Rozdiel medzi Spring MVC a Spring Boot

Bc. Michal Rendek, Bc. Simona Richterová, Bc. Kristián Sokol, Bc. Barbora Ulrichová, Bc. Edita Včelková

Spring

- Open-source framework
- Rod Johnson
- Plain Old Java objects
- Vysokovýkonný, ľahko testovateľný a opakovane použiteľný kód
- Jednoduchá konfigurácia
- Efektívne programovanie

Rod Johnson je generálnym riaditeľom spoločnosti Atomist, vývojovej automatizačnej spoločnosti. Vytvoril Java's Spring Framework a bol spoluzakladateľom a generálnym riaditeľom SpringSource.

Ako funguje Spring?

PREZENČNÁ VRSTVA

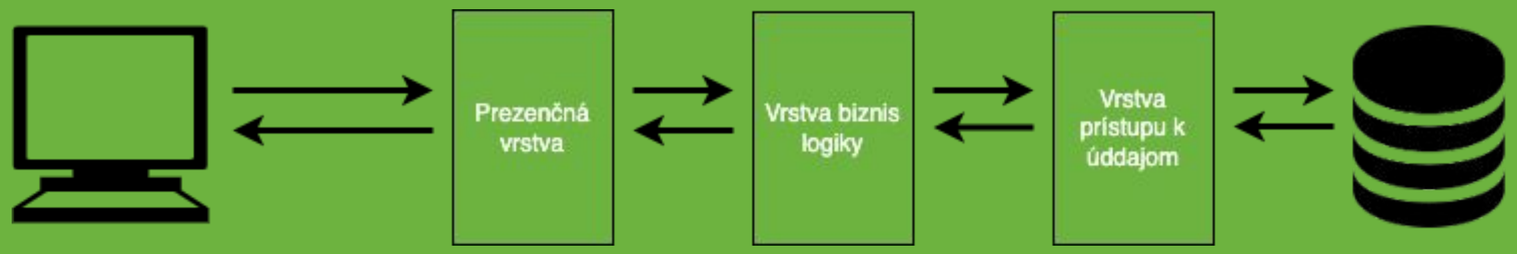
Stará o prezentáciu obsahu a interakciu s používateľom.

VRSTVA BIZNIS LOGIKY

Centrálna vrstva, ktorá sa zaoberá logikou programu.

VRSTVA PRÍSTUPU K ÚDAJOM

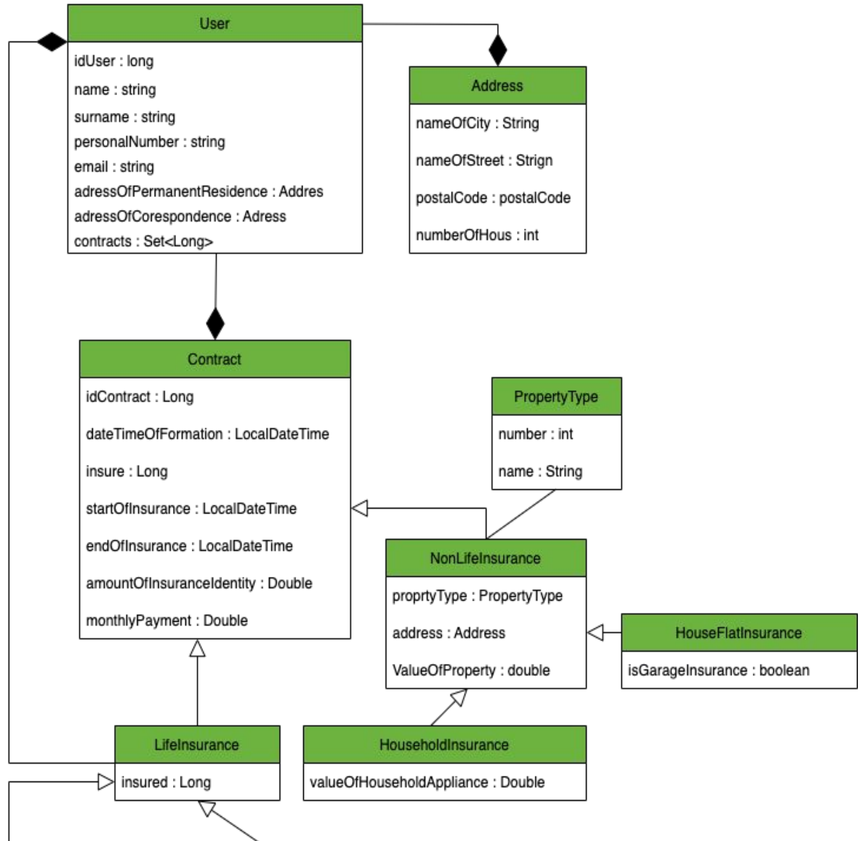
Hlboká vrstva, ktorá sa zaoberá získavaním údajov zo zdrojov.



Ako funguje Spring?

Na integráciu a prácu používa jednotný prístup a platformu založenú na princípoch:

- IoC
- Aop
- DI



@component

Dáva Springu vedieť, ktoré triedy má spravovať (vytvárať). Označí beans (objekty) ako spravované komponenty, čo znamená, že Spring bude tieto triedy autodetekovať.

