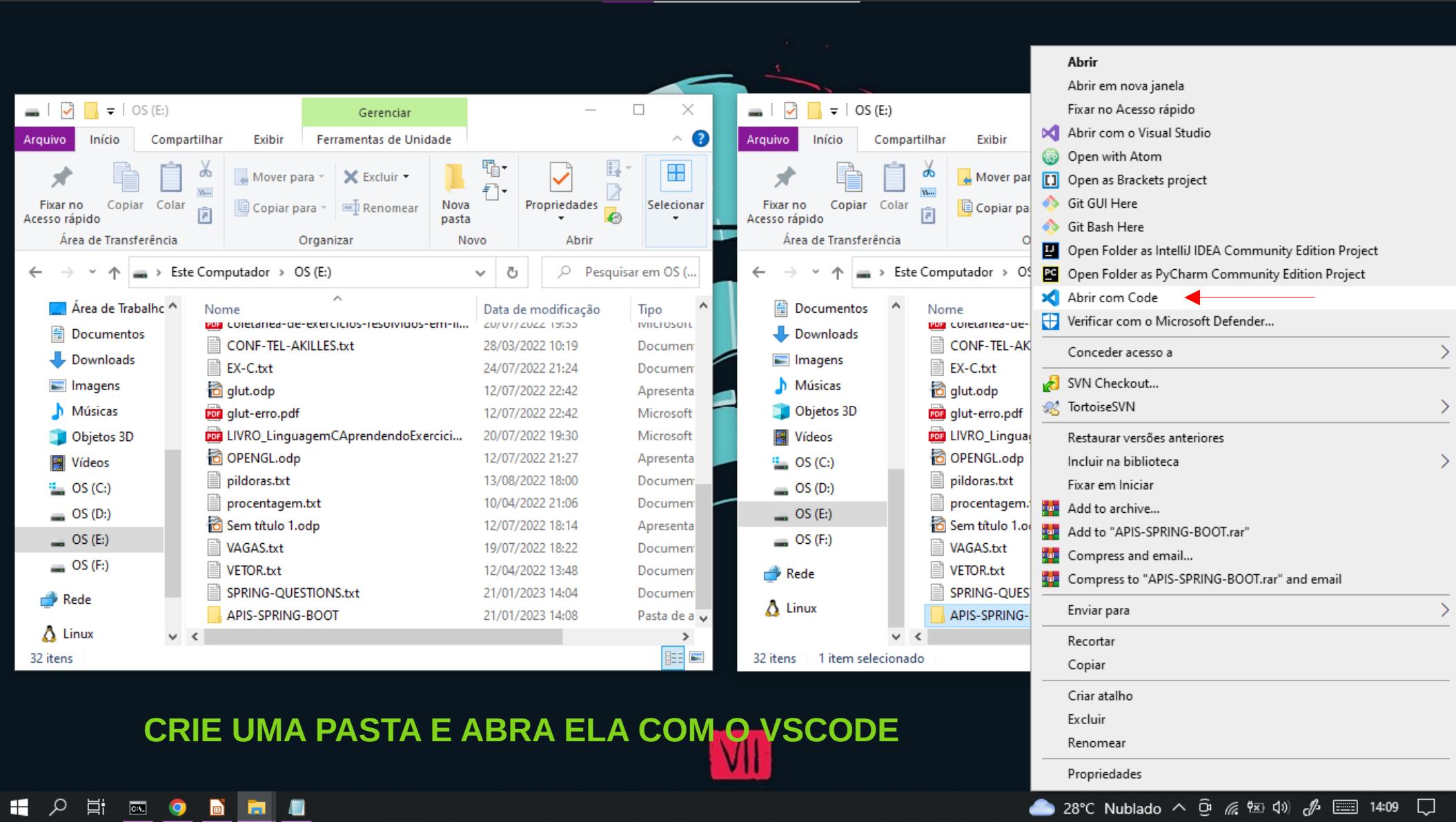


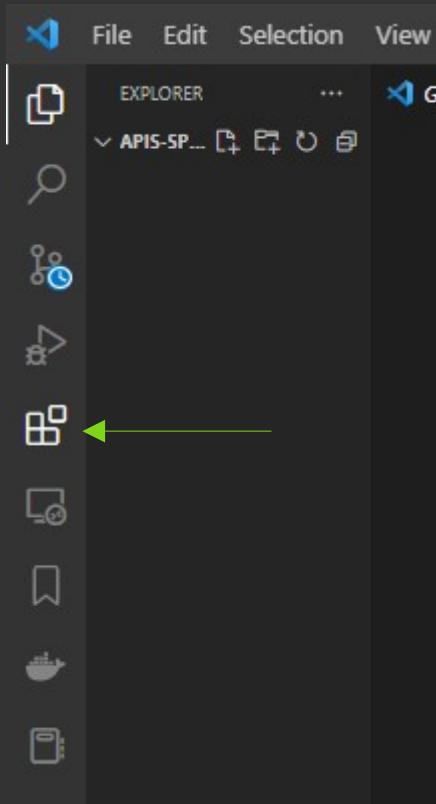
**JAVA SPRING BOOT  
2023  
NANO**

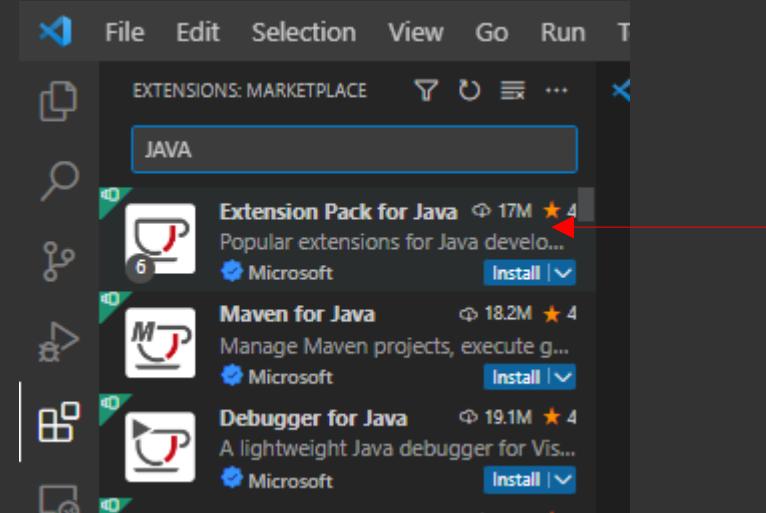
**Com a benção de DEUS iniciamos os estudos**

