

Programowanie komponentowe – ćwiczenie 1

Cel ćwiczenia:

Zbudowanie prostej aplikacji webowej Hello World!

Niezbędne narzędzia:

1. IDE - Spring Tool Suite 4
2. Maven wersja 3.5.4 lub wyższa (dostępny w STS)
3. JDK wersja 8.0 lub wyższa

1. Sprawdzenie zainstalowanych wersji.

```
$ java -version
```

```
java version "1.8.0_102"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
```

```
$ mvn -v
```

```
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T14:33:14-04:00)
```

```
Maven home: /usr/local/Cellar/maven/3.3.9/libexec
```

```
Java version: 1.8.0_102, vendor: Oracle Corporation
```

2. Utworzenie pliku pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd>
```

```
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com.example</groupId>
```

```
  <artifactId>myproject</artifactId>
```

```
  <version>0.0.1-SNAPSHOT</version>
```

```
  <parent>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-parent</artifactId>
```

```
    <version>2.4.0</version>
```

```
  </parent>
```

```
</project>
```

Wykonanie testu z IDE lub z lini komend: mvn package

3. Dodanie bibliotek (dependencies)

Spring Boot dostarcza dependency w postaci starterów. Spring-boot-starter-parent dostarcza domyślne konfiguracje w tym [dependency-management](#). Inne startery są używane do budowy specyficznych typów aplikacji, np. spring-boot-starter-web jest wykorzystywany do budowy aplikacji webowych.

```
<dependencies>
```

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

Wyświetlenie listy zależności jest możliwe z poziomu IDE lub z linii komend:

```
$ mvn dependency:tree
```

4. Dodanie kodu aplikacji

Domyślnie maven kompiluje kod z katalogu src/main/java. Należy dodać odpowiednią strukturę katalogów a następnie umieścić w niej plik HelloWorld.java zawierający poniższy kod.

```
import org.springframework.boot.*;
import org.springframework.boot.autoconfigure.*;
import org.springframework.web.bind.annotation.*;
```

```
@RestController
```

```
@EnableAutoConfiguration
```

```
public class HelloWorld {
```

```
    @RequestMapping("/") String home() {
        return "Hello World!";
    }
```

```
    public static void main(String[] args) {
        SpringApplication.run(HelloWorld.class, args);
    }
```

```
}
```

Adnotacja `@RestController` oznacza, że klasa pełni rolę kontrolera aplikacji webowej obsługującego przychodzące żądania.

Adnotacja `@RequestMapping` pozwala zamapować żądanie ze ścieżką „/” na wywołanie metody `home()`.

Adnotacja `@EnableAutoConfiguration` włącza automatyczną konfigurację Spring Boot na podstawie dołączonych starterów, np. konfigurację Spring MVC i Tomcat.

Metoda `main()` uruchamia aplikację Spring Boot za pomocą metody `run()` z klasy `SpringApplication`.

5. Uruchomienie aplikacji

Aplikację można uruchomić za pomocą IDE, mvn lub utworzyć wykonywalne archiwum jar.

```
$ mvn spring-boot:run
```

```

.  _ _ _ _ _
^ \ / _ ' _ _ _ ( ) _ _ _ _ \ \ \ \
(( \ _ | ' | ' | ' | ' \ _ ' | \ \ \ \
W _ ) | | ) | | | | | | ( | | ) ) ) )
' | _ _ | . _ | _ | | _ | _ \ _ | / / / /
=====|_|=====|_|_/_/_/_/
:: Spring Boot :: (v2.4.0)

.....
..... (log output here)

.....
..... Started Example in 2.222 seconds (JVM running for 6.514)

```

Utworzenie wykonywalnego archiwum wymaga dodania pluginu do pliku pom.xml

```

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

```

Po kompilacji mvn package otrzymamy plik myproject-0.0.1-SNAPSHOT.jar dostępny w katalogu target. Plik można uruchomić za pomocą java -jar.

```
$ java -jar target/myproject-0.0.1-SNAPSHOT.jar
```

```

.  _ _ _ _ _
^ \ / _ ' _ _ _ ( ) _ _ _ _ \ \ \ \
(( \ _ | ' | ' | ' | ' \ _ ' | \ \ \ \
W _ ) | | ) | | | | | | ( | | ) ) ) )
' | _ _ | . _ | _ | | _ | _ \ _ | / / / /
=====|_|=====|_|_/_/_/_/
:: Spring Boot :: (v2.4.0)

.....
..... (log output here)

.....
..... Started Example in 2.536 seconds (JVM running for 2.864)

```

6. Testowanie działania aplikacji.

Domyślnie aplikacja uruchamia się na tomiecie na porcie 8080. Wpisując w przeglądarce adres: <http://localhost:8080> otrzymamy następujący wynik

Hello World!