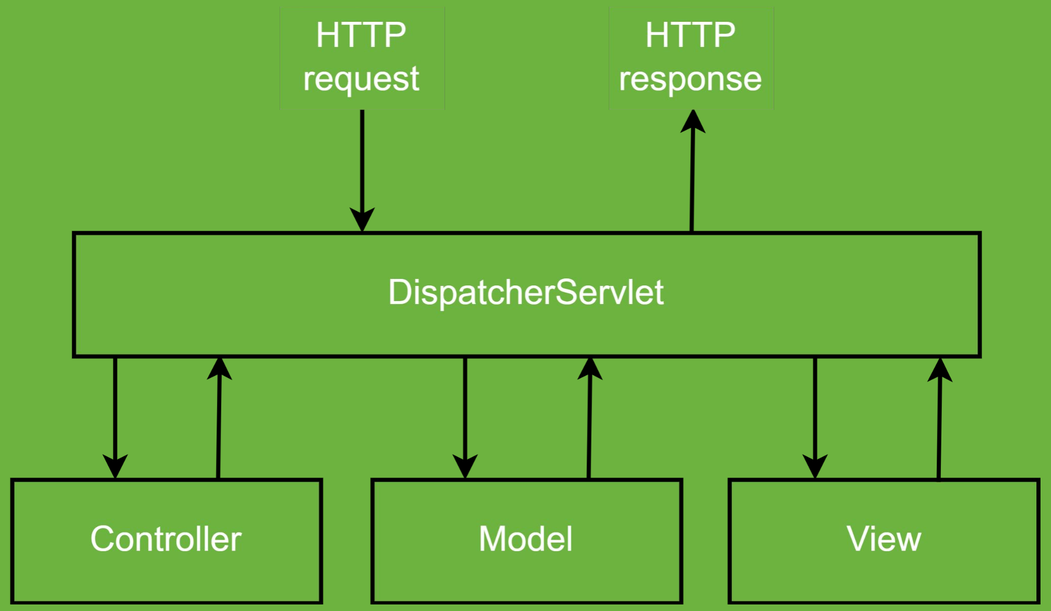
@autowired

Informuje Spring, ako má postupovať pri inštanciácii triedy (začne hľadať danú závislosť medzi komponentmi/triedami, aby našiel zhodu). To ušetrí vývojárov od spájania kódu a umožní to, aby Spring našiel, čo je potrebné kde injektovať.

SPRING MVC

Spring MVC

- Nadstavba na Spring
- Vytváranie webových aplikácií
- Generuje HTML
- Návrhový vzor MVC



Spring MVC

MODEL

Zapuzdrenie dát aplikácie vo všeobecnosti pozostávajúcich z POJO.

VIEW

View vykresľuje údaje modelu a tiež generuje HTML výstup.

CONTROLLER

Controller spracuje požiadavky používateľov a odovzdá ich do zobrazenia na vykreslenie.

Výhody

NASADENIE

Rýchle pomocou lightweight servlet ako kontajner

VÝVOJ

Efektívne prepájanie prvkov prostredníctvom MVC a možnosť paralelnej práce

AKTUALIZÁCIA

Je jednoduchšia pretože stačí iba zmeniť zdrojové súbory na serveri alebo inej platforme

VIAC ÚROVNÍ

Umožňuje rýchlejšie ladenie aplikácie

ŠKÁLOVATEĽNOSŤ

Spring MVC zabezpečuje vysokú mieru škálovateľnosti čiže rozširovanie systému

RÁMCE

Aplikácia oddeľuje rámce pre jednoduchšie vykonávanie.

Nevýhody

ROBUSTNOSŤ

Nie je vhodný pre malé aplikácie. Pretože negatívne ovplyvňuje výkon a dizajn aplikácie.

NÁROČNOSŤ

Je náročný na pochopenie a tvorbu aplikácie, zvlášť pre začiatočníka.

ZASTARANOSŤ

Objemný so starším kódom, čo zbytočne zvyšuje náročnosť aplikácie.

Použitie

Spring MVC Framework funguje nasledovne:

