**Projeto API Back End para Módulo de Temperatura**

**Introdução**

Este projeto é composto pela construção de um backend REST simples, para monitoramento de temperatura.

**Tecnologia**

O teste deve ser desenvolvido em ​**Spring Boot 2.4.2**​, em projeto padrão ​**Maven**​.

Sugestão: criar utilizando ​[**https://start.spring.io/**](https://start.spring.io/)

**Modelo de dados**

Persistidos:

Temperatura (id, valor)

Token (id, token, login, expiracao (timestamp), administrador (boolean) )

Usuario (id, login, senha, nome, administrador (boolean))

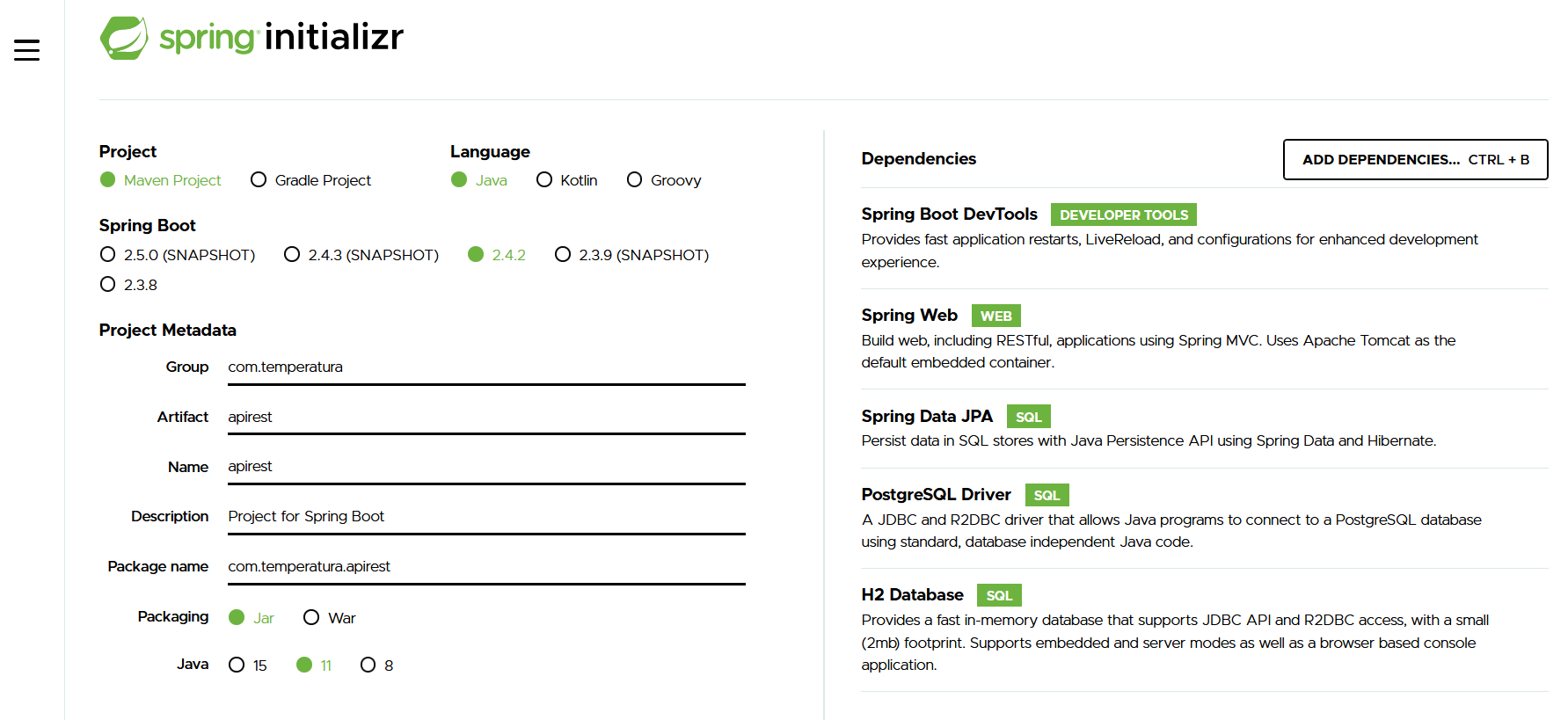
1. **Para Windows 10 64 bits, baixar:**

* eclipse-inst-jre-win64
* pgadmin4-4.29-x64
* postgresql-13.1-1-windows-x64

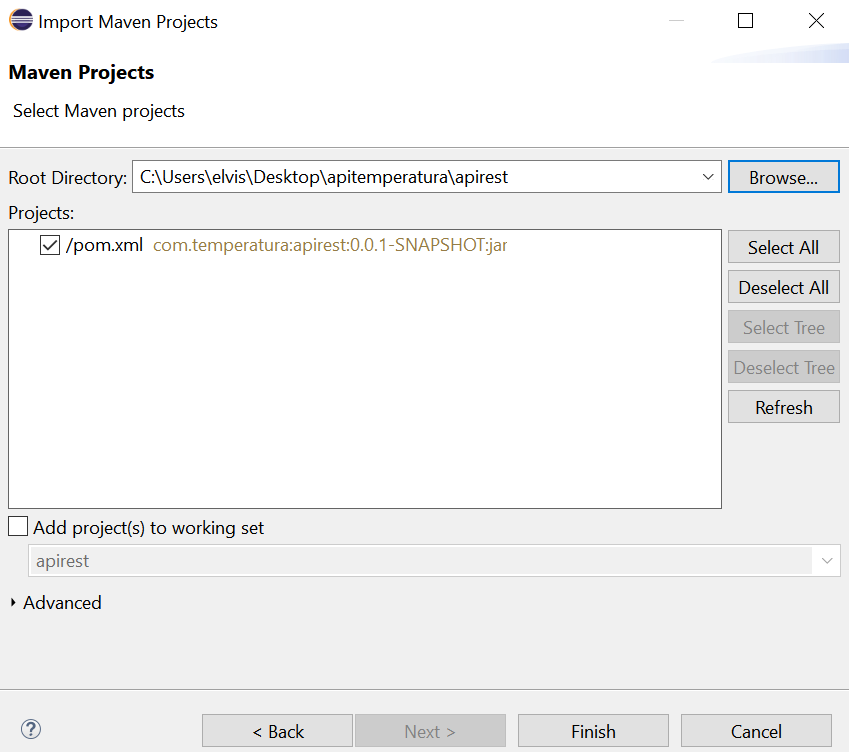
1. **Criar Projeto**

**Acessar:**

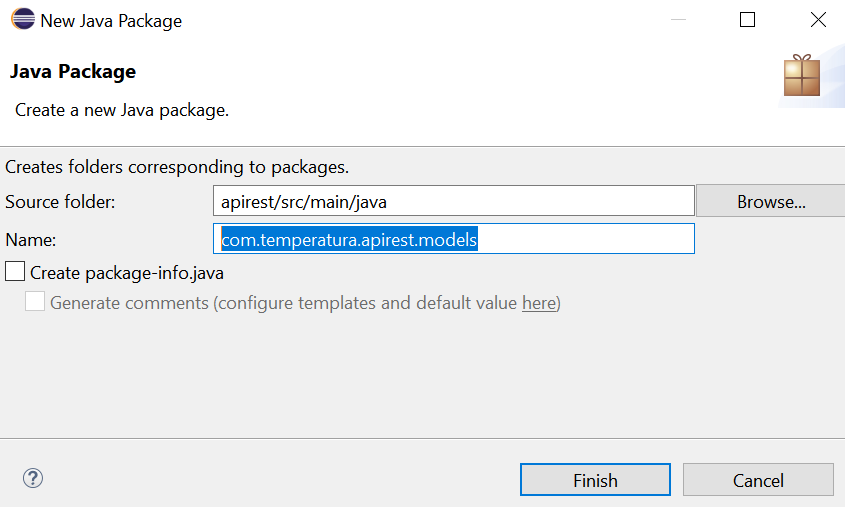
* [**https://start.spring.io/**](https://start.spring.io/)



1. **Importar projeto Maven**



1. **Criar pacote Models**



1. **Criar classe Temperatura.java dentro do pacote.models**

|  |
| --- |
| **package** com.temperatura.apirest.models;  **import** java.io.Serializable;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  @Entity  @Table(name = "TB\_TEMPERATURA")  **public** **class** Temperatura **implements** Serializable {  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  @Id  @GeneratedValue(strategy = GenerationType.