

Spring, Spring Data & Spring IO

Pavol Švidraň, Sebastián Ivan, Lukáš Löbl, Filip Michal Gajdoš

Software Framework

- ▶ predstavuje abstrakciu (generalizáciu) software-u, ponúka generalizáciu funkcionalít, ktoré sa dajú jednoducho upravovať kódom používateľa
- ▶ prináša štandardizovaný spôsob na tvorbu a nasadenie aplikácií pomocou univerzálneho a opakovane použiteľného softvérového prostredia
- ▶ môže zahŕňať aj podporné programy, kompilátory, knižnice, toolset-y alebo APIs
- ▶ rozdiely medzi framework-om a knižnicami:
 - 1) inverzia kontroly – kód napísaný programátorom dostáva flow control od frameworku
 - 2) základné správanie - Template Method Pattern v abstraktných triedach framework-u
 - 3) rozšíriteľnosť – používateľ (programátor) vie rozšíriť framework o vlastnú funkcionality
 - 4) Nemodifikovateľný kód framework-u – používateľ (programátor) by nemal upravovať zdrojový kód frameworku-u
- ▶ príklady: Laravel, Cactus framework, ...

Web Framework

- ▶ navrhnutý na tvorbu webových aplikácií v ktorých sú zahrnuté web services, web resources a APIs
- ▶ predstavuje štandardizovaný spôsob vývoju webových aplikácií, ktoré sú určené pre World Wide Web (WWW)
- ▶ sú zamerané na automatizáciu správy aktivít spojených s webovým vývojom: knižnice na správu databáz, templating, správa relácie (session), viacnásobné využitie rovnakého kódu
- ▶ hlavné zameranie na dynamické weby
- ▶ príklady: Spring, Zend framework, Catalyst, Apache Wicket, ...

Spring Framework

- ▶ open-source webový framework pre Java platformu
- ▶ zahrnuté moduly:
 1. Spring Core Container – základný modul pre spring kontajnery (BeanFactory and ApplicationContext)
 2. Autentifikácia a autorizácia - konfigurácia bezpečnostných procesov
 3. Prístup k dátam (Data access) - práca a správa relačných databáz (NoSQL)
 4. Zasielanie správ (Messaging) - konfigurácia a registrácia message listener-ov
 5. Model-view-controller - poskytuje rozšírenia a prispôsobenie pre webové aplikácie a RESTful Web services
 6. Remote access framework - vzdialené vykonávanie funkcionalít, ktoré by boli za bežných okolností vykonávané lokálne
 7. Transaction management - spracovávanie viacerých transakcií od odlišných klientov
 8. Vzdialená správa - konfigurácia obnaženia Java objektov pre lokálnu a vzdialenú konfiguráciu
 9. Testovanie - podpora tvorby tried učených pre unit a integračné testy



Spring Data

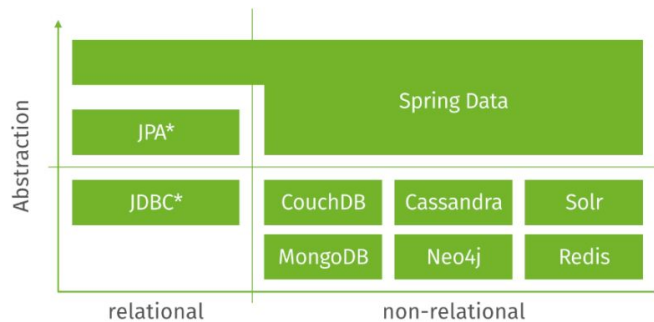
Spring Data Obsah:

- ▶ Čo je/sú Spring Data ?
- ▶ Vlastnosti
- ▶ Hlavné Moduly
- ▶ Informácie o niektorých moduloch
- ▶ Výhody a Nevýhody Spring Data

- ▶ High level SpringSource projekt
- ▶ Zjednocuje, uľahčuje a poskytuje konzistentnú vrstvu prístupu k údajom pre rôzne úložiska údajov.