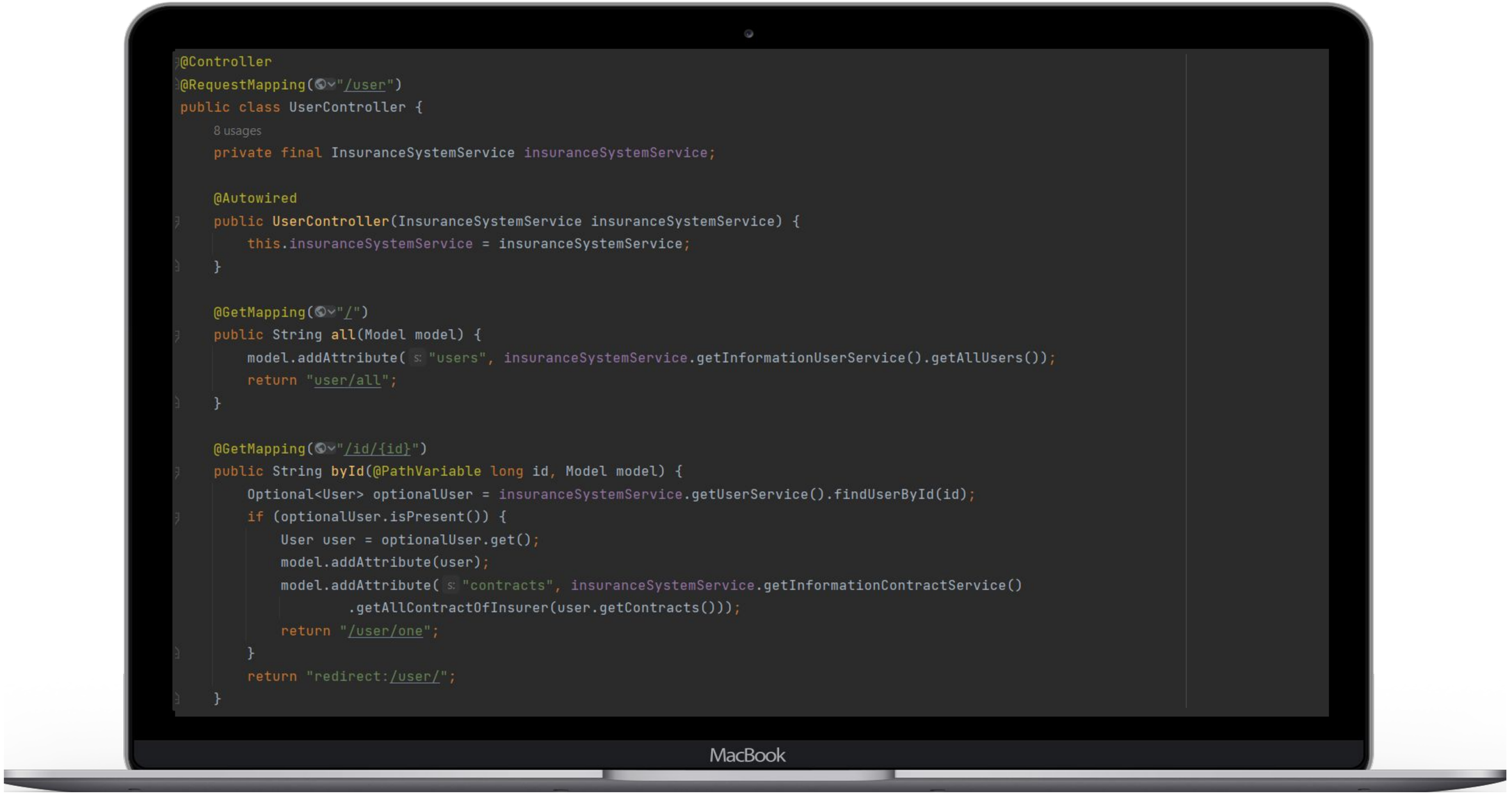
- Všetky prichádzajúce požiadavky sú zachytené DispatcherServletom, ktorý funguje ako front controller.
- DispatcherServlet potom získa záznam mapovania obsluhy zo súboru XML a odovzdá požiadavku kontroleru.
- Objekt ModelAndView je vrátený kontrolerom.
- DispatcherServlet skontroluje záznam resolveru view v súbore XML a zavolá príslušný view komponent.

Závislosť pre použitie template language Thymeleaf:

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

spring MVC

Použitie



TEMPLATE ENGINES

Template engines

THYMELEAF

- Server-side
- Rýchly vývoj
- Vytváranie šablón bez backendu
- HTML šablóny
- Open-source knižnica

APACHE FREEMARKER

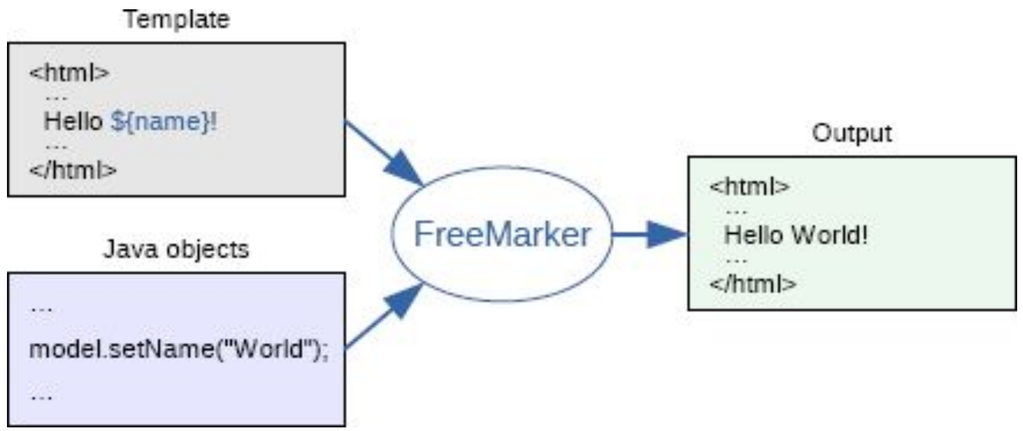
- Java knižnica
- Generovanie textového výstupu
- Jazyk - FreeMaker Template Language (FTL)