***AUTO***)  **private** **long** idDispositivo;  **private** String valorSensor;  **public** **long** getIdDispositivo() {  **return** idDispositivo;  }  **public** **void** setIdDispositivo(**long** idDispositivo) {  **this**.idDispositivo = idDispositivo;  }  **public** String getValorSensor() {  **return** valorSensor;  }  **public** **void** setValorSensor(String valorSensor) {  **this**.valorSensor = valorSensor;  }  } |

1. **Configurar o banco de dados no arquivo application.properties**

|  |
| --- |
| spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation=true  #Banco local - Elvis  spring.datasource.url= jdbc:postgresql://localhost:5432/temperatura\_apirest  spring.datasource.username=postgres  spring.datasource.password=banco123  spring.jpa.hibernate.ddl-auto=update |

Em que:

* temperatura\_apirest é nome do banco de dados

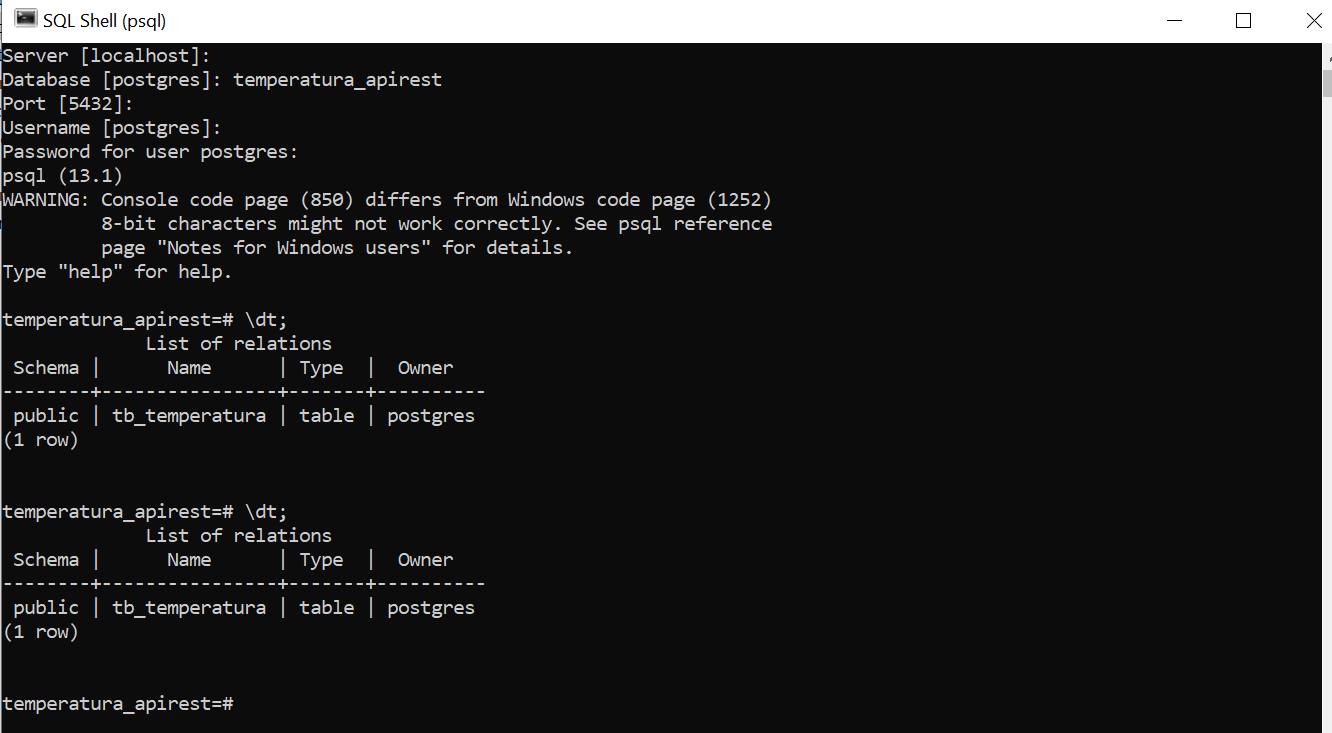
1. **Criar o banco local**

* SQL Shell (psql)
* create database temperatura\_apirest;
* acessar o banco de dados temperatura\_apirest

1. **Testando a conexão da aplicação com o banco de dados**

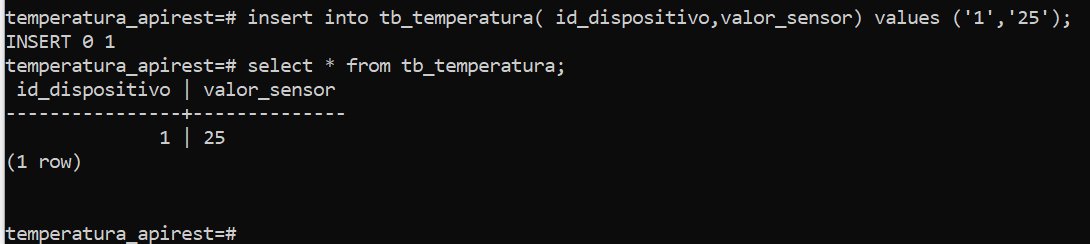
* Botão direito em ApiRestApplication.java
* Run as > 1 Java Application





