Programiranje internet aplikacija

Elektrotehnički fakultet, Univerzitet u Beogradu

2024/2025

Spring

- Open-source framework za kreiranje modernih Java EE aplikacija
- Komplementarnost sa Jakarta EE

- Modularnost
- Glavne prednosti
  - Fleksiblinost
  - Dobra podrška
  - Velika zajednica
  - Brzina
  - Jednostavnost

Jedan od najpoznatijih Java framework-a

Spring projekti

- Za različite infrastrukture i potrebe aplikacija koje želimo
- Najpoznatiji:
  -  Spring Boot
  -  Spring Data
  -  Spring Cloud
  -  Spring Security
  -  Spring Batch
  -  Spring Session

Spring Boot

- Minimalna konfiguracija za brzo kreiranje i pokretanje Spring aplikacija
- 3.3.4. -> Java 21 (na fakultetu: 21.0.4)
- Ugrađeni build alati:
  - Maven
  - Gradle
- Ugrađeni servlet container-i:
  - Tomcat
  - Jetty
  - Undertow

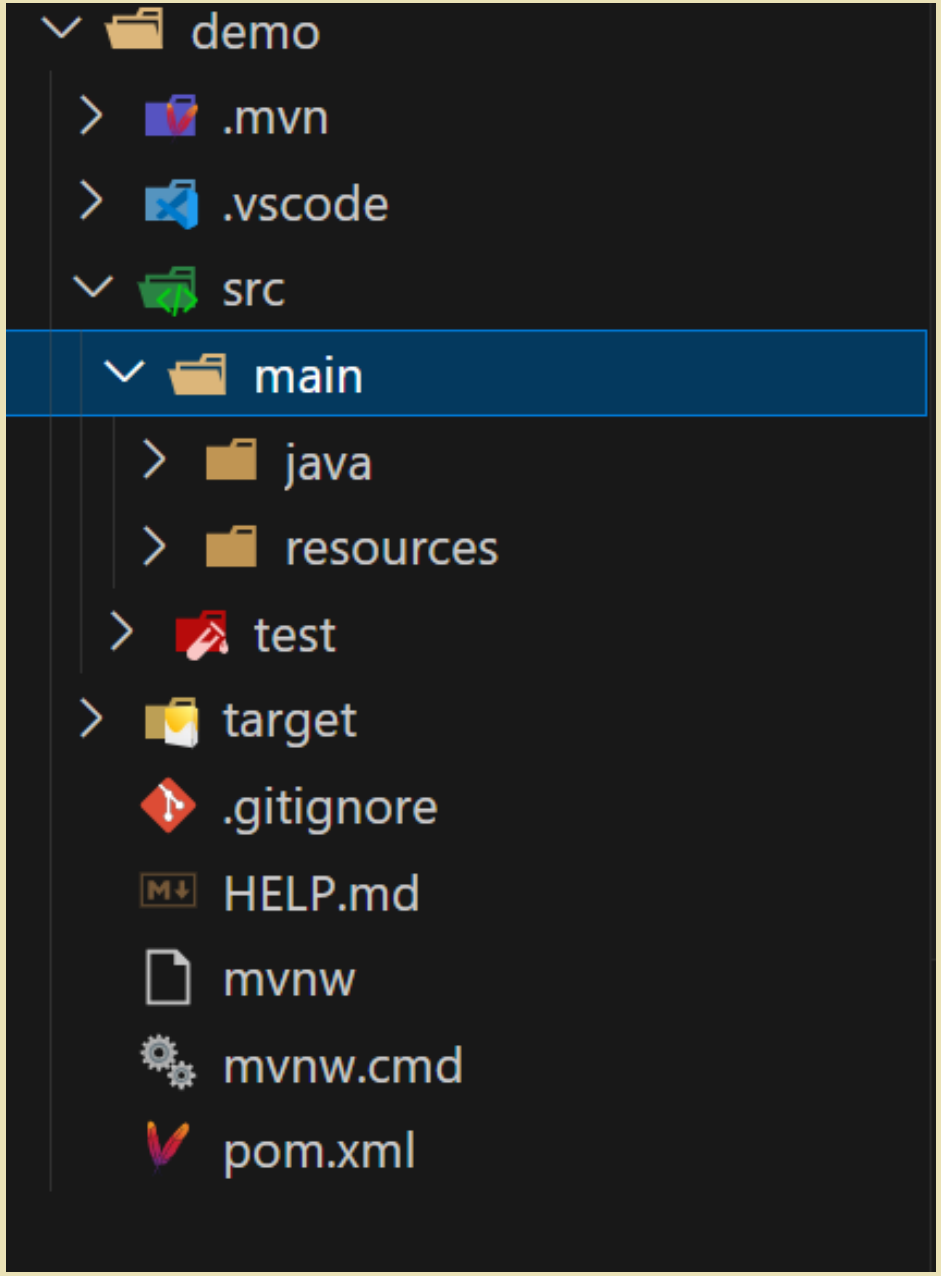
Inicijalizacija

- **Spring Boot Extension Pack** - https://code.visualstudio.com/docs/java/java-spring-boot

