

Property files, logging, profiles

# Property files

---

- ▶ `application.properties` is the default properties file in Spring Boot
- ▶ Typically it is found in the `src/main/resources` directory and this is because Spring Boot will look for it there, it will be auto-detected
- ▶ The `application.properties` file can be used to override default configuration
- ▶ For example, change the port of a webapp from default 8080 to 8081:
- ▶ `server.port=8081`

# Logging

---

- ▶ Logging is an important aspect of any software application
- ▶ It helps troubleshooting and debugging issues
- ▶ Spring Boot comes pre-configured with logging support
- ▶ Spring Boot offers simple configuration options for logging in the application.properties file
- ▶ Spring Boot provides support for multiple log levels such as TRACE, DEBUG, INFO, WARN and ERROR

# Profiles

---

- ▶ Profiles in Spring Boot are a powerful feature that allows developers to configure and manage different environments or deployment scenarios
- ▶ With profiles you can easily specify different sets of configuration options for different environments, for example use different databases when running the project in development or production
- ▶ The active profile can be set with `spring.profiles.active` in `application.properties` or as an environment variable
- ▶ The configuration for the profile `prod` is specified in the alternative property file `application-prod.properties`

# Demo 1 - Logging and properties

---

- ▶ Using the application.properties file
- ▶ Logging and logging levels
- ▶ Using profiles