## **CONFIGURANDO O VISUAL ESTUDIO CODE**



# INSTALANDO AS EXTENSÕES NECESSÁRIAS







File Edit Selection View Go Run

Terminal Help

Extension: Extension Pack for Java - APIS-SPRING-BOOT - Visual Studio Code



EXTENSIONS: MARKETPLACE

JAVA

- Extension Pack for Java** ⚡ 17M ★ 4  
Popular extensions for Java development  
Microsoft [Install](#) | [View details](#)
- Maven for Java** ⚡ 18.2M ★ 4  
Manage Maven projects, execute goals...  
Microsoft [Install](#) | [View details](#)
- Debugger for Java** ⚡ 19.1M ★ 4  
A lightweight Java debugger for Visual Studio Code  
Microsoft [Install](#) | [View details](#)
- Project Manager for Java** ⚡ 17M ★ 4  
Manage Java projects in Visual Studio Code  
Microsoft [Install](#) | [View details](#)
- Test Runner for Java** ⚡ 17.5M ★ 4  
Run and debug JUnit or TestNG tests in VS Code  
Microsoft [Install](#) | [View details](#)
- Language Support for Java** ⚡ 21.2M ★ 3.5  
Java Linting, Intellisense, formatting, refactoring, and more  
Red Hat [Install](#) | [View details](#)
- Spring Initializr Java Extension** ⚡ 2M ★ 3.5  
A lightweight extension based on Spring Initializr  
Microsoft [Install](#) | [View details](#)
- Java Language Support** ⚡ 1.4M ★ 3  
Java support using the Java Compiler API  
George Fraser [Install](#)
- Java Debugger** ⚡ 724K ★ 3.5  
Java Debugger (break points, variable inspection, step into/step over)

Extension: Extension Pack for Java



## Extension Pack for Java v0.25.7 Preview

Microsoft | ⚡ 17,062,063 | ★★★★☆ (54)

Popular extensions for Java development that provides Java IntelliSense, debugging, testing, Maven/Gradle support, project management, and more.

[Install](#) | [View details](#)

[Details](#) [Feature Contributions](#) [Changelog](#)

### Extension Pack (6)



#### IntelliCode

AI-assisted development  
Microsoft



#### Language Support for Java(TM) by Red Hat

Java Linting, Intellisense, formatting, refactoring, and more  
Red Hat



#### Debugger for Java

A lightweight Java debugger for Visual Studio Code  
Microsoft



#### Maven for Java

Manage Maven projects, execute goals, generate reports, and more  
Microsoft

## Extension Pack for Java

Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out [Java in VS Code](#) to get started.

### Categories

[Programming Languages](#) [Snippets](#)

[Linters](#) [Debuggers](#) [Formatters](#)

[Extension Packs](#)

### Extension Resources

[Marketplace](#)

[Repository](#)

[License](#)

[Microsoft](#)

### More Info

Published [9/27/2017, 06:30:00](#)

Last released [11/29/2022, 02:20:00](#)

File Edit Selection View Go Run Terminal Help

Install New JDK - APIS-SPRING-BOOT - Visual Studio Code

EXTENSIONS: MARKETPLACE JAVA

Install New JDK × Get Started ×

## Install New JDK

Adoptium's Temurin Others

Version

- 8 (LTS)
- 11 (LTS)
- 17 (LTS)

JVM

- hotspot

Download windows-x64  
jdk-17.0.6+10 | 180.05 MB

After you finish JDK installation, please reload Visual Studio Code to make it effective.

Reload Window Having trouble?

Get Started with Java Development

Your first steps to set up powerful Java tools in a lightweight, performant editor!

Get your runtime ready

The Extension Pack for Java requires at least one Java runtime to be installed.

Install JDK

Explore your project

Launch, debug and test

Extensions for additional tools and frameworks

Se for preciso instale o jdk eu já possuia a versão 17  
No cmd javac -version  
Java -version  
Pra saber qual versão

Go Live

File Edit Selection View Go Run Terminal Help

EXTENSIONS: MARKPLACE JAVA

Install New JDK × Get Started ×

## Install New JDK

Adoptium's Temurin Others

Version

- 8 (LTS)
- 11 (LTS)
- 17 (LTS)

JVM

- hotspot

Download windows-x64  
jdk-17.0.6+10 | 180.05 MB

After you finish JDK installation, please reload Visual Studio Code to make it effective.

Reload Window Having trouble?

Get Started with Java Development

Your first steps to set up powerful Java tools in a lightweight, performant editor!

Get your runtime ready

The Extension Pack for Java requires at least one Java runtime to be installed.

Install JDK

Explore your project

Launch, debug and test

Extensions for additional tools and frameworks

Se for preciso instale o jdk eu já possuia a versão 17  
No cmd javac -version  
Java -version  
Pra saber qual versão

Go Live

EXTENSIONS: MARKETPLACE

spring

- Spring Initializr Java Support** ⚡ 2M ★ 3.5  
A lightweight extension based on Spring Initializ...  
Pivotal [Install](#)
- Spring Boot Tools** ⚡ 1.8M ★ 5  
Provides validation and content assist for Spring ...  
Pivotal [Install](#)
- Spring Boot Dashboard** ⚡ 1.6M ★ 3.5  
Spring Boot Dashboard for VS Code  
Microsoft [Install](#)
- Spring Boot Extension Pack** ⚡ 1.3M ★ 5  
A collection of extensions for developing Spring ...  
Pivotal [Install](#)
- Spring Boot Support** ⚡ 96K ★ 5  
Spring Boot properties completion  
ecmel [Install](#)
- Java + Spring Extension Pack** ⚡ 53K ★ 5  
Some of the most popular and useful Java and S...  
Loiane Groner [Install](#)
- Spring Boot Snippets** ⚡ 59K ★ 5  
Useful snippets for Spring Boot projects.  
Developer Soapbox [Install](#)
- Ra Spring Light Theme** ⚡ 20K ★ 5  
Here is the best theme for your eye care!  
Rahman Yerli [Install](#)
- Spring Boot Developer Extension ...** ⚡ 28K ★ 5  
Useful extensions needed for effectively develop...  
Developer Soapbox [Install](#)
- Spring Theme** ⚡ 10K  
Spring Theme ported from the Spring TextMate ...  
gerane [Install](#)
- Extension Pack for Java** ⚡ 390ms  
Popular extensions for Java development that pr...  
Microsoft [Install](#)
- Azure Spring Apps** ⚡ 5K ★ 5  
An Azure Spring Apps extension for Visual Studi...  
Microsoft [Install](#)

Extension: Spring Boot Extension Pack X

## Spring Boot Extension Pack v0.2.0

Pivotal | ⚡ 1,375,369 | ★★★★★(13)

A collection of extensions for developing Spring Boot applications

[Install](#)

[Details](#) [Feature Contributions](#)

### Extension Pack (3)

- Spring Initializr Java Support** ⚡ 1.3M ★ 5  
A lightweight extension based on Spring Init...  
Pivotal [Install](#)

Instale essa extensão

#### Categories

Programming Languages Linters  
Extension Packs

#### Extension Resources

Marketplace Repository Pivotal

#### More Info

Published 11/28/2017, 17:43:15  
Last released 12/14/2022, 10:57:22  
Identifier pivotal.vscode-boot-dev-pack

## VS Code Spring Boot Application Development Extension Pack

(also known as [Spring Tools 4](#) for Visual Studio Code)

Collection of extensions for developing and deploying Spring Boot Application

### Spring Boot

The [Spring Boot Tools](#) extension provides:

- IDE Java tooling for developing and troubleshooting Spring Boot applications.
- Support for editing Spring Boot Application configuration properties files (`.properties` and `.yml`)

### Spring Initializr Java

The [Spring Initializr Java Support](#) extension provides support for generating quickstart Spring Boot Java projects with Spring Initializr API.

