

# CV01 – Základy Spring Boot

Cílem cvičení je seznámit studenty se základy Spring ekosystému pro tvorbu backend aplikací. Konkrétně Spring Boot.

## Předpoklady

- ☕ Znalost jazyka Java a OOP (minimálně Java 1.8).
- 🌲 Verzovací systém Git (orientace v GitHub nebo GitLab výhodou).
- 🧠 Pokročilejší orientace v modernějším IDE.















## Výstupy

- Student zvládne vytvořit Spring Boot aplikaci.
- Vytvořit jednoduchý GET endpoint, který vrátí jako odpověď text/plain.
- Využívat parametry URL adresy (path, query, request).

## Zadání


1. Inicializujte Spring Boot aplikaci pomocí Spring Initializeru: <https://start.spring.io>
  - a. Gradle projekt (ideálně Gradle – Groovy).
  - b. Jazyk Java (případně Kotlin).
  - c. Verze: 2.7.8.
  - d. Project metadata vyplňte vhodně dle uvážení.
  - e. Packaging jar.
  - f. Verzi Javy zvolte tak aby byla kompatibilní s tou, kterou používáte.
2. Do projektu přidejte následující závislosti:
  - a. Project Lombok
  - b. Spring web
  - c. Spring Boot DevTools
3. Vygenerujte základní kostru projektu a uložte jej na úložiště zařízení.
4. 🖋️ V projektu inicializujte nový Git repositář (můžete použít Git Bash, případně prostředky dostupné v IDE).
5. 🖋️ Vytvořte inicializační commit s kostrou projektu.
6. 🖋️ Vytvořte si účet na GitHub, GitLab nebo obdobné službě.
7. 🖋️ Pushněte commity na vzdálený repositář.
8. 🤖 Teoretická 5 minutovka.
9. V adresáři pro zdrojové kódy vytvořte novou Java/Kotlin třídu s názvem HelloController.java/HelloController.kt.
10. Třidu opatřete anotacemi:
  - a. @RestController
  - b. @RequestMapping("/api/v1")
  - c. ? Do komentáře nad třídou napište rozdíl mezi anotací @RestController a @Controller.
11. Do třídy vložte novou metodu s následující signaturou:


```
@GetMapping("")
public String helloWorld() {
    return "Hello world from Spring Boot application.";
}
```

12. Spusťte Spring Boot aplikaci pomocí příkazu: `gradle bootRun` v rootu projektu.
  - a. Z logu detekujte, na jaké portu se aplikace spustila.
  - b. Přes webový prohlížeč navštivte adresu: `http://localhost:<port>/api/v1/`
  - c.  Stáhněte si aplikaci Postman a zkuste přes něj provolat GET request se stejnou adresou.
13.  Teoretická 5 minutovka.
14.  Vytvořte další tři endpointy (metody) a to tak aby:
  - a.  Přijímal v path parametru libovolný řetězec, který pak zahrňte do odpovědi.
  - b.  Přijímal v query parametru libovolný řetězec, který pak zahrňte do odpovědi.
  - c.  Obsahoval Request Body s libovolnými atributy, které pak zahrňte do odpovědi.
    - i.  Request body bude další parametr metody a bude to již složitější objekt, který budete muset realizovat další Java/Kotlin třídou.
      1.  Třidu a její atributy opatřete gettery/settery, toString metodou a parametrickým konstruktorem (klasicky dodržte zapouzdření).
      2.  Při funkčnosti řešení zkuste tuto třídu přepsat s využitím Lomboku (tedy tak aby obsahovala pouze anotace a atributy, nic jiného).
      3.  V jazyce Kotlin je možné využít tzv. data classess.
15.  Teoretická 5 minutovka.
16. Prozkoumejte možnosti, které Vám nabízí soubor `application.properties`.
  - a. Změňte pomocí něho port, na kterém aplikace běží na 9000.
  - b.  Odeberte anotaci `@RequestMapping` a definujte `basePath` pomocí `application.properties` souboru.
  - c.  Přepište soubor do konvence YAML neboli `application.yaml`.
  - d.  Do komentáře ve třídě `HelloController.java` napište rozdíl mezi YAML, JSON a XML syntaxí.

## Vysvětlivky

 Samostatná práce s podporou vyučujícího.

 Teoretická otázka. Odpověď studenti napíší do komentáře v kódu.

 Teoretická 5 minutovka.

## Odkaz pro odevzdání

[https://docs.google.com/forms/d/e/1FAIpQLScF2SQqaPbB5uxp-5Et4RewSX0v4KBuRBSA6eEeA02NiSy\\_gA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLScF2SQqaPbB5uxp-5Et4RewSX0v4KBuRBSA6eEeA02NiSy_gA/viewform?usp=sf_link)