

# Java Technologies - Lab 2

[valid 2025-2026]

## Spring Boot

### Compulsory (1p)

- Create a Spring Boot application using `spring initializr`. Add dependencies for Dev Tools, Spring Web.
  - Change the port at which Tomcat runs to 8081.
  - Add an endpoint `/hello` returning a greeting message defined in `application.properties`.
  - Add support for Spring Actuator in `pom.xml` and test some endpoints.
- 

### Homework (2p)

Create a Spring Boot application that must run in multiple environments, such as development, and production. Each environment uses a different database, and the database connection details are provided in `application-{profile}.yml`. The application should use JDBC for database access, conditional bean creation with Spring Expression Language (SpEL) and config property injection.

- The host, port, database name, user and password will be specified in the environment.
  - Choose between `@ConfigurationProperties` or `@Value` in order to read config data and motivate your decision.
  - Ensure that the application connects to the correct database automatically based on the active profile by printing some values from a database table; use a `CommandLineRunner` or `ApplicationLineRunner`.
  - Verify that command-line arguments take precedence over the ones in `application.yml`.
  - Define a scenario where `@Profile` is used together with `@ConditionalOnExpression` in order to choose the right database.
  - Implement a custom Actuator endpoint or an `InfoContributor` to return information about the database to which the application is connected.
- 

### Advanced (2p)

Create a scenario in which multiple services must share configuration but also adapt to different environments (dev, prod), using Spring Cloud Config.

- A Config Server backed by Git
  - Two Config Clients (Service A and Service B)
  - Environment-specific profiles
  - Runtime refresh for dynamic configuration updates
  - Prove that your setup is working (screenshots, server/client logs, etc.)
- 

## Resources

- Spring Boot Reference Documentation
- YAML: YAML Ain't Markup Language
- Spring Profiles

- JDBC Basics