MUSTACHE

- Šablóny bez logiky
- Neexistujú príkazy ako if, else ani cykly for
- Len značky

Thymeleaf

<#FREEMARKER>



Template engines

APACHE VELOCITY

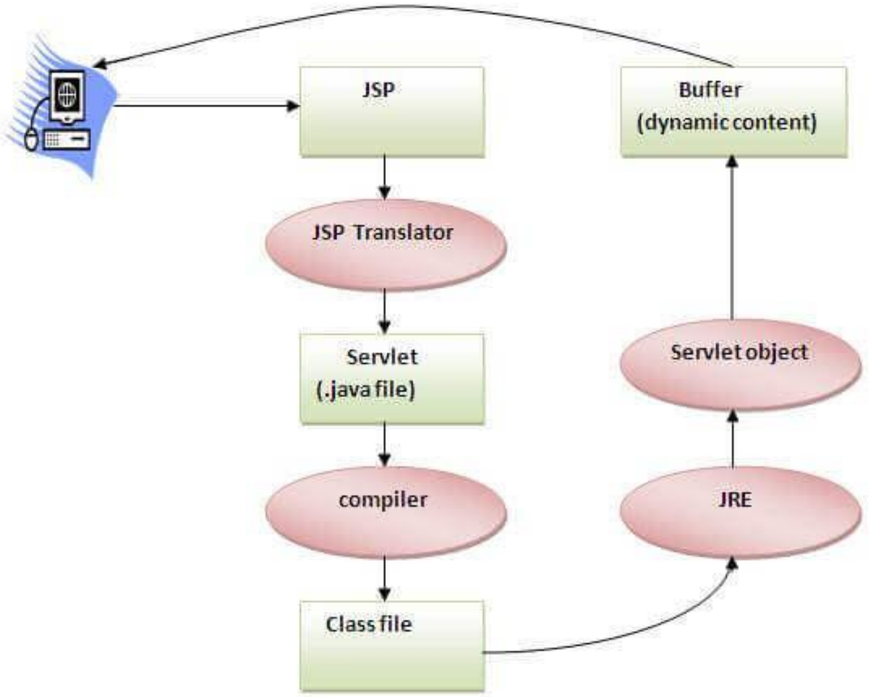
- Životaschopná alternatíva k JSP
- Generovanie SQL, PostScript a XML zo šablón
- Open-source

Apache Velocity™

JSP - JAVASERVER PAGES

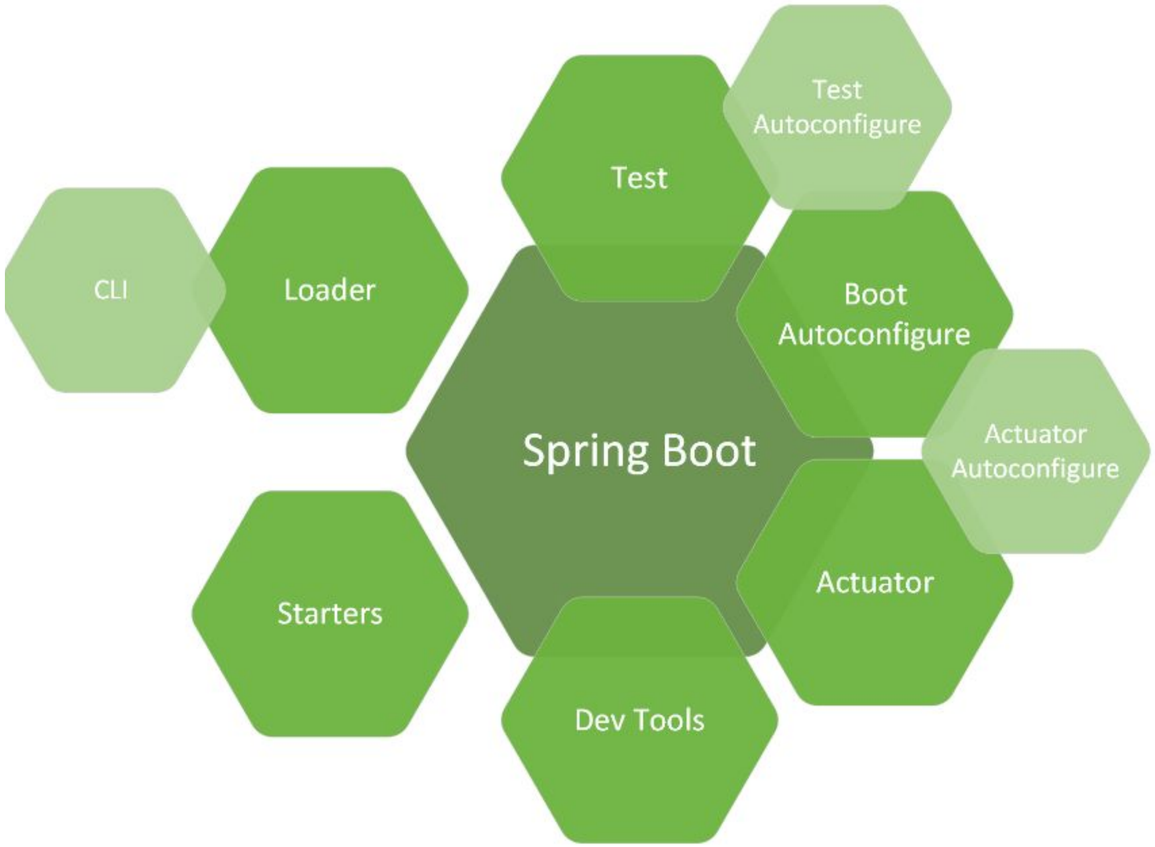
- Server-side
- Možno považovať ako rozšírenie servletu
- Vlastné výrazy, značky
- Jednoduchá údržba
- Rýchly vývoj