Spring Data

Spring Data



Vlastnosti Spring Data

Výkonné úložisko a
vlastné abstrakcie
mapovanie objektov

Odvodzovanie
dynamického dotazu
z názvov metód
úložisko

Základné
implementačné
doménové triedy
poskytujúce
základné vlastnosti

Podpora
transparentného
auditu

Možnosť integrácie
vlastného kódu pre
úložisko

Experimentálna
podpora pre
cross-store
perzistenciu

Integrácia Easy
Springu cez
JavaConfig a vlastné
menné priestory
XML

Pokročilá integrácia
so Spring MVC.

Hlavné moduly

- ▶ Spring Data Commons
- ▶ Spring Data JDBC
- ▶ Spring Data JPA
- ▶ Spring Data MongoDB
- ▶ Spring Data Rest
- ▶ Spring Data LDAP
- ▶ Spring Data Neo4j
- ▶ ...
- ▶ [Odkaz na ostatné moduly](#)

Spring Data JDBC

Spring Data JDBC má za cieľ byť konceptne jednoduchý.

Je persistentný framework, ktorý nie je taký zložitý ako Spring Data JPA

Spring Data JDBC neponúka generovanie schém

Spring Data JPA

- ▶ Spring Data JPA je jedným z mnohých projektov Spring Data.
- ▶ Mnoho ľudí považuje Spring Data JPA za implementáciu JPA.
- ▶ Čo je JPA Spring Data, ak nie implementácia JPA?

Spring Data MongoDB

- ▶ Spring Data MongoDB poskytuje integráciu s databázou dokumentov MongoDB.
- ▶ Klúčové funkčné oblasti Spring Data MongoDB sú POJO

Spring Data REST

Spring Data REST stavia na repozitároch Spring Data, analyzuje model domény vašej aplikácie a odhaľuje hypermédiami riadené zdroje HTTP pre agregáty obsiahnuté v modeli.

Spring Data LDAP

Spring Data Neo4j

Ponúka tri rôzne úrovne abstrakcie pre prístup . Klient , šablóna a úložiská .

Výhody Spring Data

1.
Repository
pattern bez
kódu

2. Znižuje
štandardný
kód

3. Generuje
queries

Nevýhody Spring Data

Paralelný mechanizmus - Veľký výber možností.

```

public interface MongoRepository<T, ID> extends PagingAndSortingRepository<T, ID>, QueryByExampleExecutor<T> {
    <S extends T> List<S> saveAll(Iterable<S> entities);

    List<T> findAll();

    List<T> findAll(Sort sort);

    <S extends T> S insert(S entity);

    <S extends T> List<S> insert(Iterable<S> entities);

    <S extends T> List<S> findAll(Example<S> example);

    <S extends T> List<S> findAll(Example<S> example, Sort sort);
}

```

```

@Repository
public interface StudentRepository extends MongoRepository<Student, String> {

    @Query("{ 'name' : ?0 }")
    Student findStudentByName(String name);
}

```

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {  
    <S extends T> S save(S entity);  
  
    <S extends T> Iterable<S> saveAll(Iterable<S> entities);
```

```
    Optional<T> findById(ID id);
```

```
    boolean existsById(ID id);
```

```
    Iterable<T> findAll();
```

```
    Iterable<T> findAllById(Iterable<ID> ids);
```

```
    long count();
```

```
    void deleteById(ID id);
```

```
    void delete(T entity);
```

```
    void deleteAllById(Iterable<? extends ID> ids);
```

```
    void deleteAll(Iterable<? extends T> entities);
```

```
    void deleteAll();  
}
```

```
    public Iterable<Student> getAllStudents() { return studentRepository.findAll(); }
```

```
    public void deleteAllStudents() { studentRepository.deleteAll(); }
```

```
    public void deleteStudentById(String id) { studentRepository.deleteById(id); }
```