### Spring Boot Dashboard

File Edit Selection View Go Run Terminal Help Extension: Spring Boot Extension Pack - APIS-SPRING-BOOT - Visual Studio Code

EXTENSIONS: MARKETPLACE

spring

**Spring Initializr Java Support**  
A lightweight extension based on Spring Initializ...  
Microsoft

**Spring Boot Tools**  
Provides validation and content assist for Spring ...  
Pivotal

**Spring Boot Dashboard**  
Spring Boot Dashboard for VS Code  
Microsoft

**Spring Boot Extension Pack**  
A collection of extensions for developing Spring ...  
Pivotal

**Spring Boot Support**  
Spring Boot properties completion  
ecmel [Install](#)

**Java + Spring Extension Pack**  
Some of the most popular and useful Java and S...  
Loiane Groner [Install](#)

**Spring Boot Snippets**  
Useful snippets for Spring Boot projects.  
Developer Soapbox [Install](#)

**Ra Spring Light Theme**  
Here is the best theme for your eye care!  
Rahman Yerli [Install](#)

**Spring Boot Developer Extension ...**  
Useful extensions needed for effectively develop...  
Developer Soapbox [Install](#)

**Spring Theme**  
Spring Theme ported from the Spring TextMate ...  
gerane [Install](#)

**Extension Pack for Java**  
Popular extensions for Java development that pr...  
Microsoft

**Azure Spring Apps**  
An Azure Spring Apps extension for Visual Studi...  
Microsoft [Install](#)

**Spring Boot Extension Pack** v0.2.0

Pivotal | 1,375,369 | ★★★★★(13)

A collection of extensions for developing Spring Boot applications

[Disable](#) [Uninstall](#)

This extension is enabled globally.

**Details Feature Contributions**

**Extension Pack (3)**

**Spring Initializr Java Support**  
A lightweight extension based on Spring Init...  
Microsoft

**Categories**

Programming Languages Linters Extension Packs

**VS Code Spring Boot Application Development Extension Pack**

(also known as [Spring Tools 4](#) for Visual Studio Code)

Collection of extensions for developing and deploying Spring Boot Application

**Spring Boot**

The [Spring Boot Tools](#) extension provides:

- IDE Java tooling for developing and troubleshooting Spring Boot applications.
- Support for editing Spring Boot Application configuration properties files (`.properties` and `.yml`)

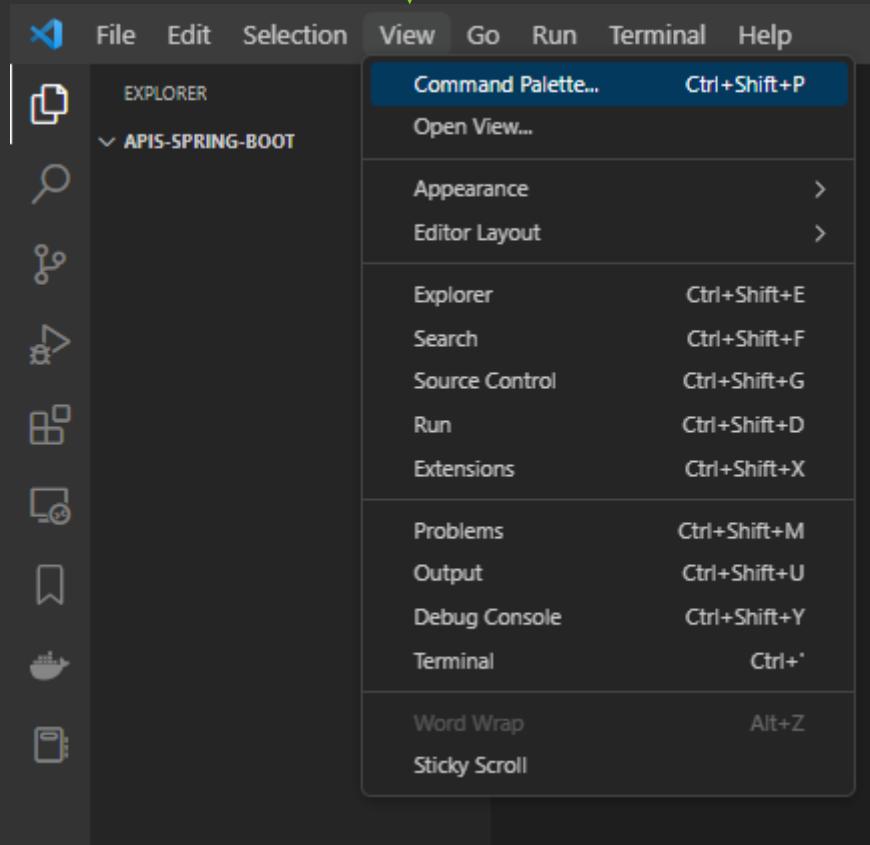
**Spring Initializr Java**

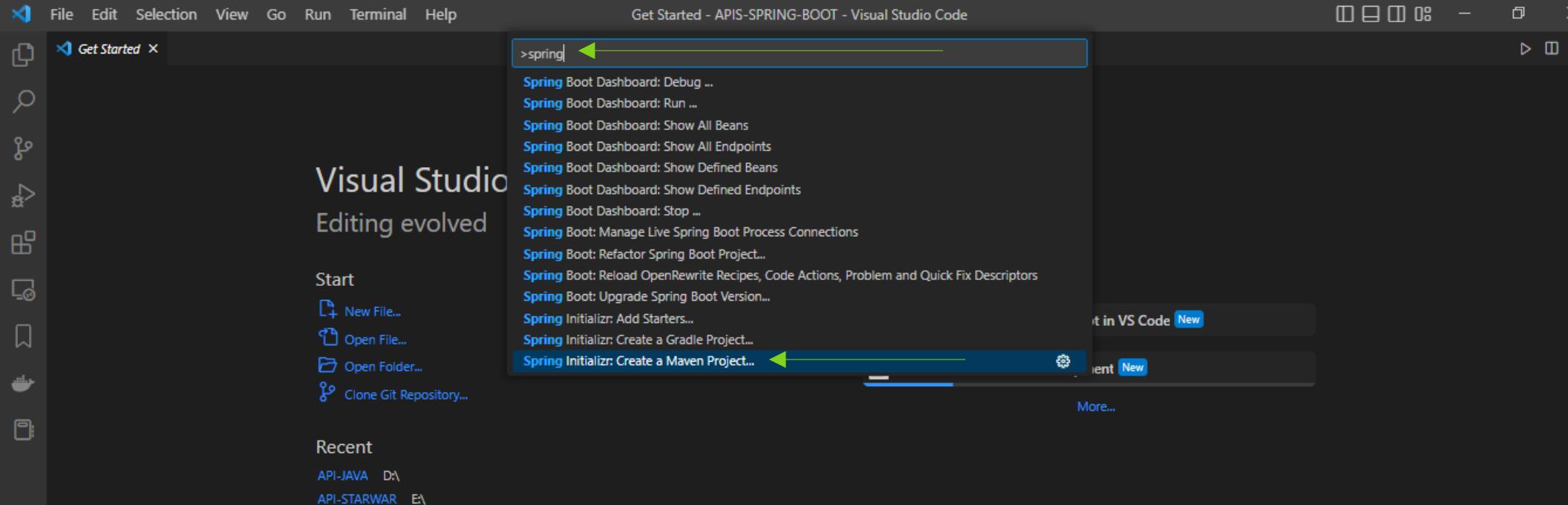
The [Spring Initializr Java Support](#) extension provides support for generating quickstart Spring Boot Java projects with Spring Initializr API.

