

Практическое задание №6.

Подготовительная работа.

Для выполнения данного задания должна быть проведена подготовительная работа из задания №5 и корректно настроено подключение к БД.

Задание.

В рамках данной практической работы необходимо реализовать REST сервис, используя стандартные средства библиотеки spring boot starter, а также разработать демонстрационное приложение, использующее данный REST сервис. Помимо этого необходимо научиться использовать произвольные hibernate модели, не привязанные к изначальным таблицам БД.

1. Добавьте в pom файл проекта новый элемент parent для указания базовой библиотеки spring boot:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.22.RELEASE</version>
</parent>
```

2. Для реализации REST сервиса потребуется использовать зависимость spring-boot-starter-web

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

Обратите внимание, что указание версии в данном случае не требуется, поскольку она уже была указана в родительском элементе.

3. После всех зависимостей добавьте плагин сборки spring boot, который позволит запускать web-сервер без дополнительных настроек и подключений:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

4. После всех настроек не забудьте выполнить погрузку библиотек, если она не была настроена автоматически.
5. Для проверки запуска веб сервера создайте новый класс Application, который будет содержать классический метод main. Единственной строчкой в данном методе будет старт сервера:

```
SpringApplication.run(Application.class, args);
```

Для того, чтобы указать spring, что данный класс является точкой входа ему необходимо указать следующую аннотацию: @SpringBootApplication

6. Запустите приложение, проверьте, что web сервер функционирует
7. *Создайте новый класс, который будет являться основным REST контроллером. Для этого необходимо, чтобы у класса были указаны аннотации:

```
@RestController
@RequestMapping("/test")
```

где /test – точка обращения к данному контроллеру: localhost:8080/test

8. Добавьте в данный класс метод, использующий написанные ранее DAO классы, для получения списка объектов. Для этого добавьте аннотацию @GetMapping("students") к методу, где students – путь обращения к данному методу в рамках контроллера: localhost:8080/test/students.
Метод должен возвращать список заданных объектов, полученных из БД в формате JSON (см. <https://www.baeldung.com/java-org-json>).
9. Используя другие аннотации методов HTTP для REST добавьте в данный класс методы для создания нового объекта и удаления существующего по id (см. <https://spring.io/guides/tutorials/rest/>)
10. Проверьте работоспособность методов используя Rest-клиент
11. Создайте модель, включающую поля из нескольких таблиц БД. Используя синтаксис addScalar и setResultTransformer(Transformers.aliasToBean(CustomClass.class)), где CustomClass – класс созданной модели, выгрузите результаты запроса в List<CustomClass>
12. Реализуйте REST метода для получения данного списка в формате JSON.

Дополнительное задание

13. Разработайте демонстрационное приложение (используя любые изученные технологии), которое использует методы GET, POST, DELETE реализованного REST-API, а также способно отображать полученный через GET результат в виде таблицы.

***ПРИМЕЧАНИЕ:** для корректной работы DAO классов с spring boot может потребоваться несколько дополнительных настроек конфигурации:

1) Добавить в файл конфигурации hibernate следующие параметры:

```
<property name="org.hibernate.flushMode">ALWAYS</property>
<property name="hibernate.temp.use_jdbc_metadata_defaults">>false</property>
```

2) Переопределить метод настройки BulkIdStrategy у стандартного диалекта. Для этого необходимо создать новый класс наследник CacheDialect7, переопределить у него метод согласно инструкции ниже и использовать новый класс в hibernate.cfg

Инструкция: <https://community.intersystems.com/post/support-java-hibernate-5#comment-14726>