#########

spring.io

#########

=>Novo Projeto:

restapi\src\main\java\br.com.aula.restapi

1).\RestapiApplication:

package br.com.aula.restapi;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestapiApplication {

   public static void main(String[] args) {

      SpringApplication.run(RestapiApplication.class, args);

   }

}

2).\controller

AlunoController:

package br.com.aula.restapi.controller;

import br.com.aula.restapi.repository.AlunoRepository;

import br.com.aula.restapi.request.RequestAluno;

import br.com.aula.restapi.service.AlunoService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.web.bind.annotation.\*;

import java.security.Provider;

import java.util.List;

@RestController

@RequestMapping ("aluno")

public class AlunoController {

    @PostMapping

    @ResponseStatus (HttpStatus.CREATED)

    public void cadastroaluno (@RequestBody RequestAluno aluno){

        System.out.println("Nome: " + aluno.getNome());

        System.out.println("CPF: " + aluno.getCPF());

        System.out.println("RA: " + aluno.getRA());

        service.save(aluno);

    }

    @Autowired

    private AlunoService service;

    @GetMapping

    public List<RequestAluno> consultaaluno (){

        return service.findall();

    }

}

3).\repository

AlunoRepository.java:

package br.com.aula.restapi.repository;

import br.com.aula.restapi.request.RequestAluno;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface AlunoRepository extends JpaRepository <RequestAluno, Integer>{

}

\request

RequestAluno.java:

package br.com.aula.restapi.request;

import javax.persistence.\*;

@Entity

@Table (name="alunos")

public class RequestAluno {

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private Integer id;

    @Column

    private String nome;

    @Column

    private String RA;

    @Column

    private String CPF;

    public String getNome() {

        return nome;

    }

    public void setNome(String nome) {

        this.nome = nome;

    }

    public String getRA() {

        return RA;

    }

    public void setRA(String RA) {

        this.RA = RA;

    }

    public String getCPF() {

        return CPF;

    }

    public void setCPF(String CPF) {

        this.CPF = CPF;

    }

}

4).\service

AlunoService:

package br.com.aula.restapi.service;

import br.com.aula.restapi.repository.AlunoRepository;

import br.com.aula.restapi.request.RequestAluno;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.stereotype.Service;

import javax.annotation.processing.Generated;

import java.util.List;

@Service

public class AlunoService {

    @Autowired

    private AlunoRepository repository;

    public void save(RequestAluno aluno){

        repository.save(aluno);

    }

    public List<RequestAluno> findall (){

        return repository.findAll();

    }

    }

5).\security

SecurityConfig:

package br.com.aula.restapi.security;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.oauth2.config.annotation.web.configuration.EnableAuthorizationServer;

import org.springframework.security.oauth2.config.annotation.web.configuration.EnableResourceServer;

@EnableWebSecurity

@EnableAuthorizationServer

@EnableResourceServer

public class SecurityConfig {

}

6) Arquivos de configuração do projeto:

A)restapi\.mvm

maven-wrapper.properties:

distributionUrl=https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.8.4/apache-maven-3.8.4-bin.zip

wrapperUrl=https://repo.maven.apache.org/maven2/org/apache/maven/wrapper/maven-wrapper/3.1.0/maven-wrapper-3.1.0.jar

restapi

B)application.properties:

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

security.oauth2.client.scope=password

C)pom.xml:

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.6.7</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>br.com.aula</groupId>

<artifactId>restapi</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>restapi</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- <dependency>-->

<!-- <groupId>org.springframework.boot</groupId>-->

<!-- <artifactId>spring-boot-starter-security</artifactId>-->

<!-- </dependency>-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.security.oauth.boot</groupId>

<artifactId>spring-security-oauth2-autoconfigure</artifactId>

<version>2.0.1.RELEASE</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- <dependency>-->

<!-- <groupId>org.springframework.security</groupId>-->

<!-- <artifactId>spring-security-test</artifactId>-->

<!-- <scope>test</scope>-->

<!-- </dependency>-->

<!-- https://mvnrepository.com/artifact/com.h2database/h2 -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<version>1.4.199</version>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

7)POSTMAN: criar collection

A)New collectoin (ex: TesteCadastroAluno)

B)Add Request -> POST=localhost:8080/aluno

Body tab (raw) -> JSON type (rodar depois do item "F"):

{

    "nome": "Maria",

    "cpf": "55388811166",

    "ra": "23312"

}

C)Add Request -> GET=localhost:8080/aluno

Body tab (none)

Response Body Example (rodar depois do item "F"):

[

    {

        "nome": "Pedro",

        "ra": "12312",

        "cpf": "12388811166"

    },

    {

        "nome": "Joao",

        "ra": "33312",

        "cpf": "33388811166"

    },

    {

        "nome": "Maria",

        "ra": "23312",

        "cpf": "55388811166"

    }

]

D)Add Request -> POST=localhost:8080/oauth/token

Body tab => Add Keys:

-grant\_type:password

-username:user

-password:"log no console do IntelliJ => Using generated security password:512833a6-..."

Authorization tab:

Username:"log no console do IntelliJ => OAuth2 Client.. security.oauth2.client.client-id = b759d5ee-..."

Password:"log no console do IntelliJ => OAuth2 Client.. security.oauth2.client.client-secret = 97f79289-2..."

E)Rodar o item "D", copiar o token ("access\_token"), voltar no item "B" na tab Authorization e escolher Type "Bearer Token" e colar o valor do Token no campo "Token".

F)Repetir o item "E" para adicionar o "Token" no item "C".

G)Rodar o item "B" (algumas vezes com valores diferentes no Body da requisição POST).

H)Rodar o item "C" e verificar a lista retornada no Response Body com os dados gravados por persistência em Banco de Dados (h2).