**Задание**

Разработать REST API приложение, которое позволит

- отправить sms сообщение

- принять sms сообщение

- отозвать sms сообщение

Покрыть функциональность unit тестами. Документировать через swagger.

**Инструменты**

Java 8, Spring Boot, СУБД PostgreSQL, в качестве ORM JOOQ, сервер приложения WildFly.

**Реализация**

1) Создать приложение Spring Boot, используя инициалайзер с официального сайта start.spring.io

Скачать полученный архив, открыть файл pom.xml

Возможно не все зависимости сразу прокачаются. Может быть что брандмауэр виндов будет блокировать закачку обновлений.

Была ошибка

Plugin 'org.springframework.boot:spring-boot-maven-plugin:' not found

Помогло. Запустить в командной строке

>mvn spring-boot:run

После этого дать разрешение для брандмауэра виндовс на закачку файлов.

Добавить в зависимость конкретную версию плагина

<plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <version>2.4.5</version>  
</plugin>

2) создал сервисный класс

services/SmsController.java

3) в методе main получил контекст и вывел все названия всех бинов, которые создаются при старте приложения.

Возникла ошибка

Web server failed to start. Port 8080 was already in use.

Исправил, путём добавления настоек в файл applicatonion.properties

server.port = 8090

Собрать приложение

mvn package

В результате в папке target будет нужная сборка restSMS-0.0.1-SNAPSHOT.war

Если просто запустить в браузере

<http://localhost:8090/>

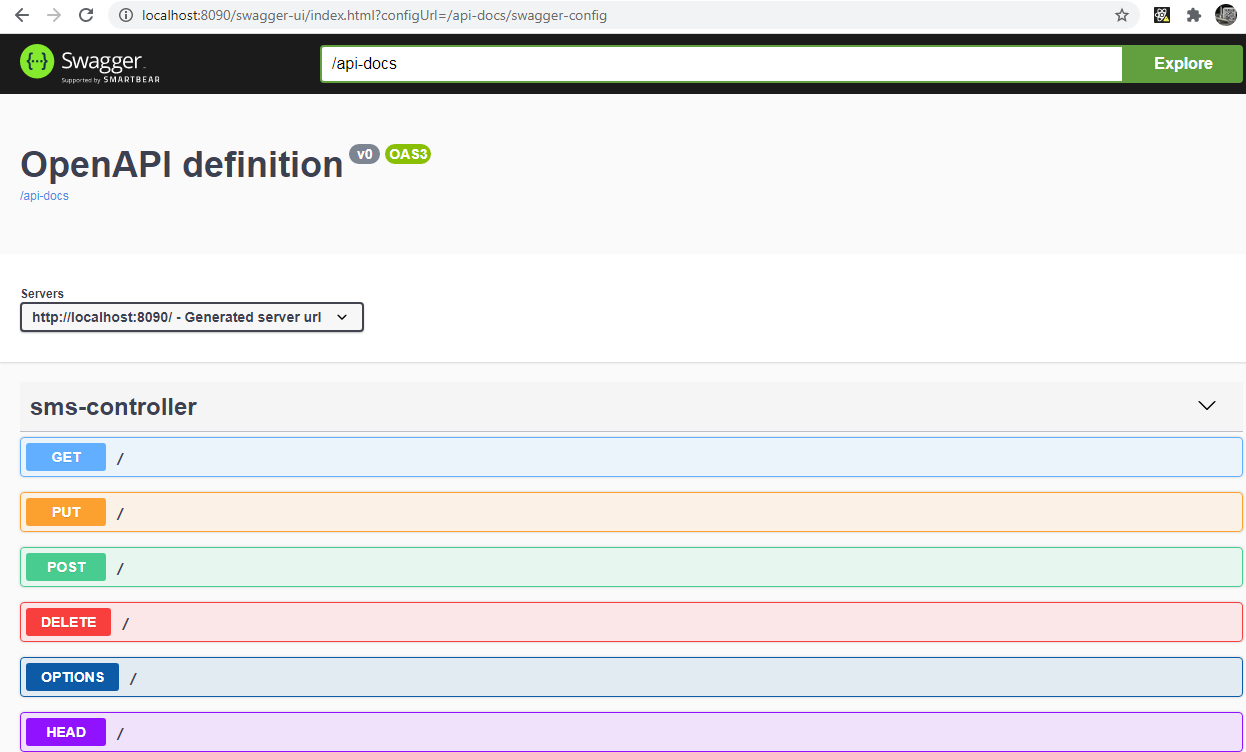
4) добавил документирование через Swagger (openAPI 3 )

Добавить зависимость org.springdoc.