**Spring Boot Dashboard**

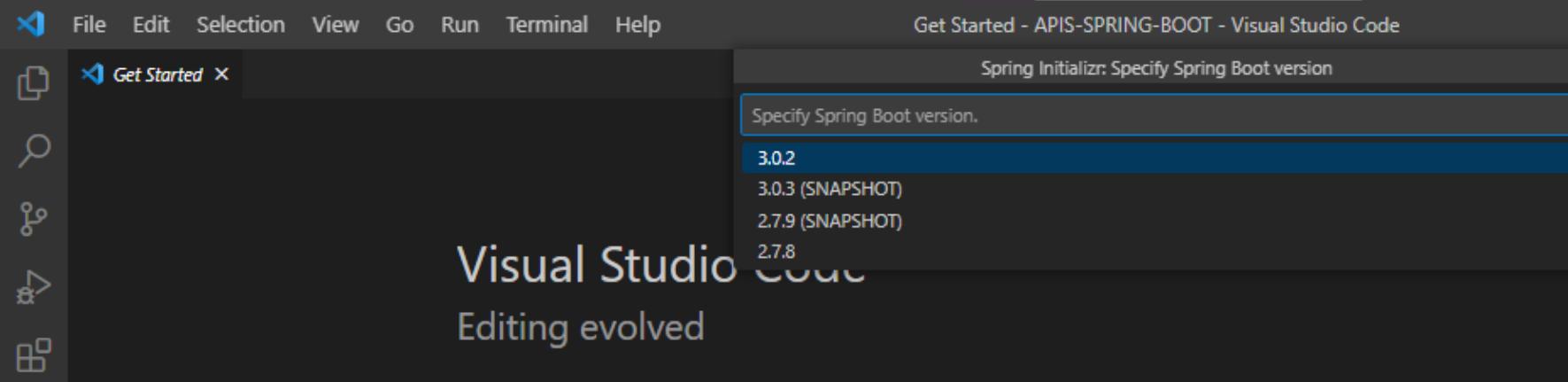
25°C Nublado Go Live 14:22

## CRIANDO O PROJETO

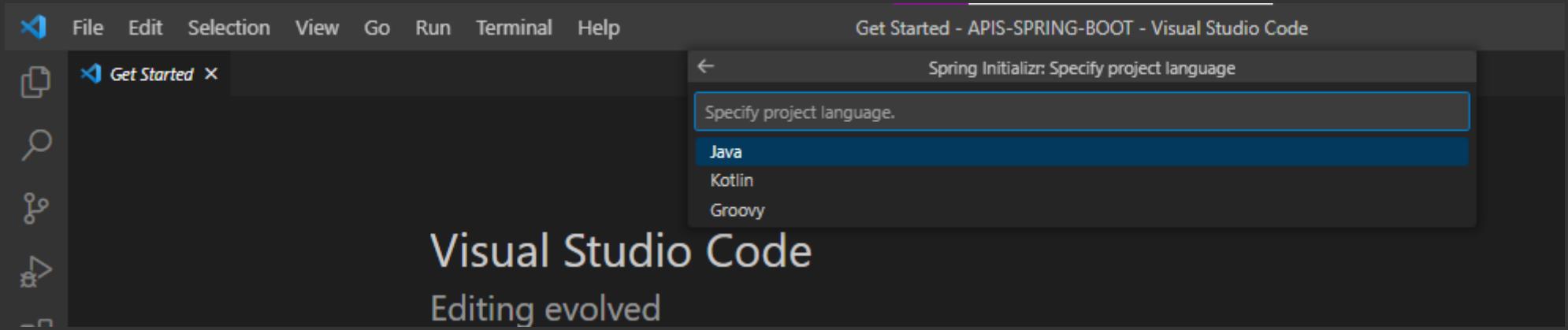




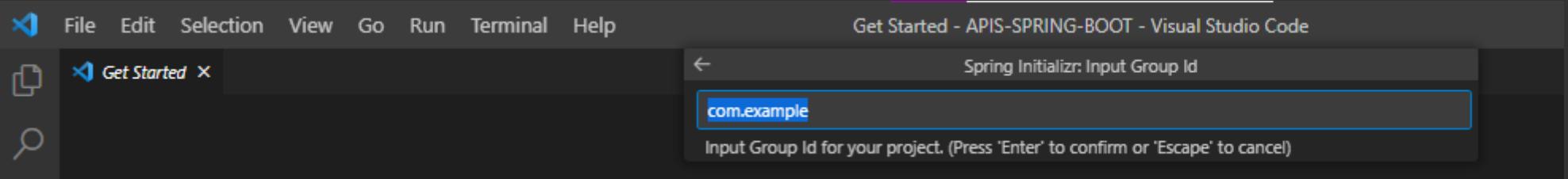
Digite spring e adicione spring initializr do maven  
Essa que esta marcada ai



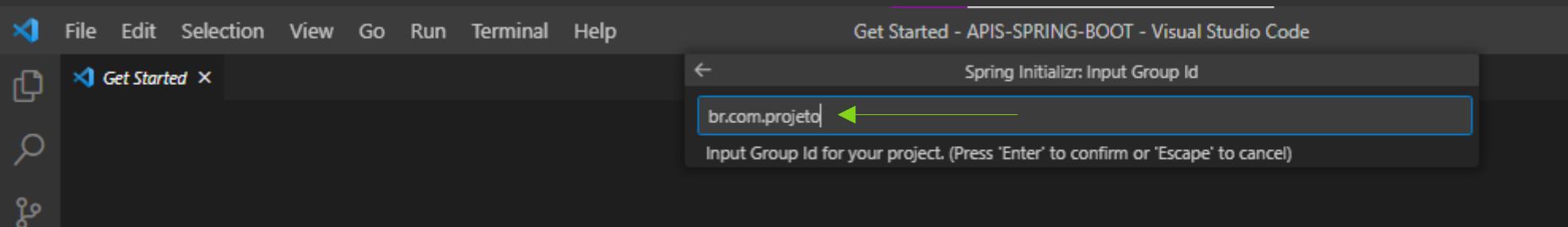
**Escolha sua versão nada impede de troca-la mais tarde**



