontrolerima (po defaultu će openapi skenirati sve metode koje su anotirane sa @Path i odgovarajućom HTTP metodom)
- ▶ @ApiResponse – može da se koristi na nivou metode ili kao deo @Operation anotacije, kako bi se definisali HTTP odgovori

- 
- ▶ @Content definiše tip sadržaja, ako se ne vraća ništa, content je prazan
 - ▶ @ArraySchema definiše tip niza koji se koristi za sadržaj odgovora
 - ▶ @Schema definiše tip odgovora
 - ▶ @Parameter se koristi za anotaciju parametara metode

```
► @Operation(summary = "Update fields of an existing greeting", description = "Update fields of an existing greeting")
  @ApiResponses(value = {
      @ApiResponse(responseCode = "200", description = "Greeting successfully edited",
          content = {
              @Content(mediaType = "application/json", schema = @Schema(implementation = Greeting.class))
          }
      ),
      @ApiResponse(responseCode = "404", description = "Greeting not found", content = @Content)})
  @PatchMapping(value =("/{id}", consumes = MediaType.APPLICATION_JSON_VALUE, produces = MediaType.APPLICATION_JSON_VALUE)
  public ResponseEntity<Greeting> updateGreetingText(@RequestBody GreetingTextDTO greetingDTO,
      @Parameter(description = "Greeting id to update", required = true) @PathVariable long id)
  {
      Greeting updatedGreeting = null;
      try {
          updatedGreeting = greetingService.updateGreetingText(greetingDTO, id);
      } catch (Exception e) {
          e.printStackTrace();
          return new ResponseEntity<Greeting>(HttpStatus.NOT_FOUND);
      }

      return new ResponseEntity<Greeting>(updatedGreeting, HttpStatus.OK);
  }
```