- CTRL+SHIFT+P(The Command Palette): Spring Initializr: Create a Maven Project
- Verzija: 3.3.4. ; Jezik: Java
- GroupId i artifactId
- Packaging type i java verzija

- Zavisnosti: Spring Web(Spring Boot Starter Web) – za kreiranje web aplikacija, uključujući RESTful, koristeći Spring MVC, i Tomcat kao servlet container

Struktura projekta



```java
@SpringBootApplication
public class DemoApplication {

    Run | Debug
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

- **@SpringBootApplication**
- Meta-anotacija koja kombinuje:
  - **@SpringBootConfiguration**
  - **@EnableAutoConfiguration**
  - **@ComponentScan**

- **Main metoda**
- Ulazna tacka
- Započinje bootstraping, startuje Spring i pokreće Tomcat
- Prosleđujemo glavnu Spring komponentu i ostale argumente

Kontroleri i mapiranje zahteva

- @RestController = @Controller + @ResponseBody

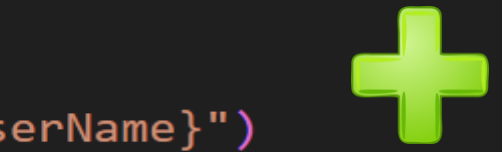
- @RequestMapping
  - @GetMapping
  - @PostMapping
  - @PutMapping
  - @DeleteMapping
  - @PatchMapping


```java
@RestController
public class UserController {
    @RequestMapping("/users")
    public String helloUsers() {
        return "Hello users";
    }
}
```

```java
@RestController
@RequestMapping("/users")
public class UserController {

    @GetMapping()
    public List<User> getAllUsers() {
        ArrayList<User> allUsers = new ArrayList<>();
        allUsers.add(new User(firstName:"Petar", lastName:"Petrovic"));
        allUsers.add(new User(firstName:"Mirko", lastName:"Mirkovic"));
        allUsers.add(new User(firstName:"Jova", lastName:"Jovic"));
        return allUsers;
    }

    @GetMapping("/{userName}")
    public User getUser(@PathVariable String userName) {
        if(userName.equals(anObject:"Petar"))
            return new User(firstName:"Petar", lastName:"Petrovic");
        else
            return null;
    }
}
```



@RequestParam

@CrossOrigin(origins = ”http://localhost:4200/”)

```java
@PostMapping("/login")
public User login(@RequestBody User u){
    for(int i =0; i<this.allUsers.size(); i++){
        if(this.allUsers.get(i).getUserName().equals(u.getUserName())
        && this.allUsers.get(i).getPassword().equals(u.getPassword())){
            return this.allUsers.get(i);
        }
    }
    return null;
}
```

Rad sa bazom podataka

- Wamp 3.3.5 -> MySQL server 8.4.2
- MySQL Workbench 8.0.38 CE

```xml
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
    <scope>runtime</scope>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
```

Konfiguracija baze podataka

```java
@Configuration
public class DB {

    @Bean
    public static DataSource source()
    {

        DriverManagerDataSource dataSource = new DriverManagerDataSource();
        dataSource.setDriverClassName(driverClassName:"com.mysql.cj.jdbc.Driver");
        dataSource.setUrl(url:"jdbc:mysql://localhost:3306/baza");
        dataSource.setUsername(username:"root");
        dataSource.setPassword(password:"");

        return dataSource;
    }
}
```

Izvršavanje upita

- Uzimanje konekcije iz izvora podataka

```
Connection conn = DB.source().getConnection();
```

- Upit

```
PreparedStatement stm = conn.prepareStatement("...");
stm.set …
```

- Izvršavanje
  - Select -> **`stm.executeQuery()`**
  - Insert/Update/Delete -> **`stm.executeUpdate()`**
- Oslobađanje resursa

HVALA NA PAŽNJI!