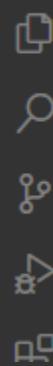
**Escolha a linguagem do projeto**



Group é o lugar onde vai ficar o projeto



Depois de dar o nome tecle enter



Get Started X



Spring Initializr: Input Artifact Id

demo

Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

# Visual Studio Code

Editing evolved

De nome ao projeto



Get Started X



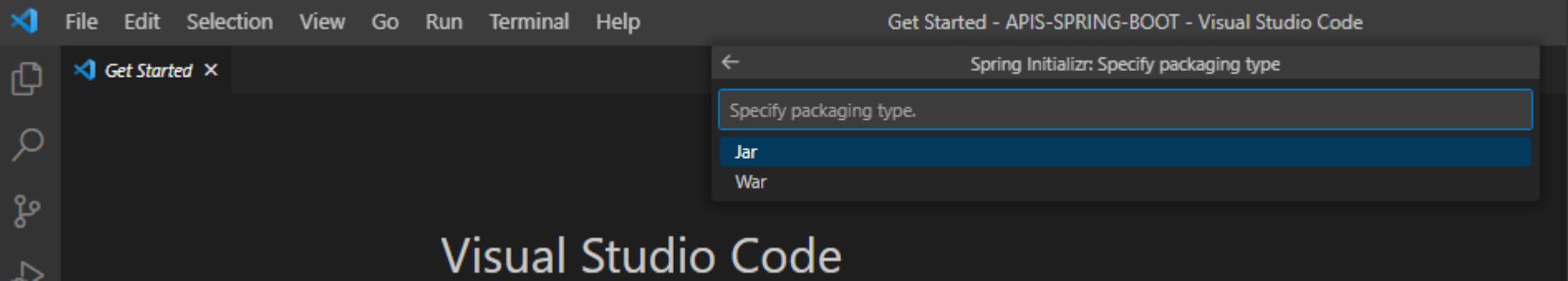
Spring Initializr: Input Artifact Id

api

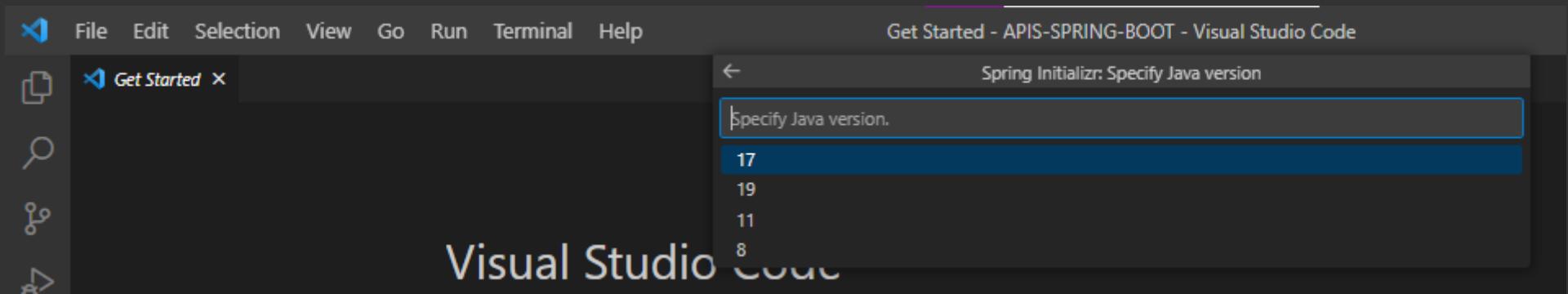
Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

# Visual Studio Code

Editing evolved



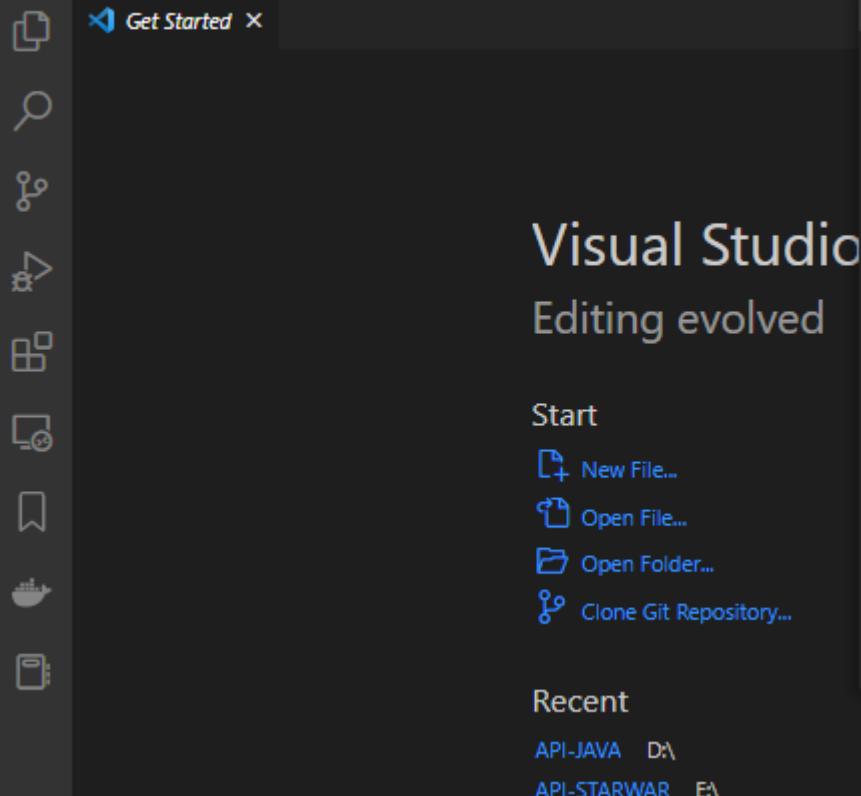
**Defina o empacotamento pra jar isso pode ser mudado  
futuramente**



**Escolha a versão compativel com a sua versão de java e do jdk a minha  
como disse é a 17 mais nesse ano 2023 já existe a 19**

File Edit Selection View Go Run Terminal Help

Get Started - APIS-SPRING-BOOT - Visual Studio Code



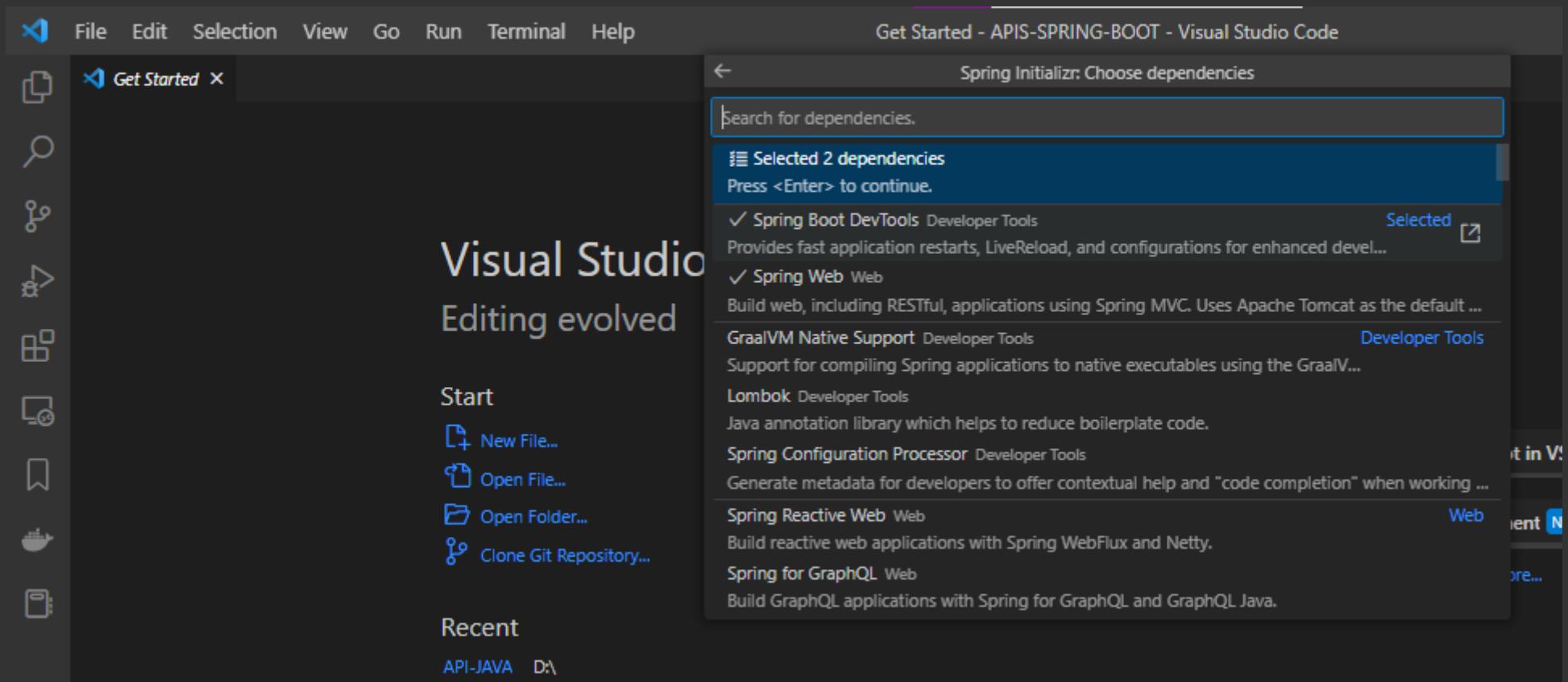
← Spring Initializr: Choose dependencies

Search for dependencies.