Можно прописать путь, по которому автоматическая документация будет вызываться в application.properties

Вот так выглядит через UA

http://localhost:8090/swagger-ui/index.html?configUrl=/api-docs/swagger-config



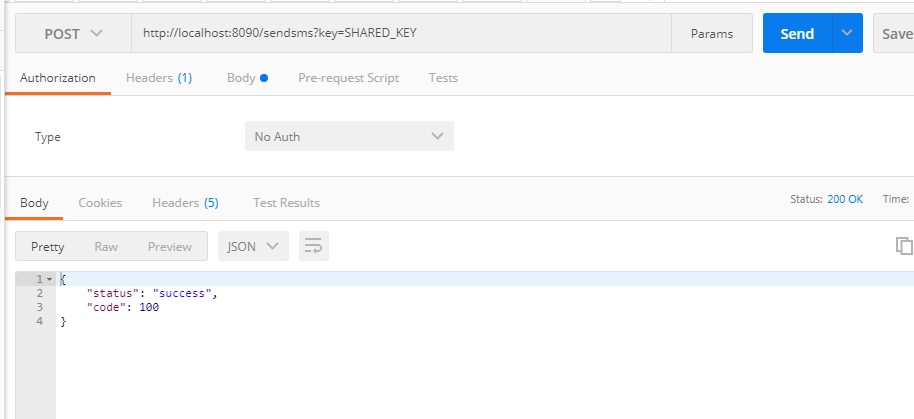
5) добавил реализацию классов

SmsRequest – запрос на СМС

BaseResponse – возвращает код ответа и значение

6) добавил реализацию метода @PostMapping("/sendsms") в контроллер SmsController. Для корректного ответа нужно передать параметр key

http://localhost:8090/sendsms?key=SHARED\_KEY



7) сделал сервис SmsController. В нём реализовал несколько методов GET, POST,PUT,DELETE

**Чтобы метод одновременно сервис мог возвращать и xml и json**

**Можно добавить**

produces = {MediaType.*APPLICATION\_JSON\_VALUE*, MediaType.*APPLICATION\_XML\_VALUE*}

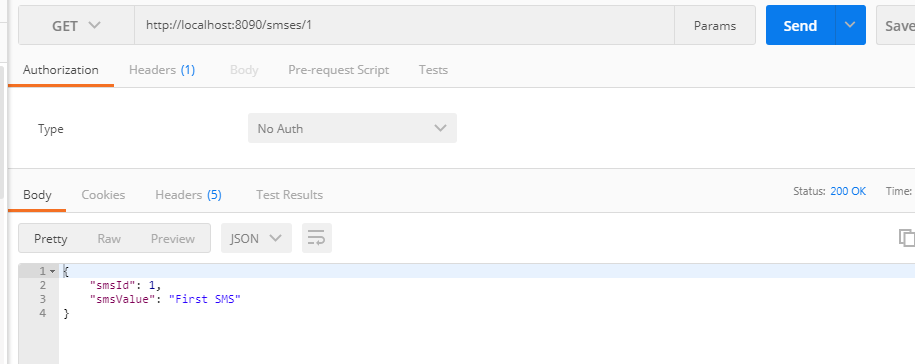
8) тестируем вручную через POST MAN.

Быстрее понять что они работают через swagger UA

Методы GET

http://localhost:8090/smses

http://localhost:8090/smses/1

****

Методы POST

<http://localhost:8090/smses>

body

{

"smsId": 4,

"smsValue": "4 SMS. I'm add a new SMS"

}

Метод PUT

<http://localhost:8090/smses>

тело запроса

{

"smsId": 2,

"smsValue": "Update SMS. Manual"

}

9) Добавим в реализацию базу данных postgresql

/// настроем и создадим БД. С несколькими таблицами

1. Создаём БД в postgress (делал через SQL Shell)

CREATE DATABASE DBSMS;

* Подключение к БД

\c dbsms

create table users(id bigint primary key, first\_name varchar(100), last\_name varchar(2000), created timestamp default current\_timestamp);

CREATE UNIQUE INDEX users\_id\_uindex ON users (id);

INSERT INTO users VALUES (1, 'Maksim', 'Sharafutdinov', current\_timestam);

INSERT INTO users VALUES (2, 'Andrei', 'Uharev', current\_timestamp);

INSERT INTO users VALUES (3, 'Kirill', 'Cleimenov', current\_timestamp);

create table sms(id bigint primary key, value varchar(80), creator\_id bigint references users(id), created timestamp default current\_timestamp);

CREATE UNIQUE INDEX sms\_id\_uindex ON sms (id);

