

Spring Boot

Дополнение к Spring, позволяющий создавать приложения, используя минимум конфигурирования Spring

Будем рассматривать на примере приложения “Игра по мотивам сказки”

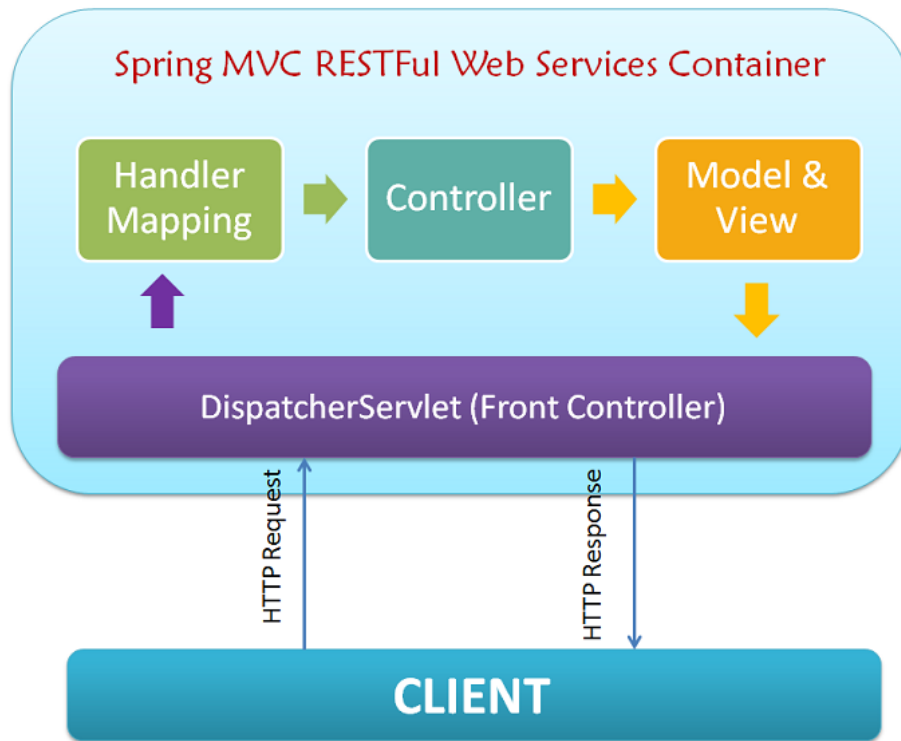
1. Создание проекта с помощью Spring initializr
2. Конфигурация
3. Использование шаблона Spring MVC
4. Работа с БД(Hibernate и SpringData)
5. Spring Security



Создание проекта с помощью Spring initializr и Конфигурирование

1. <https://start.spring.io/>
2. @SpringBootApplication класс, помеченный этой аннотацией используется как основной класс конфигурации приложения
3. Application.properties (свойства Spring Boot, полный список <https://docs.spring.io/spring-boot/docs/current/reference/html/application-properties.html>)
4. Аннотации для компонентов: @Component, @Service, @Controller
5. Варианты инжекта бинов (@Primary, @Qualifier)
6. @Scope
7. @PostConstruct

MVC



Контроллеры и представления - наша область ответственности в Spring MVC

Работа с базой данных. Создание простой сущности

Добавить нужные зависимости для работы с JPA и с выбранной БД

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

1. В файле application.properties добавить параметры подключения к БД

```
spring.datasource.url=jdbc:postgresql://localhost/magicians
spring.datasource.username=sd
spring.datasource.password=123
spring.jpa.generate-ddl=true
```

1. Создать сущность
2. Создать репозиторий для получения и сохранения этой сущности в БД
3. Создать сервис для оперирования сущностью
4. Сконфигурировать Spring, указав какие пакеты сканировать для поиска сущностей и репозиторийев (над классом MagicianApplication добавить аннотации @EnableJpaRepositories(""), @EntityScan(""))
5. В контроллере добавить метод для создания сущности и замапить его на нужный путь