

# Zaawansowanych biblioteki programistyczne Java

Mateusz Sapała

# Spring

Spring jest frameworkiem aplikacyjnym oraz kontenerem odwrócenia sterowania dla aplikacji Java. Główne funkcje frameworku mogą być używane przez dowolną aplikację Java, ale jego rozszerzenia są przeznaczone głównie do tworzenia aplikacji webowych na platformie Java EE. Framework Spring jest projektem open source.


# Odwrócenie sterowania

Odwrócenie sterowania jest odpowiedzią na problem zależności pomiędzy klasami w kodzie. Przede wszystkim należy sobie uświadomić dlaczego posiadanie zależności w kodzie jest problemem. Po pierwsze, posiadanie zależności w kodzie prowadzi do powstawania dużej liczby zależności pomiędzy klasami co w konsekwencji może prowadzić do sytuacji gdzie zmiana w kodzie jednej klasy wymusza zmianę w kodzie innej klasy a co gorsza ta zmiana może być konieczna do zaaplikowania w wielu miejscach.

Drugim problemem wynikającym z powstawania zależności w kodzie jest problem z napisaniem dobrych testów jednostkowych. Powiązane klasy mogą okazać się problematyczne, a czasami wręcz niemożliwe do przetestowania z użyciem testów jednostkowych.

# Spring initializr

Spring Initializr to aplikacja internetowa, do generowania struktury projektu Spring Boot. Nie generuje ona żadnego kodu aplikacji, ale zapewnia podstawową strukturę projektu wraz ze specyfikacją kompilacji Maven lub Gradle. Aplikacja pozwala konfigurować listę obsługiwanych przez JVM zależności, język aplikacji oraz wersję Javy.



**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy

☒ **Maven**

**Spring Boot**  
☐ 3.0.1 (SNAPSHOT) ☒ **3.0.0** ☐ 2.7.7 (SNAPSHOT) ☐ 2.7.6

**Project Metadata**

Group

com.example

Artifact

demo1

Name

demo1

Description

Demo project for Spring Boot

Package name

com.example.demo1

Packaging

☒ **Jar** ☐ War

Java

☐ 19 ☒ **17** ☐ 11 ☐ 8

**Dependencies**

ADD DEPENDENCIES... CTRL + B

No dependency selected



GENERATE CTRL + G


EXPLORE CTRL + SPACE

SHARE...

# Przykład 1 - Spring Boot

# Spring Data JPA

Implementacja warstwy dostępu do danych od zawsze była uciążliwa. Konieczne było napisanie długiego identycznego kodu dla prostych zapytań. Zadaniem Spring Data JPA jest usprawnienie implementacji dostępu do danych poprzez zmniejszenie potrzebnego nakładu pracy. Programista ze swojej strony musi zdefiniować jedynie interfejsy swojego repozytorium natomiast Spring zapewni ich implementację.

 **spring**initializr

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy  
☒ **Maven**



**Spring Boot**  
☐ 3.0.1 (SNAPSHOT) ☒ **3.0.0** ☐ 2.7.7 (SNAPSHOT) ☐ 2.7.6

**Project Metadata**  
Group   
Artifact   
Name   
Description   
Package name   
Packaging ☒ **Jar** ☐ War  
Java ☐ 19 ☒ **17** ☐ 11 ☐ 8

**Dependencies** ADD DEPENDENCIES... CTRL + B

**Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

**MySQL Driver** SQL  
MySQL JDBC and R2DBC driver.



GENERATE CTRL + G

EXPLORE CTRL + SPACE


SHARE...

# Przykład 2 - Spring Data JPA

# Spring Web

Zadaniem Spring Web jest uproszczenie procesu tworzenia aplikacji internetowych, w tym aplikacji RESTful. Jest on zgodny z wzorcem MVC oraz implementuje odwrócenie sterowania oraz wstrzykiwanie zależności. Spring Web używa klasy DispatcherServlet, która rozdziela przychodzące żądania na odpowiednie zasoby, takie jak widoki i kontrolery.



 **spring** initializr

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Maven**

**Language**  
☒ **Java** ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.0.1 (SNAPSHOT) ☒ **3.0.0** ☐ 2.7.7 (SNAPSHOT) ☐ 2.7.6

**Project Metadata**  

Group

Artifact

Name

Description

Package name

Packaging ☒ **Jar** ☐ War



Java ☐ 19 ☒ **17** ☐ 11 ☐ 8

**Dependencies** ADD DEPENDENCIES... CTRL + B

**Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

**MySQL Driver** SQL  
MySQL JDBC and R2DBC driver.

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

GENERATE CTRL + G EXPLORE CTRL + SPACE SHARE...

# Przykład 3 - Spring Web

# Co dalej?

- ▶ <https://start.spring.io/>
- ▶ <https://github.com/MateuszSapala/AdvanceProgrammingTechniquesPresentation>
- ▶ <https://spring.io/guides>
- ▶ <https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring>
- ▶ <https://spring.io/guides/gs/accessing-data-jpa/>
- ▶ <https://spring.io/guides/gs/accessing-data-mysql/>
- ▶ <https://spring.io/guides/gs/rest-service/>
- ▶ <https://spring.io/guides/tutorials/rest/>

# Zaawansowanych biblioteki programistyczne Java

Mateusz Sapała