JSP



SPRING BOOT

Spring Boot



- Vytvorenie mikroslužieb
- Minimálna konfigurácia
- Lightweight framework
- Abstrakciu na softvér ORM
- Podporuje framework JDBC
- Vývoj rozhraní REST API

spring boot

Výhody

INTEGRÁCIA

Jednoduchosť integrovať aplikáciu s jej ekosystémom Spring, ako sú Spring JDBC, Spring ORM, Spring Data, Spring Security atď.

AUTOKONFIGURÁCIA

Pomáha vyhnúť sa manuálnej práci pri písaní boilerplate kódu, anotáciách a zložitých konfiguráciách XML.

PLUGINY

Poskytuje množstvo pluginov na veľmi jednoduchú prácu s embedded a in-memory databázami.

SERVERY

Dodáva sa s integrovanými HTTP servermi, ako sú Jetty a Tomcat, na vývoj a testovanie webových aplikácií.

CLI

Poskytuje nástroj CLI na vývoj a testovanie Spring Boot aplikácií z príkazového riadku veľmi jednoducho a rýchlo.

RÝCHLOSŤ

Rýchly a jednoduchý vývoj aplikácií založených na technológii Spring;

Výhody

SERVLET KONTAJNER

Zjednodušuje závislosť a dodáva sa s vloženým Servlet kontajnerom.

POM

Zjednodušená správa závislostí bez konfliktov verzií prostredníctvom štartovacích POM.

SLEDOVANIE ÚDAJOV

Spring Boot poskytuje koncové body HTTP na prístup k interným údajom aplikácie, ako sú podrobné metriky, vnútorná práca aplikácie, stav atď.

Nevýhody

NEDOSTATOK KONTROLY

Vytvára veľa nepoužívaných závislostí, čo má za následok veľký súbor pre nasadenie.

ZLOŽITÁ KONVERZIA

Zložitý a časovo náročný proces konverzie staršieho alebo existujúceho projektu Spring na aplikáciu Spring Boot

LIGHTWEIGHT FRAMEWORK

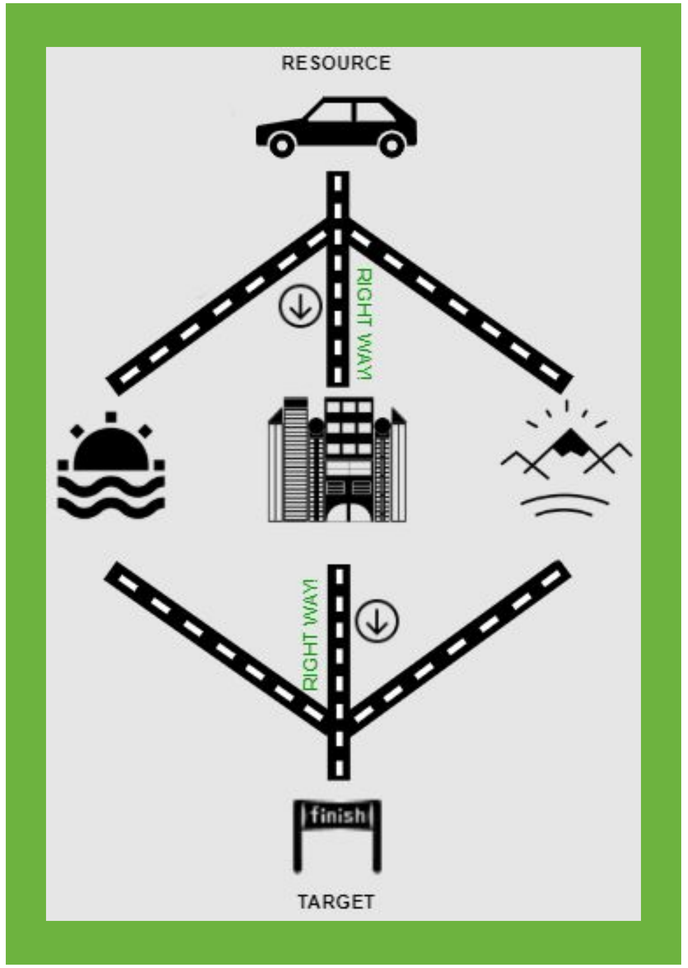
Hoci je skvelý na prácu s mikroslužbami, mnohí vývojári tvrdia, že Spring Boot nie je vhodný na vytváranie monolitických aplikácií.