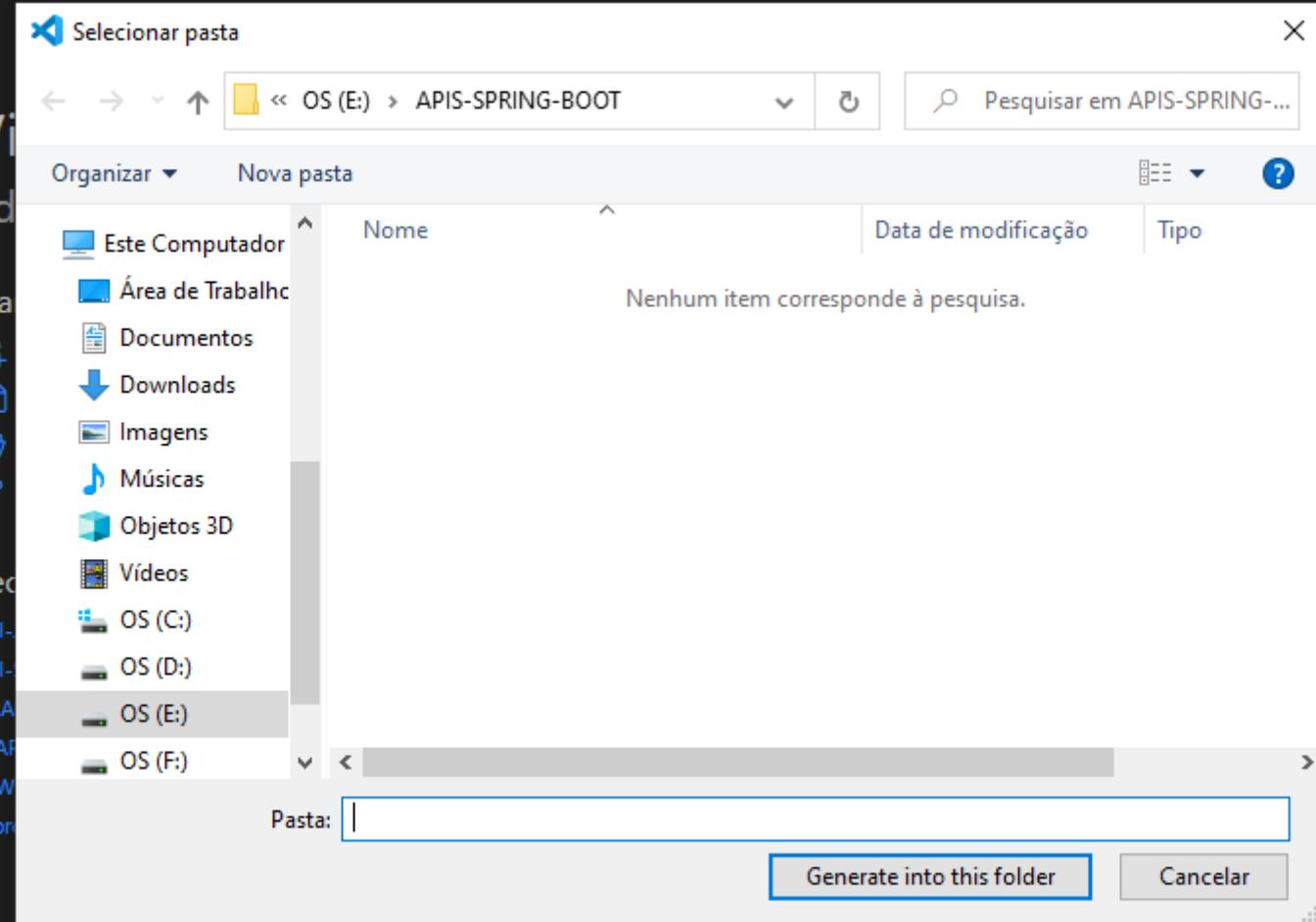
Selected 0 dependencies  
Press <Enter> to continue.

GraalVM Native Support	Developer Tools	Developer Tools
Support for compiling Spring applications to native executables using the GraalV...		
Spring Boot DevTools	Developer Tools	
Provides fast application restarts, LiveReload, and configurations for enhanced development expe...		
Lombok	Developer Tools	
Java annotation library which helps to reduce boilerplate code.		
Spring Configuration Processor	Developer Tools	
Generate metadata for developers to offer contextual help and "code completion" when working ...		
Spring Web	Web	Web
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the def...		
Spring Reactive Web	Web	Web
Build reactive web applications with Spring WebFlux and Netty.		
Spring for GraphQL	Web	Web
Build GraphQL applications with Spring for GraphQL and GraphQL Java.		