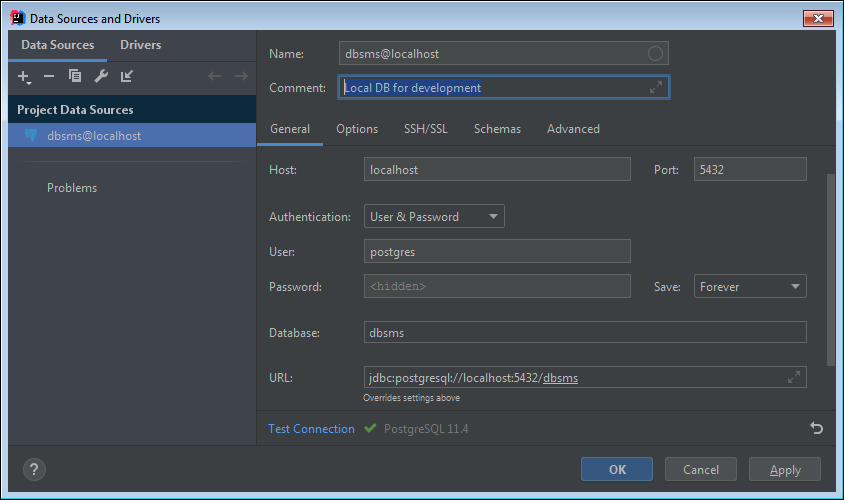
CREATE SEQUENCE sms\_id\_seq START WITH 4 INCREMENT BY 1;

INSERT INTO sms VALUES (1, 'Hello! How are you?', 1);

INSERT INTO sms VALUES (2, 'I’m fine!And you?', 2);

INSERT INTO sms VALUES (3, 'How much money earned M.Galkin', 3);

В интерфейсе IDEA добавляем БД. Справа.



См. статью

<https://javarush.ru/groups/posts/2579-dobavljaem-bd-k-restful-servisu-na-spring-boot-chastjh-1>

10) Добавим зависимости поддержку ORM jooq и БД postgresql

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-jooq</artifactId>  
 <version>2.2.6.RELEASE</version>  
</dependency>  
<dependency>  
 <groupId>org.postgresql</groupId>  
 <artifactId>postgresql</artifactId>  
 <version>42.2.10</version>  
</dependency>

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-jooq</artifactId>  
 <version>2.2.6.RELEASE</version>  
</dependency>

11) настраиваем application.properties для работы с БД

spring.datasource.url=jdbc:postgresql://localhost:5432/dbsms  
spring.datasource.username=postgres  
spring.datasource.password=postgres  
  
spring.datasource.driver-class-name=org.postgresql.Driver  
spring.jpa.database=postgresql  
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQL10Dialect

12)

**Приложение. С чём столкнулся**

ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath

Это приводит к тому, что не пишутся в консоль ошибки уровня WARN.

При возникновении 500 ошибки, не понятно было на каком сервисе падает.

Добавлять доп. зависимостей в pom.xml не нужно было. Нужно было просто обновить зависимости maven



Resolved [org.springframework.http.converter.HttpMessageNotWritableException: No converter for [class java.util.ArrayList] with preset Content-Type 'null']

Возникла 500 ошибка

Не хватало следующей зависимости

<dependency>  
 <groupId>com.fasterxml.jackson.dataformat</groupId>  
 <artifactId>jackson-dataformat-xml</artifactId>  
</dependency>

При этом также должны были быть реализованы getter и setter в классе сущности.

**Spring Boot** использует **JAXB** (Готовый в **JDK**) как библиотеку по умолчанию для конвертирования **XML** и **Java**. Но ваши классы Java должны быть аннотированы (annotated) с помощью **@XmlRootElement**,... Поэтому мой совет вам стоит использовать **jackson-dataformat-xml** как библиотеку для конвертирования**XML** и **Java**. Чтобы использовать **jackson-dataformat-xml** вам нужно объявить его в файле **pom.xml**

**No serializer found for class org.hibernate.proxy.pojo.bytebuddy.ByteBuddyInterceptor**

В endpoint где проходит поиск по id entity, которая собирается в коллекцию.

Вместо метода

*.getOne(id)*

следует использовать

.findById(smsId).get()

В таком случае и при запросе на несущесвующую запись entity будет приходить 500 ошибка.

При этом будет возникать предупреждение

'Optional.get()' without 'isPresent()' check

Если заменить на

.findById(smsId).orElse(null);

То предупреждение уйдёт. При запросе не существующей записи будет пустой ответ с кодом 200.

**При создании CriteriaBuilder возращает NullPointerException**

EntityManager должен содержать аннотацию @PersistenceContext.

@PersistenceContext  
EntityManager em;

Это позволяет соединиться с базой данных