



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH **ústav**  
TECHNOLOGIÍ **telekomunikací**

# Multimediální systémy

## Cvičení č. 3

Garant kurzu: doc. Ing. Petr Číka PhD.

Cvičící: Ing. David Kohout

Ing. Milan Bubniak

Akademický rok: 2022/2023

# Osnova

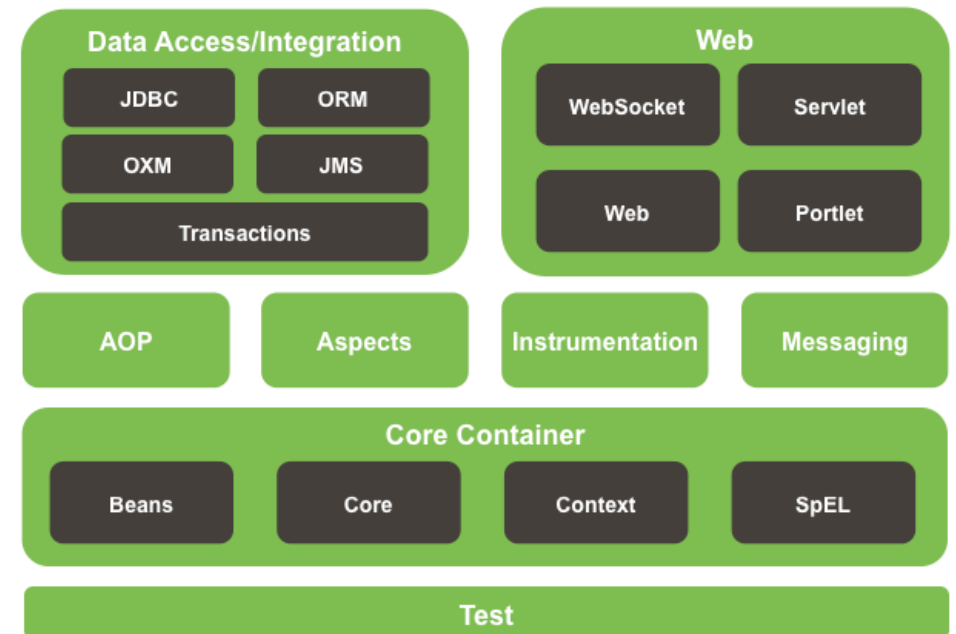
- Základy frameworku Spring
- Anotace ve Springu
- Spring Boot
- Samostatná práce na základy Springu

# Framework Spring

- Pro vývoj J2EE (Jakarta EE [Enterprise Edition]) aplikací.
- Usnadnění vývoje enterprise aplikací
- Skládá se z asi **dvaceti** modulů.




Spring Framework Runtime



- Hojně využívaný v praxi (př. Netflix)
- Licence: OpenSource – [Apache 2.0 license](#)
- GitHub: <https://github.com/spring-projects>

# Spring Boot

- Spring Boot umožňuje jednoduše vytvářet samostatné a produkční aplikace založené na Spring Frameworku, které stačí jen spustit.
- Lze jednoduše vytvořit přes  **spring®** **initializr**
- Lze vytvořit i přímo v IntelliJ IDEA Ultimate (studentské licence VUT zdarma)
- <https://start.spring.io/>

**Project**☒ Maven Project☐ Gradle Project**Language**☒ Java ☐ Kotlin☐ Groovy**Spring Boot**☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (M5) ☐ 2.7.5 (SNAPSHOT)☒ 2.7.4 ☐ 2.6.13 (SNAPSHOT) ☐ 2.6.12**Project Metadata**

Group mds

Artifact uvod

Name uvod

Description Uvod do springu

Package name mds.uvod

Packaging ☒ Jar ☐ WarJava ☐ 19 ☐ 17 ☒ 11 ☐ 8**Dependencies****ADD DEPENDENCIES...** CTRL + B**Spring Web** WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**Thymeleaf** TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

**GENERATE** CTRL + G**EXPLORE** CTRL + SPACE**SHARE...**

# Anotace

- Anotace jsou ve Springu velice důležitou součástí
  - Umožňují jednoduché a rychlé nastavování metadat
  - Zjednodušují kód
- 
- Metadata Springu je možné nastavovat také pomocí XML, případně přímo v kódu. Dnes se obvykle využívá kombinace anotací a v kódu.