```
    public Optional<Student> findStudentById(String id) { return studentRepository.findById(id); }
```

```
    public Student findStudentByName(String name) { return studentRepository.findStudentByName(name); }
```

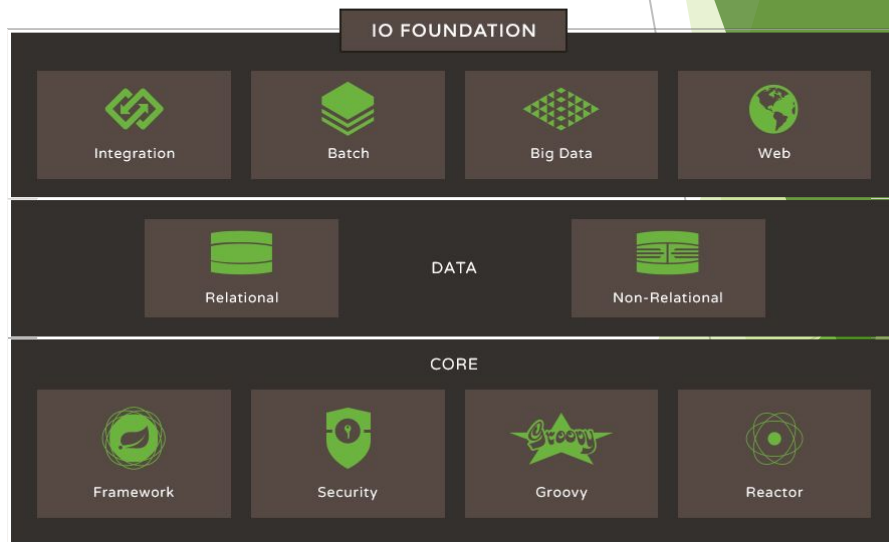
Spring IO platform

- ▶ Súdržná, verzovaná platforma na vytváranie aplikácií a podporu závislostí
- ▶ Primárne určenie na použitie s dependency management systémom
- ▶ Zahŕňa moduly Foundation Layer a DSR Execution Layer
- ▶ Funguje už s existujúcimi dependency management nástrojmi ako Maven a Gradle
- ▶ Vlastnosti :
 - ▶ Open source
 - ▶ Lean Code (štíhly kód) - jednoduchosť, efektívnosť, flexibilita a dostupnosť
 - ▶ Jedna platforma mnoho využití (Spring Boot, Spring Framework, Spring Batch ...)
 - ▶ Zjednodušený vývoj so Spring Boot
 - ▶ Výber a zjednotenie závislostí, ktoré sú kompatibilné a pracujú spolu

- ▶ Execution Layer poskytuje DSRs (Domain Specific Run Times) na dramatické zjednodušenie vytvárania „production ready“ úloh založených na Java virtual machine (JVM).



- ▶ Foundation Layer predstavuje základné zjednotené Spring moduly a súvisiace závislosti/komponenty tretích strán pre zabezpečenie priebehu vývoja.



Spring IO platform verzie

- Spring AMQP 2.0.12
- Spring Batch 4.0.3
- Spring Cloud Connectors 2.0.5
- Spring Data Kay-SR14
- Spring Framework 5.0.13
- Spring HATEOAS 0.25.1
- Spring Integration 5.0.13
- Spring Retry 1.2.4
- Spring Security JWT 1.0.10
- Spring Security OAuth 2.2.4
- Spring Security 5.0.12
- Spring Session Apple-SR9
- Spring Web Services 3.0.7

Cairo-SR8

- Spring AMQP 1.7.13
- Spring Boot 1.5.20
- Spring Cloud Connectors 1.2.8
- Spring Data Ingalls SR20
- Spring Framework 4.3.23
- Spring Integration 4.3.20
- Spring Retry 1.2.4
- Spring Security 4.2.12
- Spring Security JWT 1.0.10
- Spring Security OAuth 2.0.17
- Spring Web Services 2.4.5

Brussels-SR1
7

- Spring Boot 1.4.7
- Spring Cloud Connectors 1.2.4
- Spring Data Hopper SR 11
- Spring Framework 4.3.9
- Spring Integration 4.3.10
- Spring REST Docs 1.1.3
- Spring Security OAuth 2.0.14
- Spring Web Flow 2.4.5

Athens-SR
6

Nadenie s Maven

Platforma využíva podporu Maven na správu závislostí pre poskytnutie verzií závislostí zostavovanej aplikácie.

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>io.spring.platform</groupId>
      <artifactId>platform-bom</artifactId>
      <version>Cairo-SR8</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Nastavenie BOM-u Platformy
ako rodič POM-u Mavenu

Importovanie BOM-u (Bill of Materials)
Platformy do POM-u (Project Object Model)
súboru Mavenu