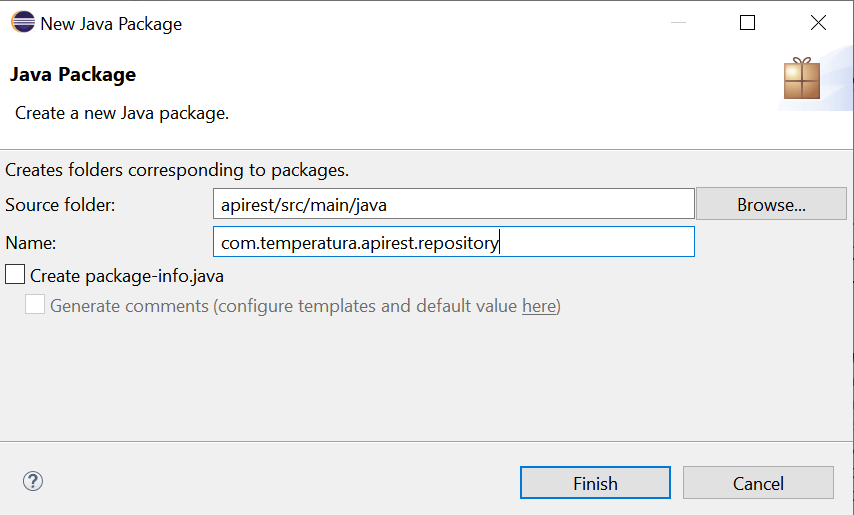
1. **Adicionando valores ao banco de dados**

* insert into tb\_temperatura( id\_dispositivo,valor\_sensor) values ('1','25');
* select \* from tb\_temperatura;