Použitie

- Medzi hlavné funkcie Spring boot patria:
  - Standalone (samostatná),
  - Opinionated (názorová),
  - Autoconfiguration (automatická konfigurácia).

spring boot

Opinionated stratégia



Použitie

- JPA =>
  - databáza v pamäti,
  - zdroj údajov,
  - hibernate entity manager.
- spring-boot-starter-data-jpa

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

Použitie

- Poskytuje sadu závislostí - Spring Boot Starter
- spring-boot-starter- *

Webová závislosť Spring Boot Starter sa používa na zápis Rest endpointov:

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

Závislosť Spring Boot Starter Security sa používa pre Spring Security:

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

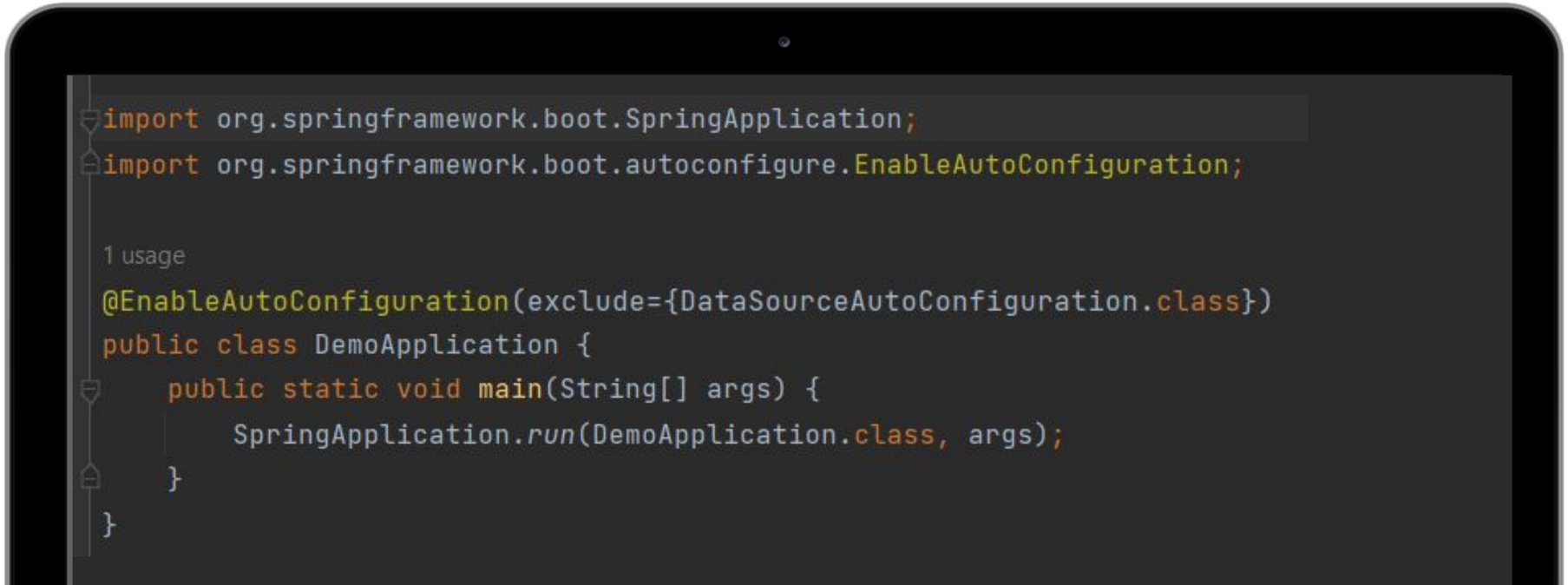
Použitie

- Spring Boot automaticky konfiguruje aplikáciu na základe závislostí
- Potrebné do main class pridať anotáciu:
  @EnableAutoConfiguration alebo **@SpringBootApplication**
- Trieda obsahujúca túto anotáciu a metódu main je vstupným bodom Spring Boot aplikácie.

Použitie

- **@EnableAutoConfiguration -** umožňuje selektívne vylúčiť určité triedy z automatickej konfigurácie pomocou atribútu exclude.