```
<parent>
  <groupId>io.spring.platform</groupId>
  <artifactId>platform-bom</artifactId>
  <version>Cairo-SR8</version>
  <relativePath/>
</parent>
```

Nasadenie s Gradle

- V prípade Gradle, importovanie BOM Platformy vyzerá podobne ako u Maven, ale musíme použiť Dependency Management Plugin.
- Použitie tohto pluginu je nevyhnutné, pretože Gradle neposkytuje rovnakú vstavanú podporu pre správu závislosti ako je u Maven-u.

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'io.spring.gradle:dependency-management-plugin:1.0.0.RELEASE'  
    }  
}  
  
apply plugin: 'io.spring.dependency-management'  
  
repositories {  
    mavenCentral()  
}  
  
dependencyManagement {  
    imports {  
        mavenBom 'io.spring.platform:platform-bom:Cairo-SR8'  
    }  
}
```

- ▶ Nie je potrebný Spring Boot na manažovanie projektov.
- ▶ Bez ohľadu na využitý prístup, do aplikácie nebudú pridané žiadne závislosti.
- ▶ Po deklarovaní závislosti obsiahnutej v Spring IO Platformy, deklarovanie verzie nie je nutné.

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
  </dependency>
</dependencies>
```

Mave
n

```
dependencies {
  compile 'org.springframework:spring-core'
}
```

Gradl
e

Koniec životnosti Spring IO Platformy

- ▶ 2014 - 2019
- ▶ Znižujúce sa rozdiely medzi Spring Boot a Spring IO Platformou.
- ▶ Zrušenie podpory platformy a prechod na Spring Boot.
- ▶ Nasadenie Spring Boot rovnaké ako u platformy a to pomocou POM alebo nastavenia ako parent.
- ▶ Podpora pre Maven ako aj Gradle.

Softvér

- Správa zoznamu študentov
 - Pridávanie
 - Odoberanie
 - Editácia
 - Prezeranie záznamov

Create new student

ID: _____

Name: _____

Surname: _____

Subject: _____

Age: _____

List of students

Name	Surname	Subject	Age	Action	
Pavol	Švidraň	ASOS	25	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