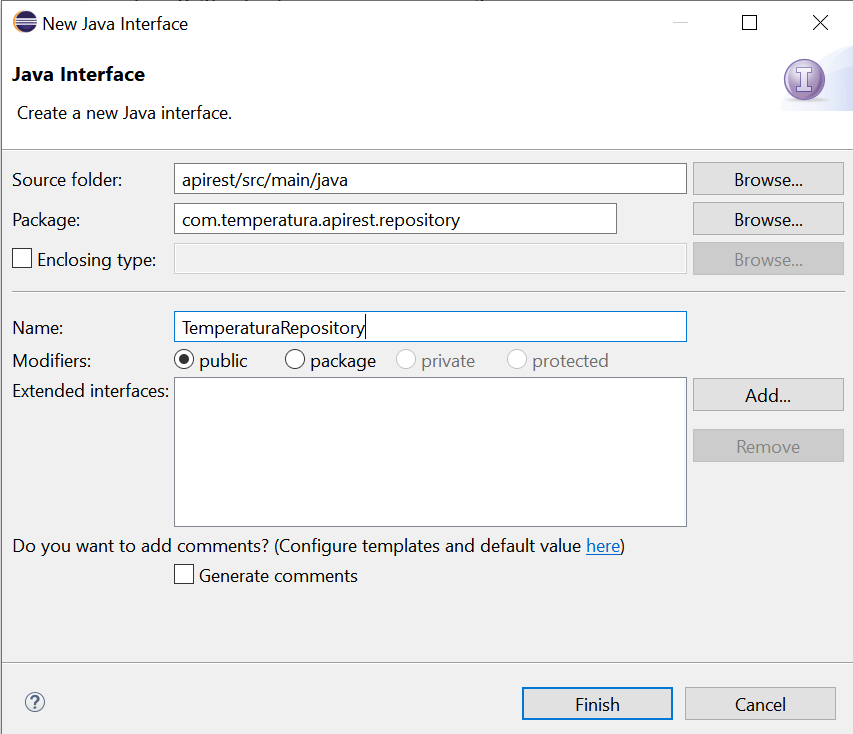


1. Criar Repositório para a classe Temperatura.java

* Criar pacote repository



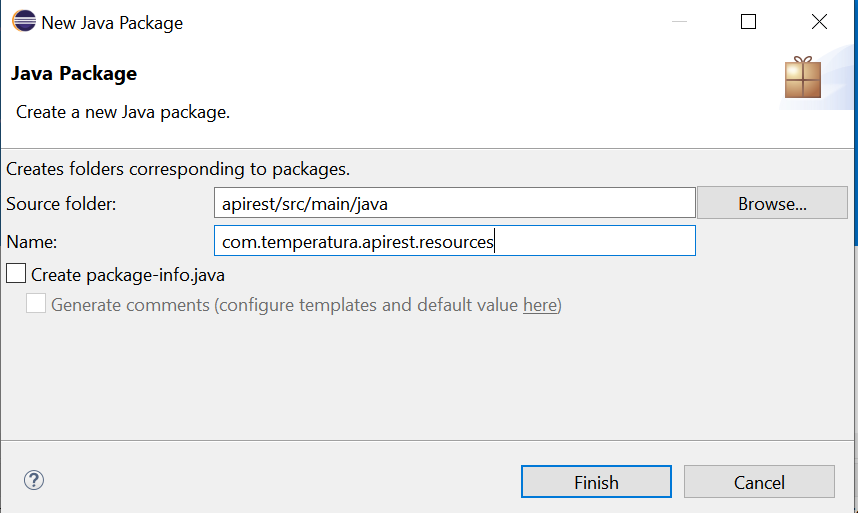
* Criar interface TemperaturaRepository



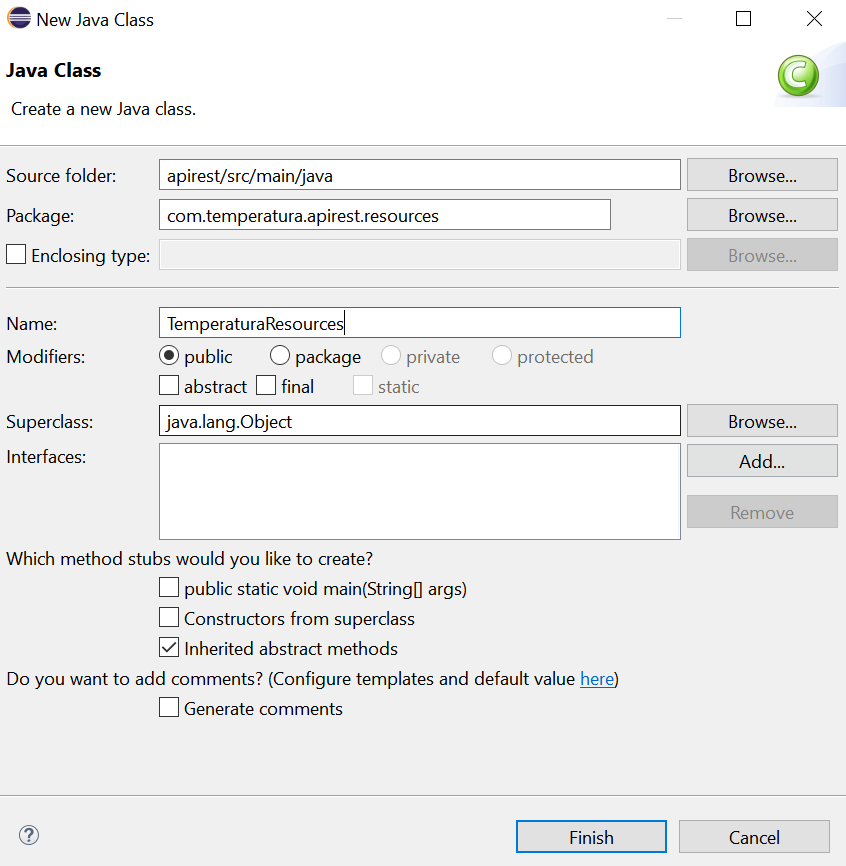
|  |
| --- |
| **package** com.temperatura.apirest.repository;  **import** org.springframework.data.jpa.repository.JpaRepository;  **import** com.temperatura.apirest.models.Temperatura;  /\*Usar o JpaRepository para fazer métodos\*/  /\*Criar instância Temperatura\*/  **public** **interface** TemperaturaRepository **extends** JpaRepository<Temperatura, Long> {  } |

1. Criar Métodos com persistência para banco de dados para a classe Temperatura.java

* Criar pacote resources



* Criar classe TemperaturaResource



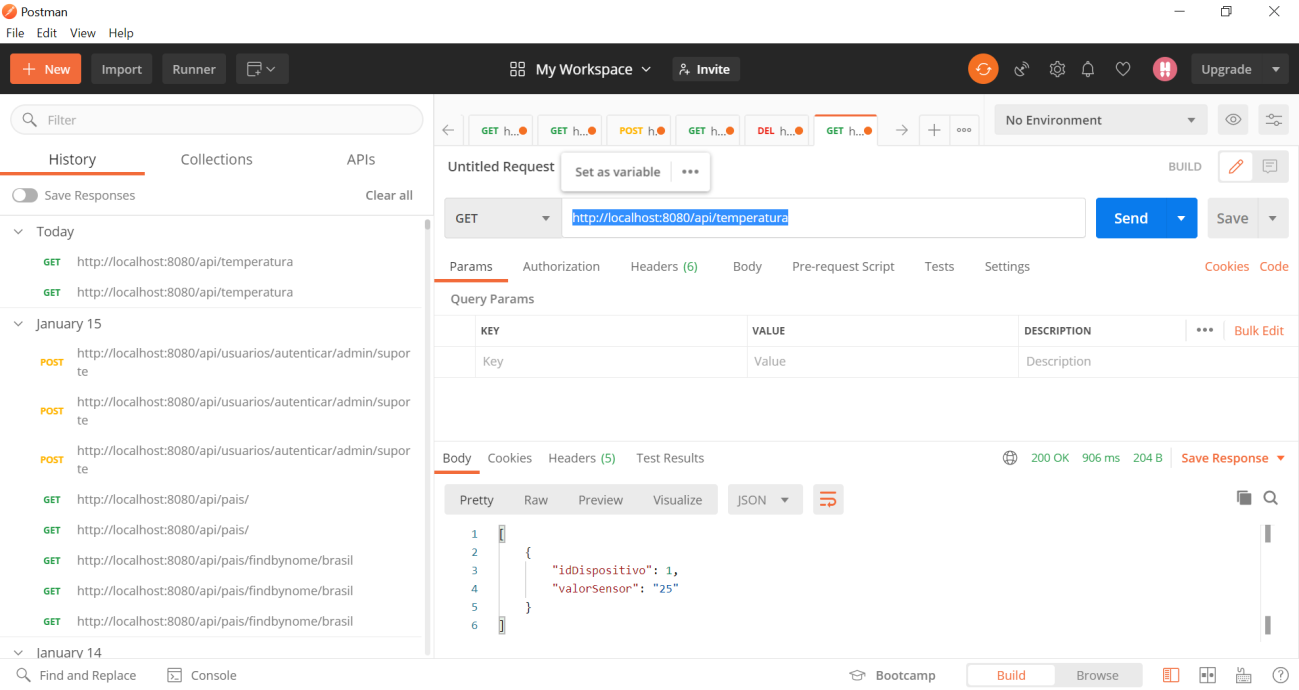
* Fazer os imports
* @RestController
* @RequestMapping
* @Autowired
* @GetMapping
* **import** java.util.List;
* **import** org.springframework.beans.factory.annotation.Autowired;
* **import** org.springframework.web.bind.annotation.GetMapping;
* **import** org.springframework.web.bind.annotation.RequestMapping;
* **import** org.springframework.web.bind.annotation.RestController;
* **import** com.temperatura.apirest.models.Temperatura;
* **import** com.temperatura.apirest.repository.TemperaturaRepository;

|  |
| --- |
| **package** com.temperatura.apirest.resources;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.temperatura.apirest.models.Temperatura;  **import** com.temperatura.apirest.repository.TemperaturaRepository;  @RestController  @RequestMapping(value = "/api")  **public** **class** TemperaturaResources {  @Autowired  TemperaturaRepository temperaturaRepository;  @GetMapping("/temperatura")  **public** List <Temperatura> listaTemperaturas() {  **return** temperaturaRepository.findAll();  }  } |