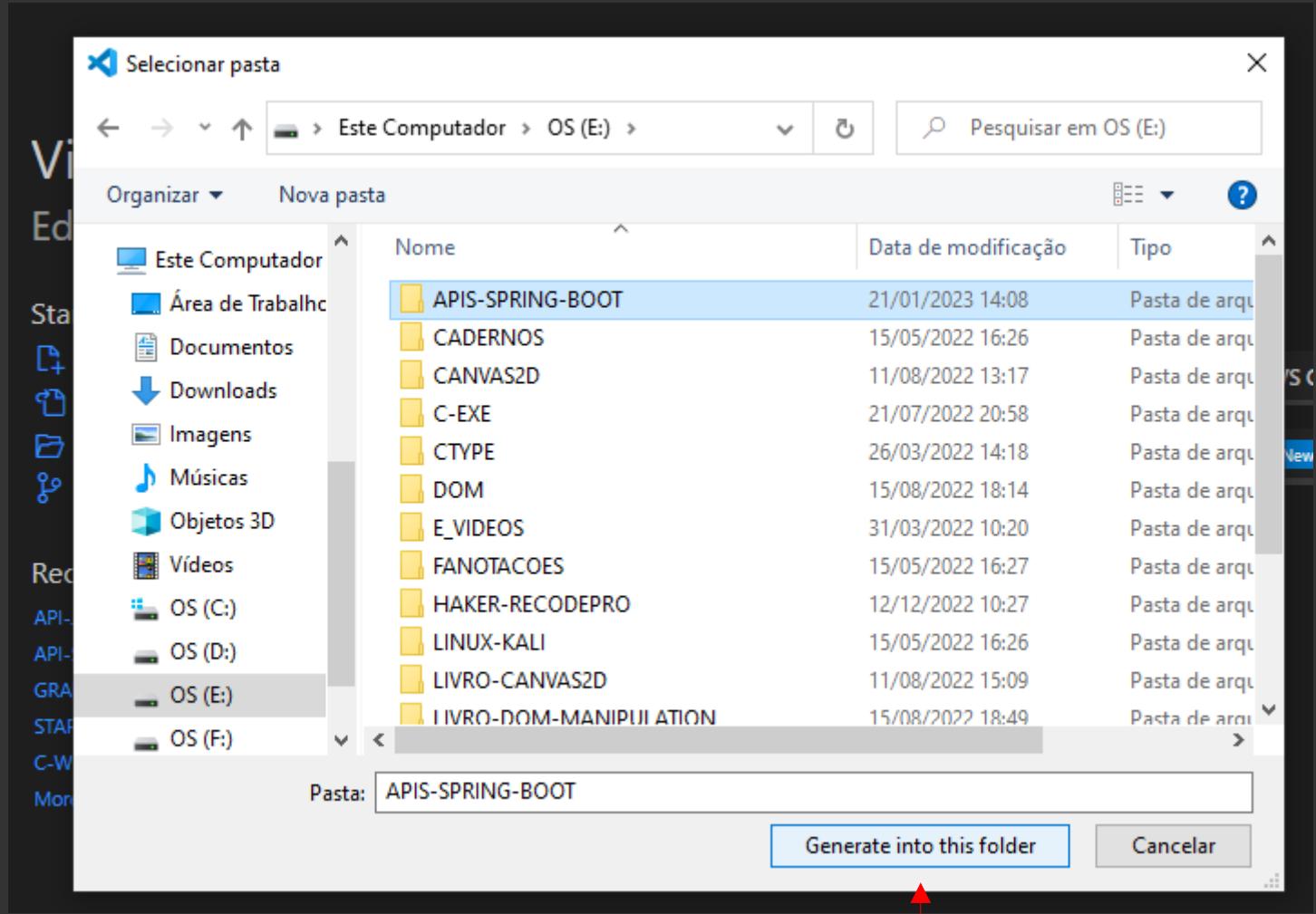
Adicione agora as dependencias do projeto  
O que são dependencias? São funções que o spring usa