# Anotace – @SpringBootApplication

- Základní anotace, která využívá automatickou konfiguraci projektu
- Ideální mít main v root složce projektu (např. src/main/java/mds/uvod)
- Ekvivalentní jako využití následujících anotací:
  - @EnableAutoConfiguration – načítá auto konfiguraci Spring Bootu
  - @ComponentScan – automatický sken @Component ve všech balíčcích
  - @Configuration – registrace extra kontextů, či načtení dodatečných konfigurací

```
@SpringBootApplication
public class SpringApplication {
    ...
}
```

```
@EnableAutoConfiguration
@ComponentScan
@Configuration
public class SpringApplication {
    ...
}
```

# Anotace – @Controller

- Třída kterou takto označíme je automaticky proskenována přes všechny metody a ostatní anotace
- Anotace označuje třídu jako Controller

```
@Controller  
public class CustomController {  
    ...  
}
```



# Anotace – @RestController

- Speciální verze Controlleru
- Funguje jako kombinace @Controller a @ResponseBody
- Není tak potřeba používat @ResponseBody u každé metody
- Ve výsledku tak zjednodušuje celou třídu

```
@RestController
public class CustomController {

    @GetMapping(„hello“)
    public String hello() {
        return „Hello World!“;
    }
}
```

```
@Controller
public class CustomController {

    @GetMapping(„hello“)
    @ResponseBody
    public String hello() {
        return „Hello World!“;
    }
}
```

# Anotace – @ResponseBody, @RequestBody

- @RequestBody – mapuje tělo http requestu pro přenos a umožňuje automatickou deserializaci z http na Java objekt
- @ResponseBody – říká controlleru, že navracený objekt je automaticky serializován do JSONu. Případně, že má Controller výsledek přímo zobrazit.

```
@GetMapping(„hello“)
@ResponseBody
public String hello() {
    return „Hello World!“;
}
```

# Anotace – @GetMapping

- Mapuje http Get requesty na určité obslužné metody

```
@GetMapping(„hello“)  
public String hello() {  
    return „Hello World!“;  
}
```

- Dotazem na <http://localhost:8080/hello> získáme:  
Hello World!

# Anotace – @RequestMapping

- Mapuje http requesty na metody. Lze specifikovat, které http požadavky má obsluhovat

```
@RequestMapping(value = "/test/info", method = POST)
@ResponseBody
public String hello() {
    return „Hello World!";
}
```

- Takto označená metoda odpoví na POST requesty z adresy:  
<http://localhost:8080/test/info>

# Anotace – @RequestParam

- Tato anotace umožňuje získávat parametry z URL adresy

```
@GetMapping(„hello“)  
public String hello(@RequestParam String name) {  
    return „Hello “ + name + „!“;  
}
```

- Dotazem na <http://localhost:8080/hello?name=David> získáme:  
Hello David!

# Anotace – další spousta

- **@Required** – proměnná musí být naplněná v době konfigurace
- **@Autowired** – vkládá instance do Bean
- **@Configuration** – využívá se pro vytvoření vlastních instancí Bean
- **@Bean** – pro vytváření Spring Bean
- **@Component** – určuje, že se tato třída zpracovává Springem
- **@Service** – pro třídy s business logikou, výpočty a volání dalších API
- **@Repository** – pro třídy, které přímo přistupují do DB
- ...

**Prostor pro dotazy**

# Společná ukázka



# Samostatný úkol

- Vytvořte třídu s názvem Student, která bude obsahovat:
  - String: surname, name
  - Int: id, year
  - Konstruktor v pořadí (surname, name, id, year) – pořadí důležité pro sdílený kód
- Z elerningu stáhnete předpřipravený seznam, který vložíte do listu students
- Vytvořte novou třídu s názvem **StudentsController**, ve které bude vaše práce
- Zde vytvořte metody, které obslouží tyto požadavky:
  - [/student](#) vypíše Student: **VašeJméno** ID: **VašeID**, Jméno a ID bude tučně
  - [/student?name=Test&id=123456](#) vypíše místo vašich hodnot zadané hodnoty
  - [/students](#) vypíše seznam všech studentů ze staženého listu
  - [/students?vutid=123456](#) vypíše pouze konkrétního studenta, pokud v listu nebude, napíše se, že [neexistuje](#). (Studenta zobrazíte z objektu třídy Student a bude tedy obsahovat jméno, příjmení, id a rok narození)
- První 2 body můžete zpracovat pouze jako String (i jejich výchozí hodnoty)

# Dokončení úkolu

- Samostatnou práci ukazujete vyučujícímu v hodině (**max. 2 b**)
- Kdo nestihne, má možnost odevzdat do elerningu, avšak maximum bodů je sníženo na 1 b, (musí být vidět práce v hodině)
- Odevzdáno musí být vždy **do čtvrtka do 23:59**.