1. **Testando a aplicação utilizando Postman**

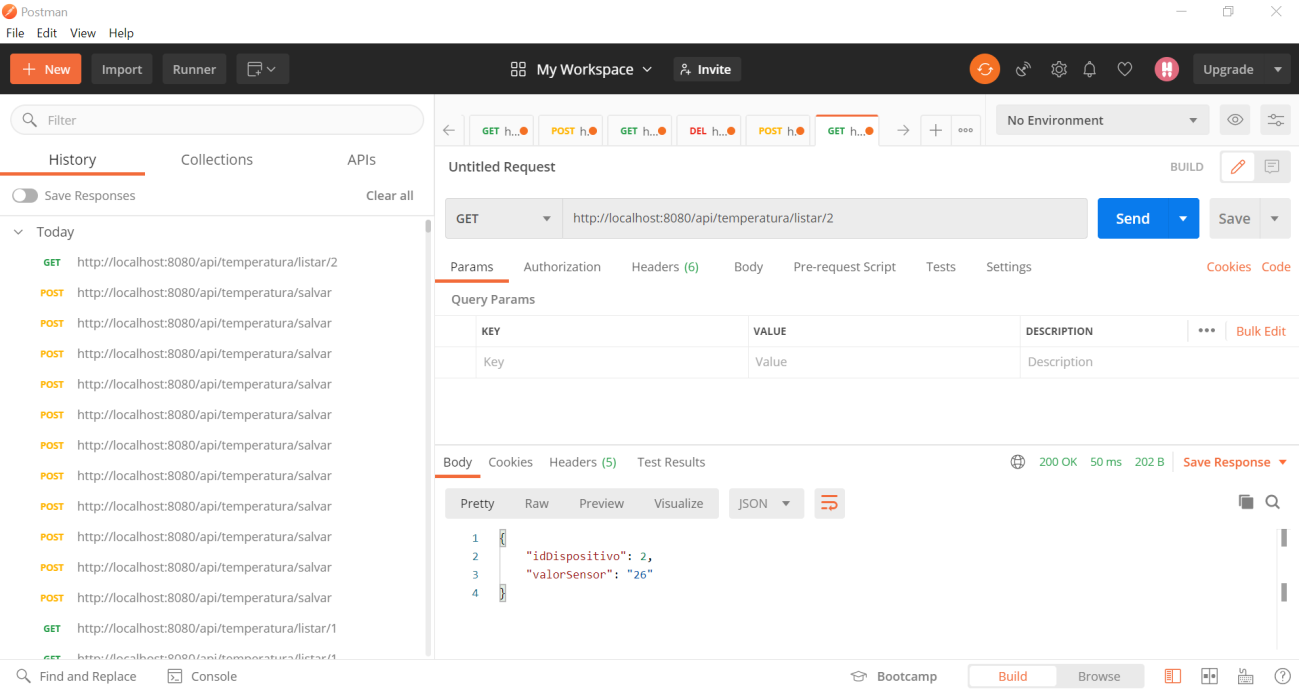
Para listar as temperaturas utilizando o método GET

* <http://localhost:8080/api/temperatura>



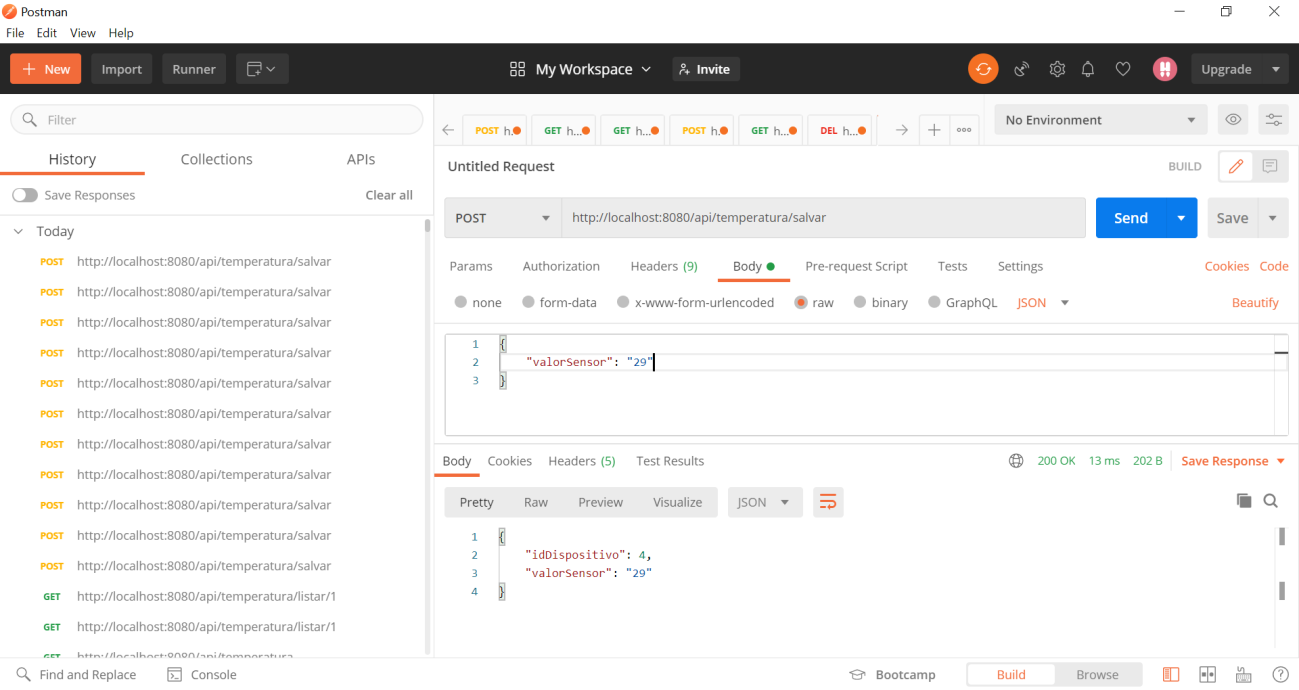
Para listar temperatura específica pelo ID utilizando o método GET

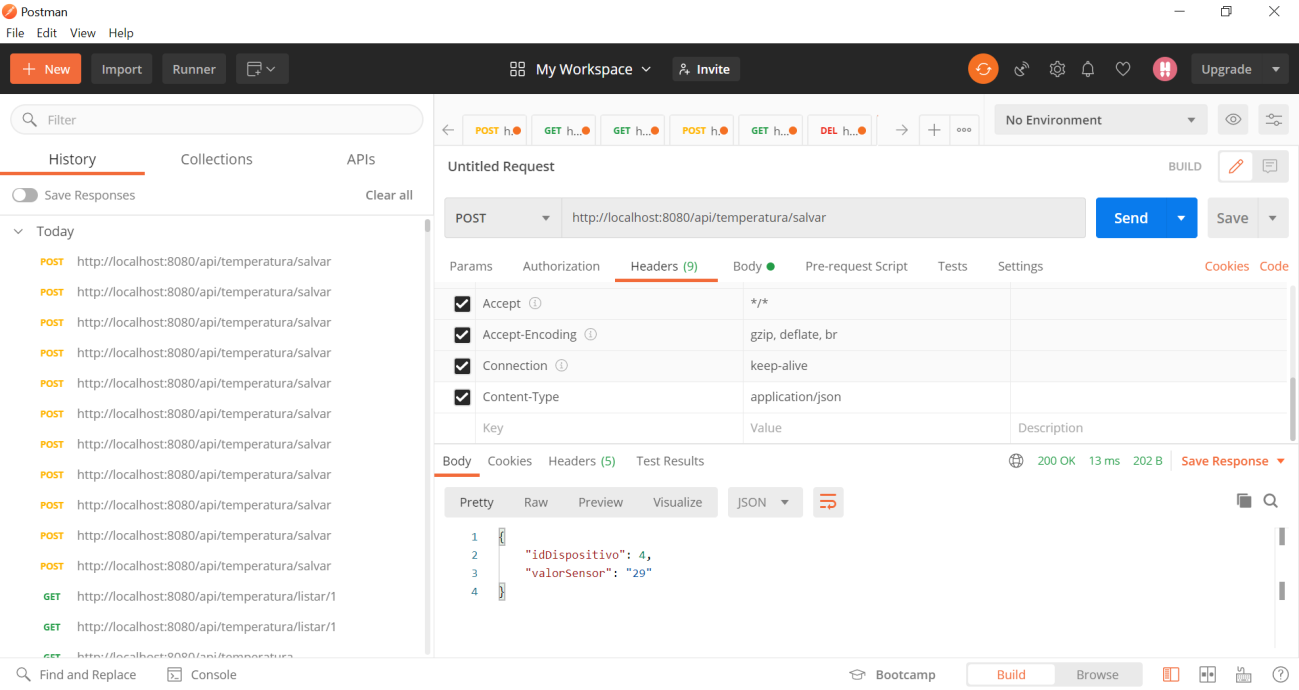
* <http://localhost:8080/api/temperatura/listar/2>