- **@ComponentScan**
- **@Configuration**

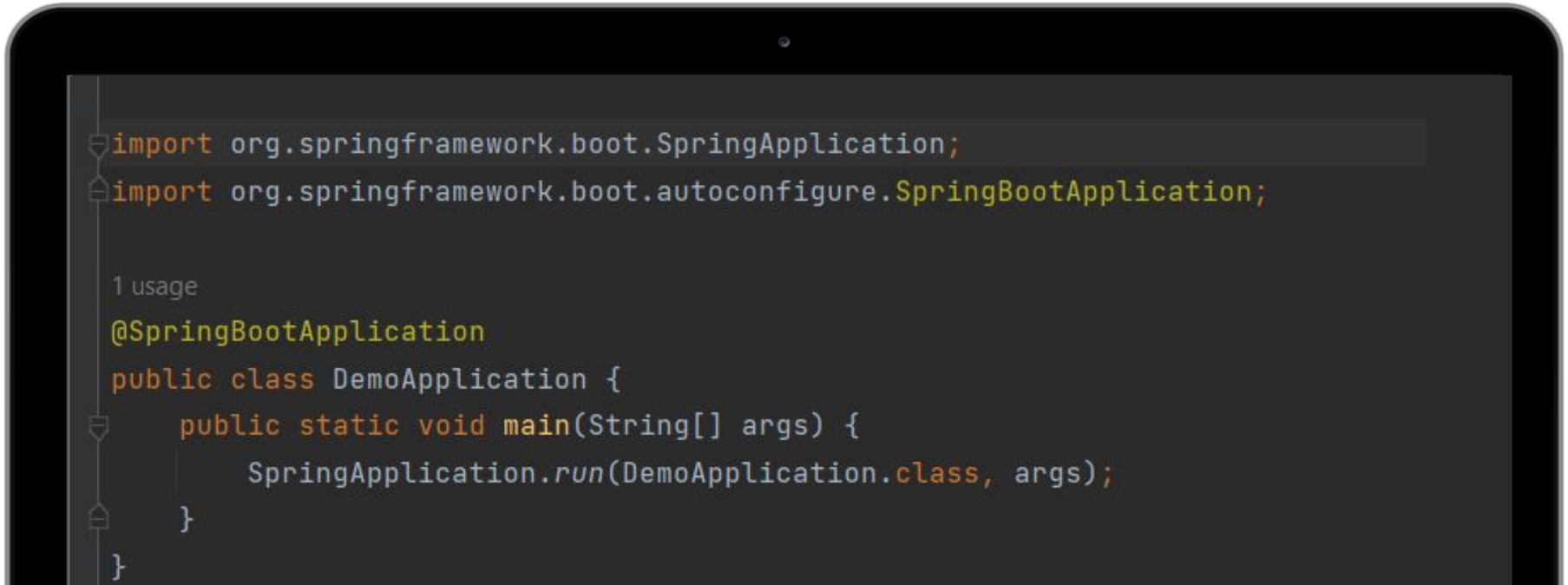


```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

1 usage
@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

Použitie

- **@SpringBootApplication** obsahuje:
  - @SpringBootConfiguration,
  - @ComponentScan ,
  - @EnableAutoConfiguration.

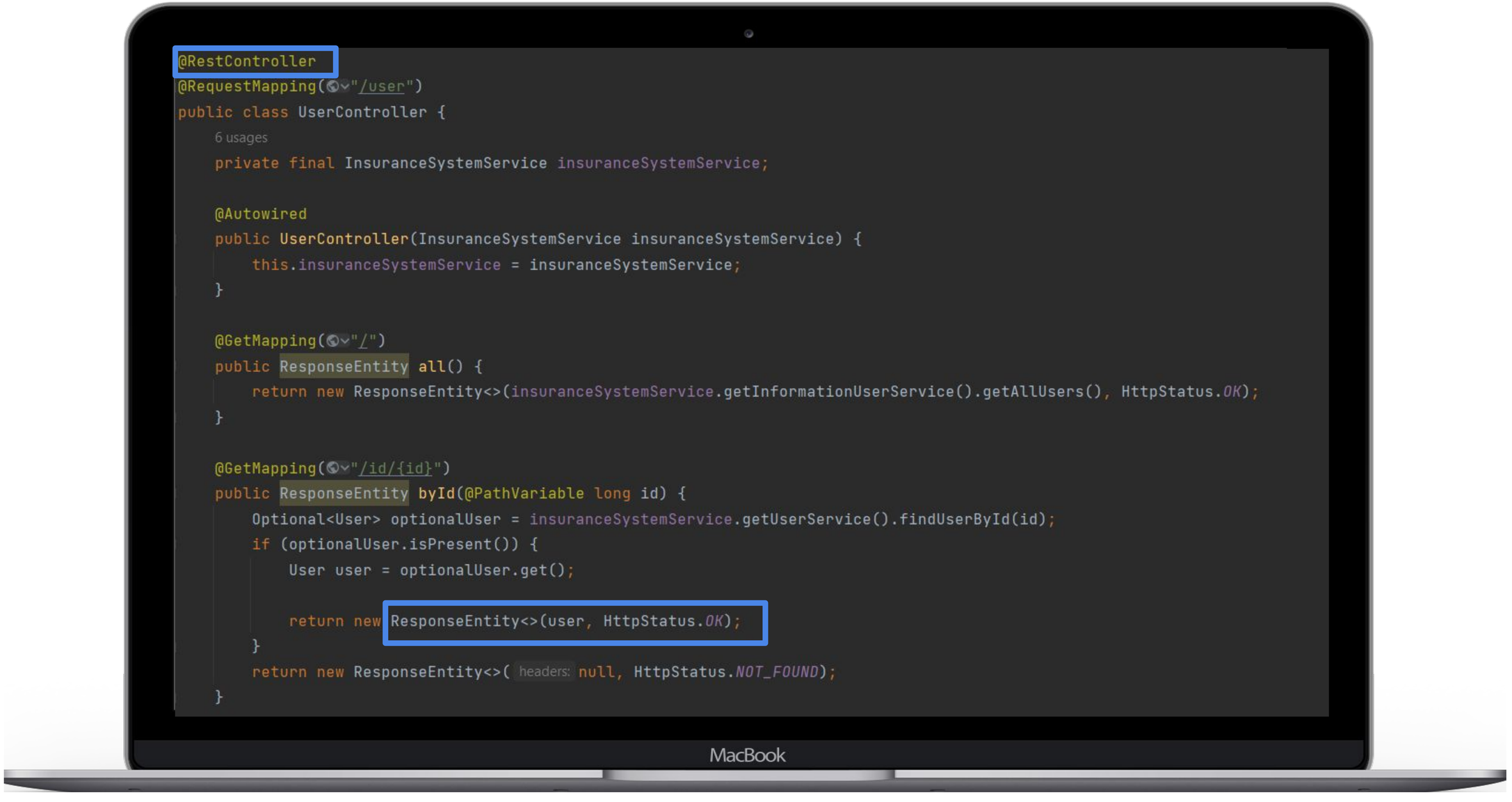


```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

1 usage
@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

spring boot

Použitie



POROVNANIE

Spring MVC

Spring MVC je framework založený na MVC.