Michal	Učienlivý	ASOS	24	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

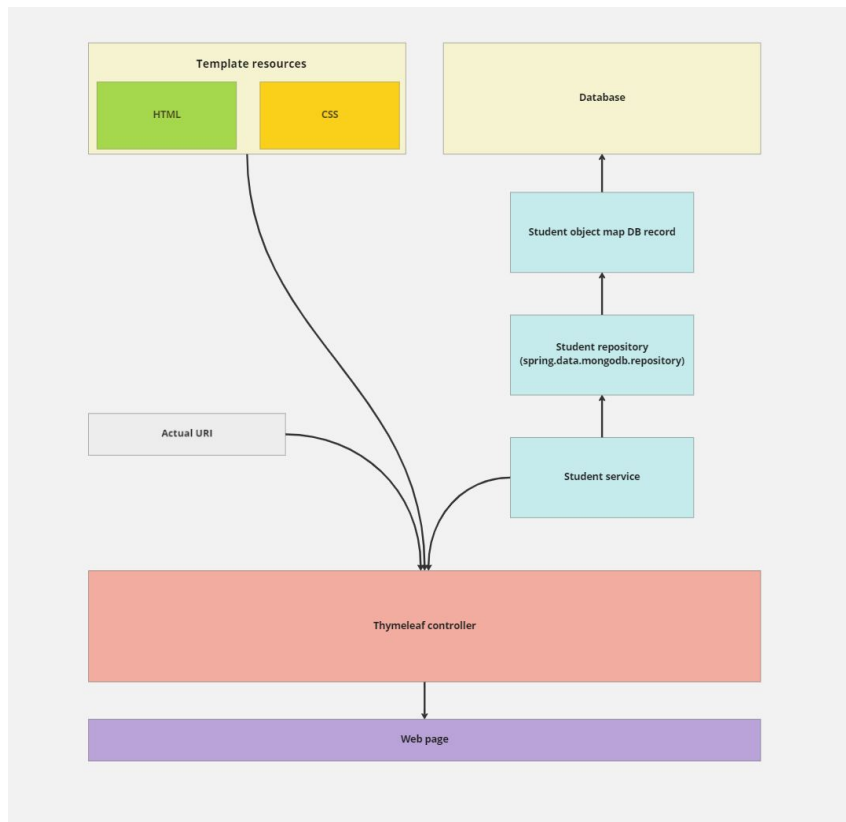
Softvér - UI

- ▶ Thymeleaf
 - ▶ Kompatibilný so Spring Boot
 - ▶ Serverový renderovací engine
 - ▶ MVC
 - ▶ Templatovanie html, xhtml, css ...
 - ▶ Routing za pomoci custom kontrolera

Softvér - uchovávanie dát

- ▶ Využitie Spring.Data.MongoDb
 - ▶ Pridávanie, úprava, mazanie záznamov
 - ▶ Mapovanie na Java objekty
 - ▶ Jednoduchá zmena db zdroja

Softvér - všeobecná schéma



Otázka na záver

- ▶ Čo sa považuje za výhody pre Spring Data?
 - ▶ Automatické generovanie queries
 - ▶ Paralelný mechanizmus
 - ▶ Repository pattern

Otázka na záver

- ▶ Čo sa považuje za výhody pre Spring Data?
 - ▶ Automatické generovanie queries ✓
 - ▶ Paralelný mechanizmus
 - ▶ Repository pattern ✓

Zdroje

- ▶ [Software Framework](#)
- ▶ [Web Framework](#)
- ▶ [Spring Framework](#)
- ▶ [Spring Data](#)
- ▶ [Spring Data Intro](#)

► Spring IO Platform

- <https://platform.spring.io/platform/>
- <https://docs.spring.io/platform/docs/current/reference/html/getting-started-using-spring-io-platform.html>
- <https://docs.spring.io/platform/docs/current/reference/htmlsingle/#maintenance>
- <https://spring.io/blog/2018/04/09/spring-io-platform-end-of-life-announcement>
- <https://mvnrepository.com/artifact/io.spring.platform/platform-bom>
- https://tadviser.com/index.php/Product:Spring_IO
- <https://stackoverflow.com/questions/34798744/understanding-spring-boot>
- <https://stackoverflow.com/questions/24438751/what-is-the-relationship-between-spring-boot-and-spring-io>
- <https://spring.io/projects>
- <https://spring.io/blog/2019/04/04/spring-io-platform-cairo-sr8>
- <https://spring.io/blog/2019/04/04/spring-io-platform-brussels-sr17>
- <https://spring.io/blog/2017/06/15/spring-io-platform-athens-sr6>