Para adicionar uma temperatura ao banco de dados utilizando o método POST:

* http://localhost:8080/api/temperatura/salvar





Até aqui código:

TemperaturaResources.java

|  |
| --- |
| package com.temperatura.apirest.resources;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;  import com.temperatura.apirest.models.Temperatura;  import com.temperatura.apirest.repository.TemperaturaRepository;  @RestController  @RequestMapping(value = "/api")  public class TemperaturaResources {  @Autowired  TemperaturaRepository temperaturaRepository;  @GetMapping("/temperatura/listar")  public List<Temperatura> listaTemperaturas() {  return temperaturaRepository.findAll();  }  @GetMapping("/temperatura/listar/{id}")  public Temperatura listaTemperaturaUnica(@PathVariable(value = "id") long id) {  return temperaturaRepository.findById(id);  }  @PostMapping("/temperatura/salvar")  public Temperatura salvaProduto(@RequestBody Temperatura temperatura) {  return temperaturaRepository.save(temperatura);  }  } |

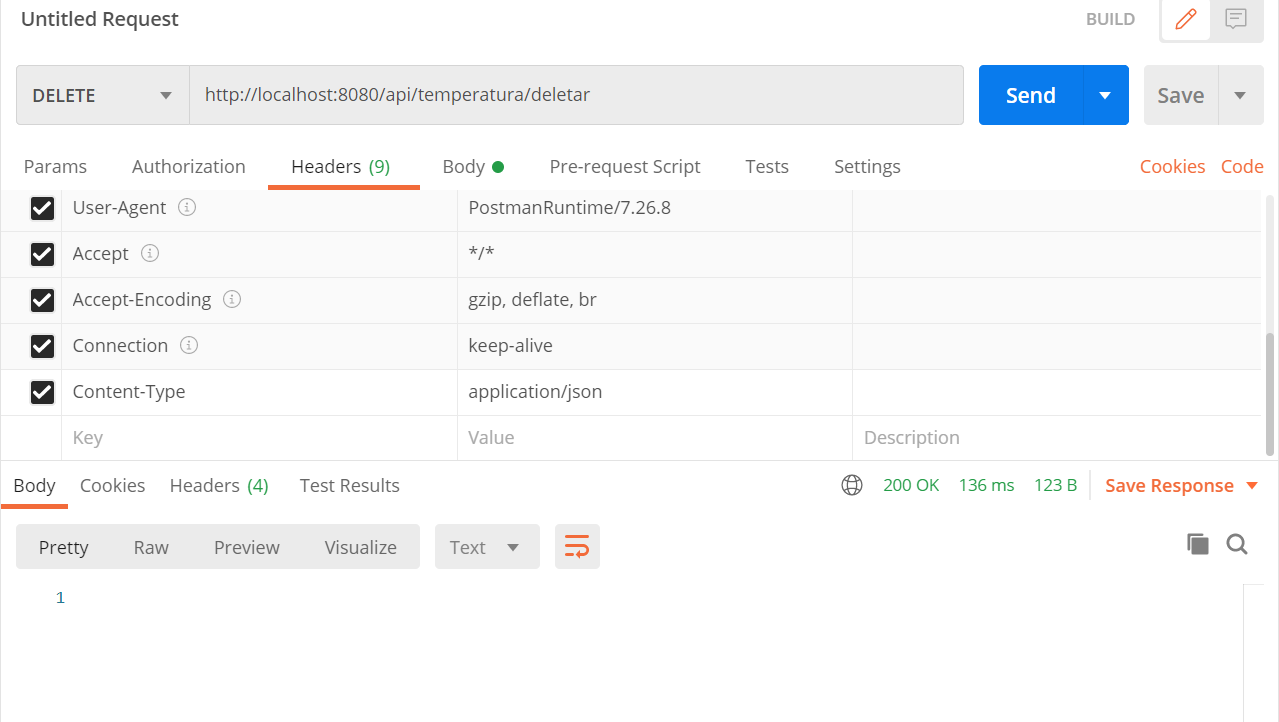
TemperaturaRepository.java

|  |
| --- |
| package com.temperatura.apirest.repository;  import org.springframework.data.jpa.repository.JpaRepository;  import com.temperatura.apirest.models.Temperatura;  /\*Usar o JpaRepository para fazer métodos\*/  /\*Criar instância Temperatura\*/  public interface TemperaturaRepository extends JpaRepository<Temperatura, Long> {  Temperatura findById(long id);  } |

Para apagar uma temperatura ao banco de dados utilizando o método DELETE:

* <http://localhost:8080/api/temperatura/deletar>

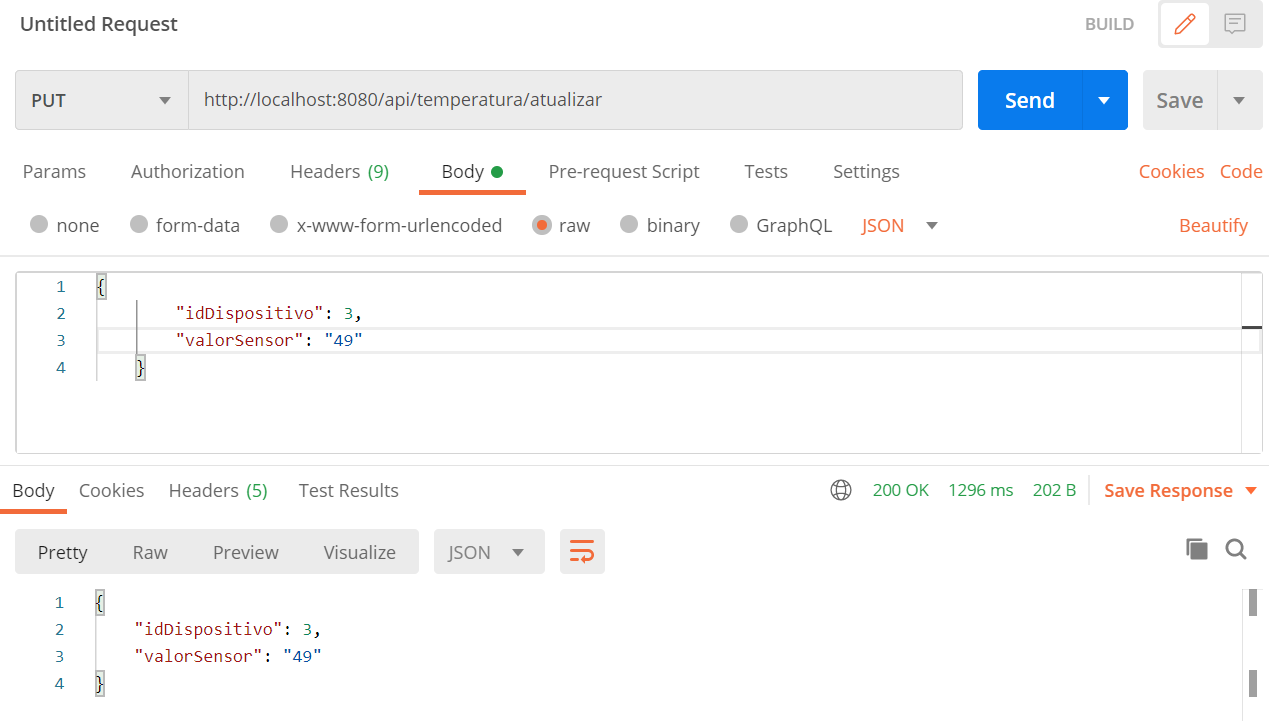
|  |
| --- |
| no raw:  {          "idDispositivo": 4,          "valorSensor": "29"  }  No header:  Content-type:  Application/json  @DeleteMapping("/temperatura/deletar")  **public** **void** deletaTemperatura(@RequestBody Temperatura temperatura) {  temperaturaRepository.delete(temperatura);  } |



Para atualizar uma temperatura ao banco de dados utilizando o método PUT:

* <http://localhost:8080/api/temperatura/atualizar>

|  |
| --- |
| no raw:  {          "idDispositivo": 3,          "valorSensor": "49"  }  No header:  Content-type:  Application/json  @PutMapping("/temperatura/atualizar")  **public** Temperatura atualizaTemperatura(@RequestBody Temperatura temperatura) {  **return** temperaturaRepository.save(temperatura);  } |



Código até aqui:

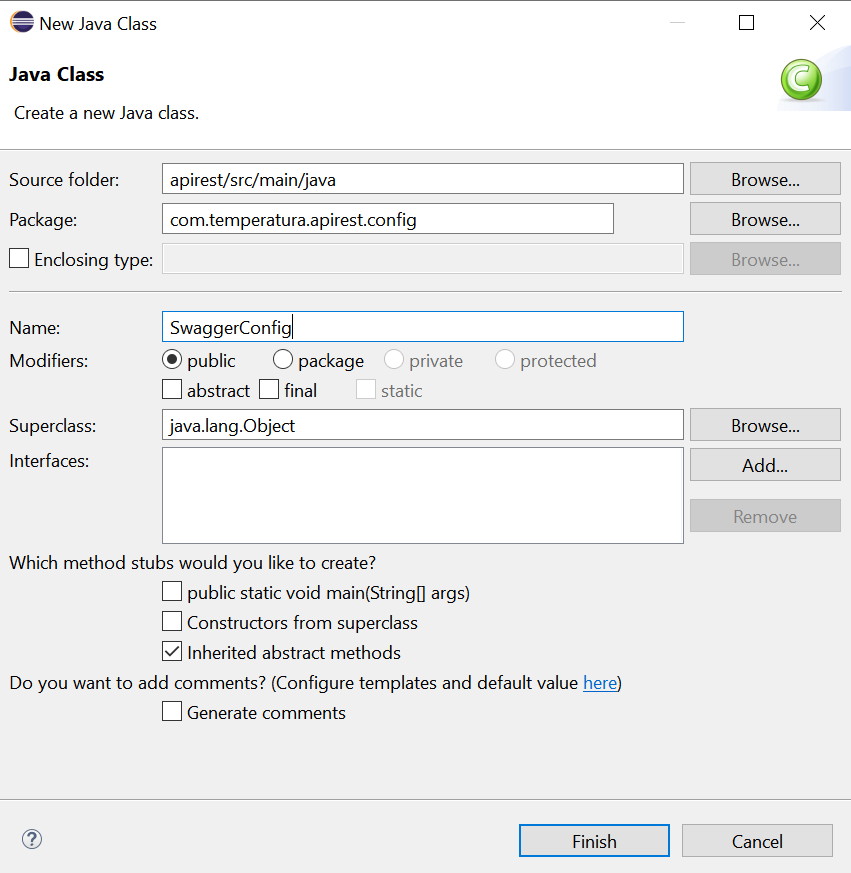
|  |
| --- |
| package com.temperatura.apirest.resources;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.DeleteMapping;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.PutMapping;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;  import com.temperatura.apirest.models.Temperatura;  import com.temperatura.apirest.repository.TemperaturaRepository;  @RestController  @RequestMapping(value = "/api")  public class TemperaturaResources {  @Autowired  TemperaturaRepository temperaturaRepository;  @GetMapping("/temperatura/listar")  public List<Temperatura> listaTemperaturas() {  return temperaturaRepository.findAll();  }  @GetMapping("/temperatura/listar/{id}")  public Temperatura listaTemperaturaUnica(@PathVariable(value = "id") long id) {  return temperaturaRepository.findById(id);  }  @PostMapping("/temperatura/salvar")  public Temperatura salvaProduto(@RequestBody Temperatura temperatura) {  return temperaturaRepository.save(temperatura);  }  @DeleteMapping("/temperatura/deletar")  public void deletaTemperatura(@RequestBody Temperatura temperatura) {  temperaturaRepository.delete(temperatura);  }  @PutMapping("/temperatura/atualizar")  public Temperatura atualizaTemperatura(@RequestBody Temperatura temperatura) {  return temperaturaRepository.save(temperatura);  }  } |

1. API REST com Spring Boot – Swagger

* Adicionar as dependências do swagger no arquivo pom.xml

|  |
| --- |
| <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger2</artifactId>  <version>2.7.0</version>  </dependency>  <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger-ui</artifactId>  <version>2.7.0</version>  </dependency> |