Vyžaduje veľa konfigurovania, napríklad konfigurácie DispatcherServlet a konfigurácie View Resolveru.

Rýchlosť vývoja sa znižuje, pretože je potrebné porozumieť závislostiam doplnkov.

Spring Boot

Spring Boot je framework na tvorbu REST API.

Spring Boot spracúva konfiguráciu automaticky pomocou funkcie automatickej konfigurácie.

Kratší čas vývoja, produktivita sa zvyšuje.

Spring MVC

Každú dependency je potrebné špecifikovať samostatne, aby sa funkcie spustili.

Spring MVC vyžaduje veľa manuálnych konfigurácií pre JAR balenie.

Pre framework Spring MVC sa vyžaduje deskriptor nasadenia.

Spring Boot

Spring Booth má koncept štartérov, ktorý po pridaní do classpath poskytne všetky dependencies potrebné na vývoj webovej aplikácie.

Spring Boot umožňuje vstavanému serveru spustiť funkciu samostatne.

Pre framework Spring Boot sa nevyžaduje deskriptor nasadenia.

Spring MVC

Štyri komponenty Spring MVC sú Model, View, Controller a Front Controller (trieda DispatcherServlet).

Spring MVC je navrhnutý len na vývoj dynamických webových stránok a webových služieb RESTful.

Spring Boot

Štyri vrstvy Spring Bootu sú prezentačná vrstva, biznis vrstva, perzistenčná vrstva a databázová vrstva.

Spring Boot umožňuje vytvárať aj rôzne iné druhy aplikácií.

Spring MVC

Neposkytuje výkonné batch spracovanie.

Každá dependency je v Spring MVC špecifikovaná samostatne.

Spring Boot

Poskytuje výkonné batch spracovanie.

V Spring Boot sú všetky závislosti zabalené do jedného celku.

Porovnanie

- So Spring Boot sa pracuje oveľa pohodlnejšie ako so Spring MVC, ale výber môže závisieť aj od typu aplikácie.
- Oba tieto frameworky majú svoje výhody, napríklad Spring MVC je dobrou voľbou na vývoj modulárnych webových aplikácií a vynecháva prekrývajúcu sa úlohu spracovania požiadaviek HTTP.
- Na druhej strane Spring Boot pomáha šetriť čas a znižuje námahu tým, že znižuje počet konfigurácií. Jednou spoločnou vecou, ktorá môže spôsobiť zmätok, je používanie anotácií na vývoj aplikácie v oboch typoch technológií.

Otázka ku skúške

V návrhovom vzore MVC figurujú 3 vrstvy. Model, view a controller. Čo zabezpečuje vrstva controller?

a) Vykresľuje údaje modelu a tiež generuje HTML výstup.
b) Prepája komponenty, spracováva požiadavky používateľov a odovzdáva ich do view na vykreslenie.
c) Zapuzdruje dáta aplikácie a komunikuje s databázovým systémom.
d) Stará sa o reprezentáciu a manipuláciu s dátami v aplikácií.

Otázka ku skúške

V návrhovom vzore MVC figurujú 3 vrstvy. Model, view a controller. Čo zabezpečuje vrstva controller?

a) Vykresľuje údaje modelu a tiež generuje HTML výstup.

b) Prepája komponenty, spracováva požiadavky používateľov a odovzdáva ich do view na vykreslenie.

c) Zapuzdruje dáta aplikácie a komunikuje s databázovým systémom.

d) Stará sa o reprezentáciu a manipuláciu s dátami v aplikácií.

ZDROJE

- https://spring.io/projects/spring-boot
- https://www.interviewbit.com/blog/spring-vs-spring-boot/
- https://www.javatpoint.com/spring-vs-spring-boot-vs-spring-mvc
- Spring vs Spring Boot vs Spring MVC - javatpoint (www-javatpoint-com.translate.goog)
- Difference Between Spring MVC and Spring Boot - InterviewBit
- Spring vs Spring Boot vs Spring MVC - javatpoint
- https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
- https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/
- https://www.baeldung.com/spring-vs-spring-boot
- https://stackify.com/what-is-spring-boot/
- https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
- Spring - MVC Framework - GeeksforGeeks
- Difference Between Spring MVC and Spring Boot - InterviewBit
- Spring Boot Vs Spring MVC | Which to Choose? (kodytechnolab.com)
- 17. Web MVC framework (spring.io)
- https://data-flair.training/blogs/advantages-of-spring/

Ďakujeme za pozornosť