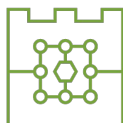


Spring Boot

Na podstawie J.B. Ottinger, A. Lombardi - "Beginning Spring 5 From Novice to Professional"



Politechnika Krakowska
im. Tadeusza Kościuszki



**Wydział Informatyki
i Telekomunikacji**

Studia magisterskie niestacjonarne
Informatyka, 1 rok, 1 semestr

Autorzy projektu:

Michał Mucha nr albumu 99222

Bartosz Szymański nr albumu 148976

Hubert Goncerz nr albumu 148940

Tomasz Rosiek nr albumu 148967

Michał Stęplewski nr albumu 149793

Czym jest Spring Boot?

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Sposobów definicji Spring Boot jest wiele, co również tyczy się samego Springa. Oto kilka potencjalnych kandydatów określić:

- ✓ jest to projekt, który łączy wiele najczęściej używanych elementów Springa w jedną spójną całość, dzięki czemu zarządzanie zależnościami jest łatwiejsze,
- ✓ środowisko wykonywalne, dzięki któremu wdrożenie bez kontenera jest całkiem wykonalne.
- ✓ Spring Boot ma na celu umożliwienie wykonywania projektów bez kontenera takiego jak Apache Tomcat.

Czym jest Spring Boot?

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Spring Boot to lekki framework pozwalający na proste tworzenie aplikacji w oparciu o framework Spring. Spring Framework jest to platforma, której głównym celem jest uproszczenie procesu tworzenia oprogramowania w technologii Java/J2EE. Rdzeniem Springa jest kontener wstrzykiwania zależności, który zarządza komponentami i ich zależnościami. Obiekty zarządzane przez Springa nazywane są beanami.

Konfigurowanie projektu

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

W zasadzie konfiguracja projektu w Spring Boot należy do łatwych czynności. Całość jest opisana na bazie konwencji narzuconej przez Gradle.

- ✓ Zaczynamy od stworzenia struktury katalogowej.
- ✓ Następnym krokiem jest pobranie i zainstalowanie Gradle oraz zapoznanie się z jego możliwościami.
- ✓ Zbudowanie kodu Javy przy użyciu Gradle'a.
- ✓ Deklaracja zależności.
- ✓ Budowanie projektu przy pomocy Gradle Wrapper.

Testowanie kodu

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Testy swojego kodu można zaimplementować na jeden z dwóch sposobów

- ✓ bezpośrednio wywołanie metod kontrolera
- ✓ symulowanie kompletnych zapytań HTTP

Testowanie kodu

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Bezpośrednie wywołanie metod kontrolera

```
@Test(dataProvider = "greetingData")

    public void testDirectGreeting(String name, String greeting) {

                                                                    assertEquals(
greetingController.greeting(name).getMessage(), greeting);

    }
```

- ✓ Czy rzeczywiście można kompletnie przetestować kontroler w ten sposób?

Testowanie kodu

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Czy można lepiej? Co dodatkowego testujemy dzięki pełnemu zapytaniu HTTP?

- ✓ Parsowanie parametrów
- ✓ Mapowanie URL
- ✓ Właściwy format zwracanej odpowiedzi

Testowanie kodu

Spring Boot

Setup



Politechnika Krakowska
im. Tadeusza Kościuszki

```
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class TestGreetingController extends AbstractTestNGSpringContextTests {
    @Autowired
    private GreetingController greetingController;
    @LocalServerPort
    private int port;
    @Autowired
    private TestRestTemplate restTemplate;

    @DataProvider
    Object[][] greetingData() {
        return new Object[][]{
            new Object[]{null, "Hello, world!"},
            new Object[]{"World", "Hello, World!"},
            new Object[]{"Andrew", "Hello, Andrew!"},
            new Object[]{"Jack Griffin", "I don't know who you are."}
        };
    }
}
```


Testowanie kodu

Spring Boot

Test

```
@Test(dataProvider = "greetingData")
public void testRestGreeting(String name, String greeting) {
    String url = "http://localhost:" + port + "/greeting/" +
        (name != null ? name : "");

    ResponseEntity<Greeting> result =
        restTemplate.getForEntity(url, Greeting.class);

    assertEquals(result.getStatusCode(), HttpStatus.OK);
    assertEquals(result.getBody().getMessage(), greeting);
}
```



Spring Boot oraz połączenie z bazą danych.

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Jak łączymy się do bazy danych?

Kroki do stworzenie aplikacji:

1. Zależności
2. Konfiguracja z bazą danych
3. Encję
4. Repozytorium

Spring Boot oraz połączenie z bazą danych.

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Dodawanie zależności oraz niezbędnych sterowników

Aby mieć możliwość korzystania z API od Spring Data należy dodać zależności:

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>
```

Dodatkowo niezbędną zależnością jest sterownik bazy danych:

```
<dependency>  
  <groupId>com.h2database</groupId>  
  <artifactId>h2</artifactId>  
  <scope>runtime</scope>  
</dependency>
```

Spring Boot oraz połączenie z bazą danych.

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Konfiguracja z bazą danych

Do pliku konfiguracyjnego application.properties należy umieścić następujące wpisy:

```
spring.jpa.properties.hibernate.hbm2ddl.auto=create
```

```
spring.datasource.url=jdbc:h2:file:./bazaDanych
```

```
spring.h2.console.enabled=true
```

```
spring.h2.console.path=/console
```

Spring Boot oraz połączenie z bazą danych.

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Tworzymy przykładową encję:

```
@Entity
public class VideoCassette {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    private Long id;
    private String title;
    private LocalDate productionYear;
    public VideoCassette() {
    }
    public VideoCassette(Long id, String title, LocalDate productionYear) {
        this.id = id;
        this.title = title;
        this.productionYear = productionYear;
    }
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public LocalDate getProductionYear() {
        return productionYear;
    }
    public void setProductionYear(LocalDate productionYear) {
        this.productionYear = productionYear;
    }
}
```

Spring Boot oraz połączenie z bazą danych.

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Repozytorium

Repozytorium umożliwia zarządzanie obiektami encji. Dzięki niemu możemy wykonywać operację:

- Dodawanie
- Usuwanie
- Modyfikacja
- Pobieranie
- Dodawanie innych operacji

Implementacja jest prosta. Wystarczy stworzyć interfejs, posiadający adnotację `@Repository` i rozszerzający interfejs `CrudRepository`. `CrudRepository` przyjmuje dwa parametry encji i typ `id`.

`@Repository`

```
public interface VideoCassetteRepo extends CrudRepository<VideoCassette, Long> {  
}
```

Podsumowanie

Spring Boot



Politechnika Krakowska
im. Tadeusza Kościuszki

Programowanie w Javie przy wykorzystaniu Spring Boot jest szybsze, łatwiejsze i bezpieczniejsze. Skupienie się na szybkości, prostocie i produktywności wpłynęły na to, że Spring to obecnie najpopularniejszy na świecie framework dla Javy.

- ✓ 62% ankietowanych developerów używa biblioteki Spring Boot, zgodnie z Java Developer Productivity Report
- ✓ Szybkość Budowania Aplikacji
- ✓ Dobra Dokumentacja
- ✓ Spring Framework umożliwia nam modularyzację aplikację
- ✓ Łatwy do wdrożenia