* Criar novo pacote .config
* Criar nova classe SwaggerConfig



Código classe SwaggerConfig

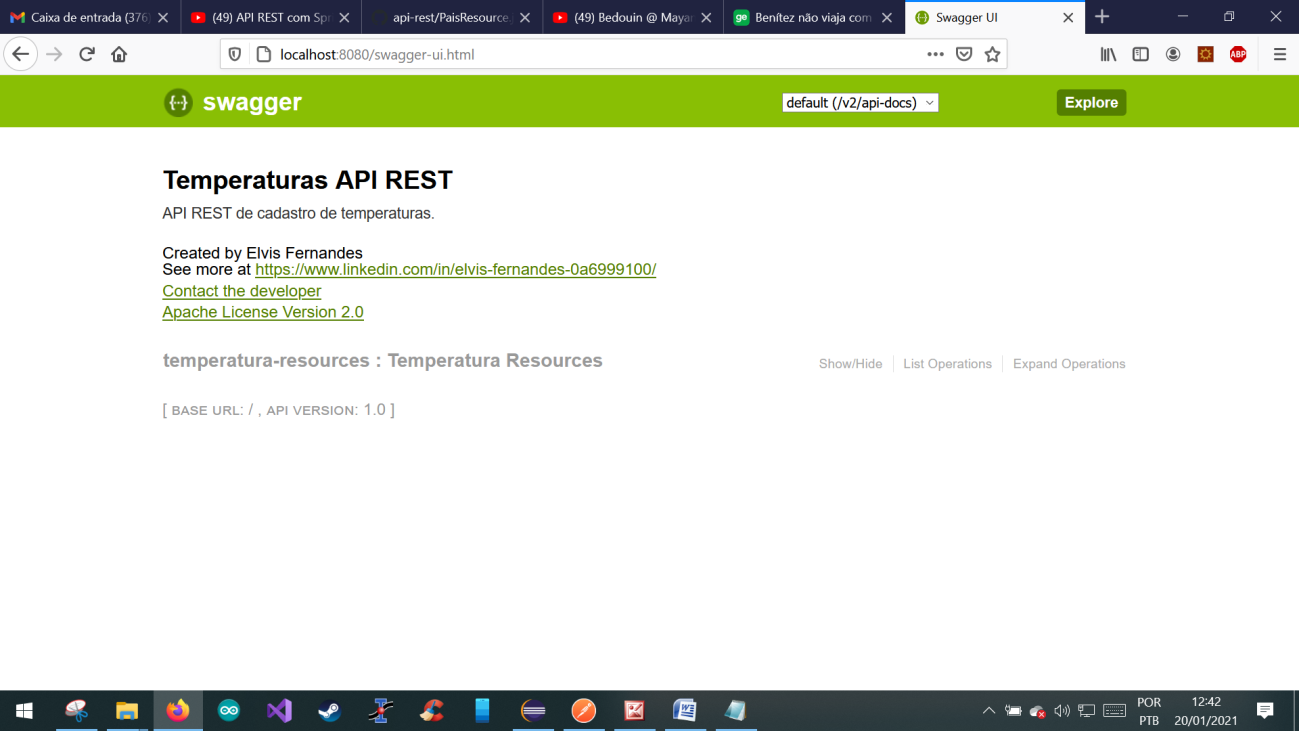
|  |
| --- |
| **package** com.temperatura.apirest.config;  **import** springfox.documentation.swagger2.annotations.EnableSwagger2;  **import** org.springframework.context.annotation.Bean;  **import** org.springframework.context.annotation.Configuration;  **import** springfox.documentation.builders.RequestHandlerSelectors;  **import** springfox.documentation.service.ApiInfo;  **import** springfox.documentation.service.Contact;  **import** springfox.documentation.service.VendorExtension;  **import** springfox.documentation.spi.DocumentationType;  **import** springfox.documentation.spring.web.plugins.Docket;  **import** **static** springfox.documentation.builders.PathSelectors.*regex*;  **import** java.util.ArrayList;  @Configuration  @EnableSwagger2  **public** **class** SwaggerConfig {    @Bean  **public** Docket temperaturaApi() {  **return** **new** Docket(DocumentationType.***SWAGGER\_2***)  .select()  .apis(RequestHandlerSelectors.*basePackage*("com.temperatura.apirest"))  .paths(*regex*("/api.\*"))  .build()  .apiInfo(metaInfo());  }  **private** ApiInfo metaInfo() {  ApiInfo apiInfo = **new** ApiInfo(  "Temperaturas API REST",  "API REST de cadastro de temperaturas.",  "1.0",  "Terms of Service",  **new** Contact("Elvis Fernandes", "https://www.linkedin.com/in/elvis-fernandes-0a6999100/",  "elvis.fernandes89@gmail.com"),  "Apache License Version 2.0",  "https://www.apache.org/licesen.html", **new** ArrayList<VendorExtension>()  );  **return** apiInfo;  }  } |

Adicionar a API dentro de TemperaturaResources:

|  |
| --- |
| package com.temperatura.apirest.resources;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.CrossOrigin;  import org.springframework.web.bind.annotation.DeleteMapping;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.PutMapping;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;  import com.temperatura.apirest.models.Temperatura;  import com.temperatura.apirest.repository.TemperaturaRepository;  import io.swagger.annotations.Api;  import io.swagger.annotations.ApiOperation;  @RestController  @RequestMapping(value = "/api")  @Api(value = "API REST Pais")  @CrossOrigin(origins = "\*")  public class TemperaturaResources {  @Autowired  TemperaturaRepository temperaturaRepository;  @GetMapping("/temperatura/listar")  @ApiOperation(value = "Retorna uma lista de temperaturas")  public List<Temperatura> listaTemperaturas() {  return temperaturaRepository.findAll();  }  @GetMapping("/temperatura/listar/{id}")  @ApiOperation(value = "Retorna uma temperatura a partir do ID")  public Temperatura listaTemperaturaUnica(@PathVariable(value = "id") long id) {  return temperaturaRepository.findById(id);  }  @PostMapping("/temperatura/salvar")  @ApiOperation(value = "Salva uma temperatura")  public Temperatura salvaProduto(@RequestBody Temperatura temperatura) {  return temperaturaRepository.save(temperatura);  }  @DeleteMapping("/temperatura/deletar")  @ApiOperation(value = "Deleta uma temperatura")  public void deletaTemperatura(@RequestBody Temperatura temperatura) {  temperaturaRepository.delete(temperatura);  }  @PutMapping("/temperatura/atualizar")  @ApiOperation(value = "Atualiza uma temperatura")  public Temperatura atualizaTemperatura(@RequestBody Temperatura temperatura) {  return temperaturaRepository.save(temperatura);  }  } |

Acessar a API:

* <http://localhost:8080/swagger-ui.html>



1. **API REST com Spring Boot - Deploy no Heroku**

* Instalar o Heroku Toolbelt
* Fazer conta no Heroku Toolbelt
* Entrar via cmd no seguinte diretório:

C:\Users\elvis\Desktop\apitemperatura>

* [elvis.fernandes89@gmail.com](mailto:elvis.fernandes89@gmail.com)
* xza784
* criar um new app no Heroku