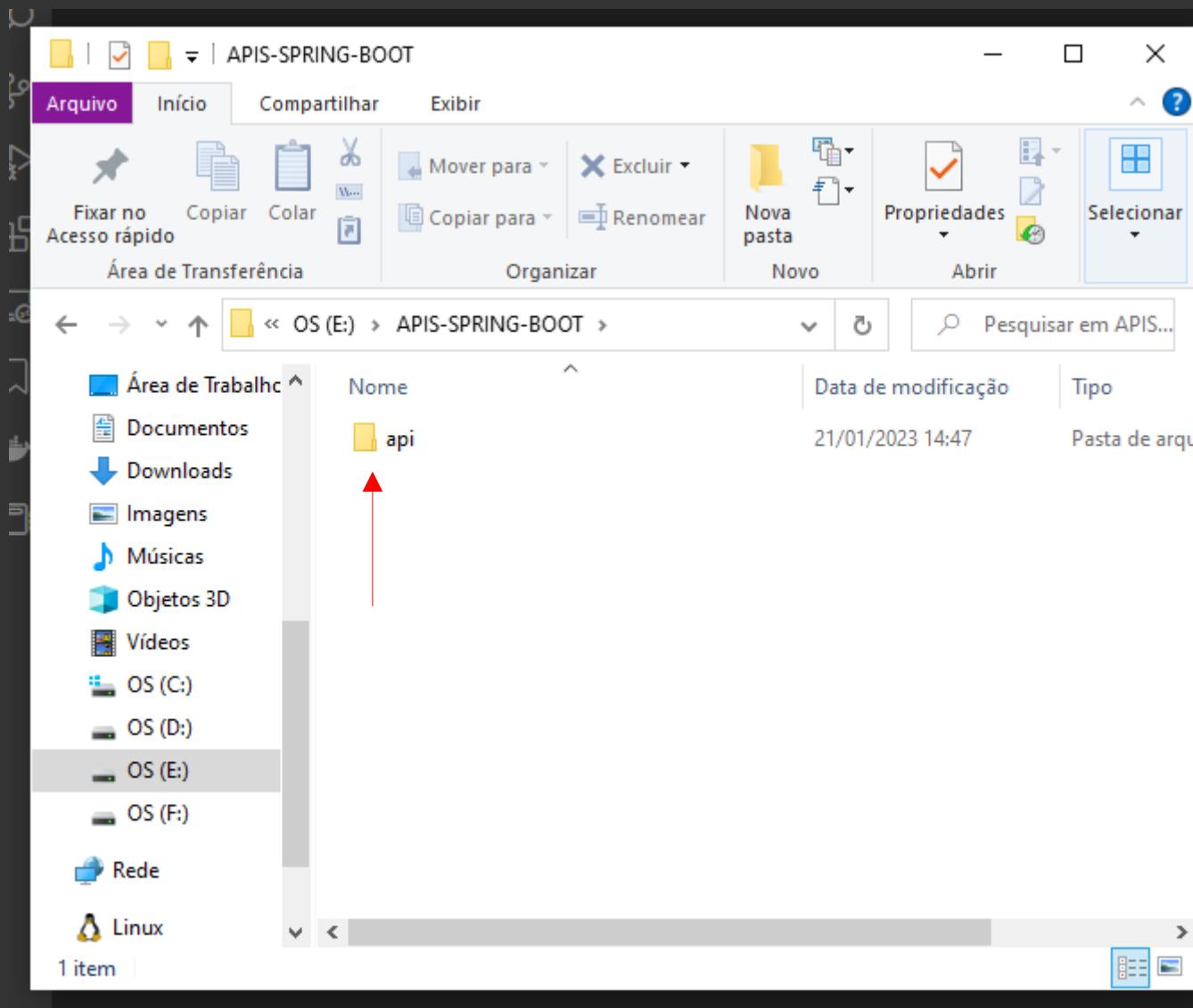


**Click para selecionar spring boot dev tools e spring web  
Por fim de enter para selecionar**

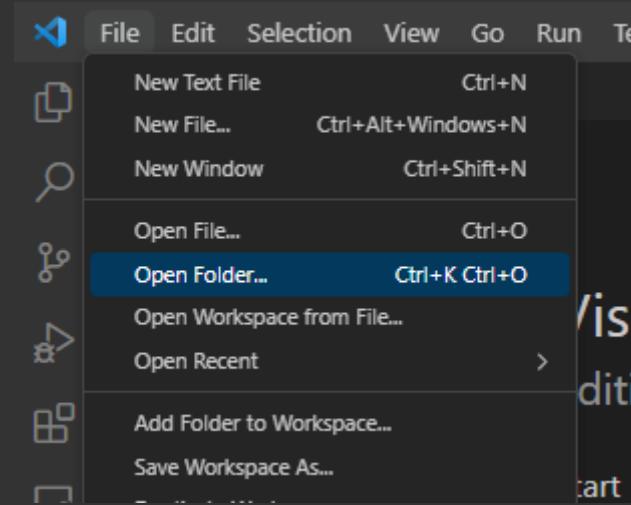


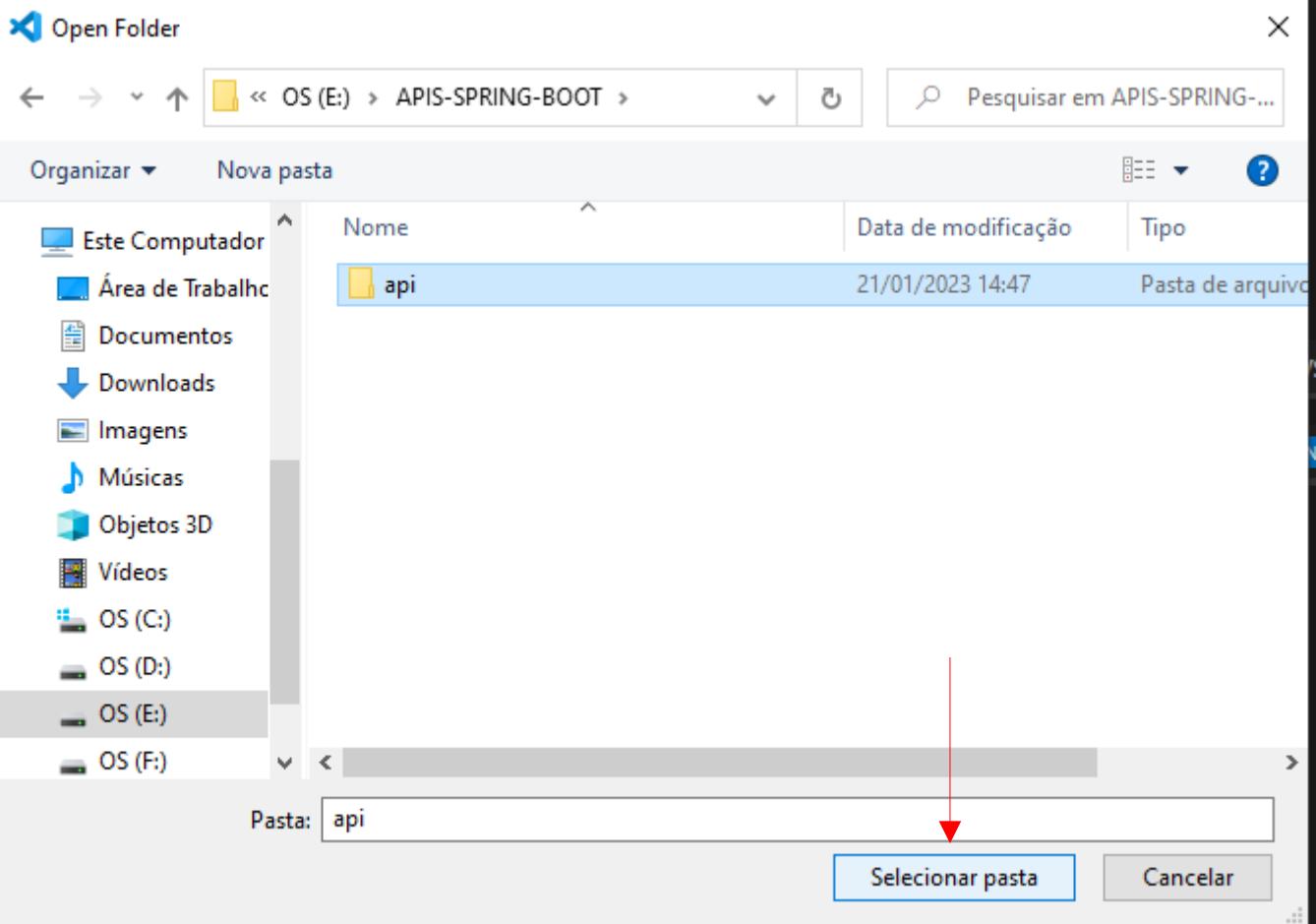
**Ele abre para você escolher onde vai salvar o projeto  
Você pode escolher qualquer pasta para salvar seu projeto**





## Abrindo a pasta que criamos







EXPLORER

- ... **Get Started** x
- API
  - > .mvn
  - > .vscode
  - > src
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml

# Visual Studio Code

## Editing evolved

### Start

- New File...
- Open File...
- Open Folder...
- Clone Git Repository...

### Recent

- APIS-SPRING-BOOT E:\
  - API-JAVA D:\
  - API-STARWARS E:\
  - GRAFICOS-JS F:\
  - STAR-WARS F:\
- More...

### Walkthroughs

- Getting Started with Spring Boot in VS Code New

- Get Started with Java Development New

More...

Espere todos os arquivos serem baixados

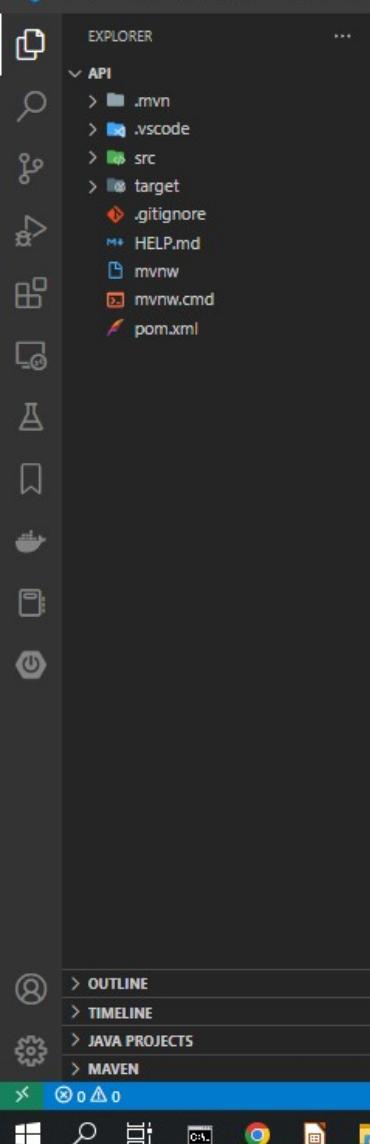
Show All Commands **Ctrl + Shift + /**

Go to File Ctrl +

**Find in Files** **Ctrl + Shift + F**

Start Debugging F

Toggle Terminal Ctrl +



## **EXECUTANDO PROJETO**

File Edit Selection View Go Run Terminal Help

ApiApplication.java - api - Visual Studio Code

EXPLORER

API

.mvn .vscode src

main

java\br\com\projeto\api

Aplication.java

resources test target

.gitignore HELP.md mvnw mvnw.cmd pom.xml

ApiApplication.java

```
src > main > java > br > com > projeto > api > ApiApplication.java > Language Support for Java(TM) by Red Hat > {} br.com.projeto.api
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ApiApplication {
8
9     Run | Debug ←
10    public static void main(String[] args) {
11        SpringApplication.run(primarySource: ApiApplication.class, args);
12    }
13
14 }
```

Preste atenção nessas duas opções  
Podemos rodar a aplicação com a opção run ou ir la embaixo em...



File Edit Selection View Go Run Terminal Help

ApiApplication.java - api - Visual Studio Code



EXPLORER

...

ApiApplication.java X

src > main > java > br > com > projeto > api > ApiApplication.java > Language Support for Java(TM) by Red Hat > {} br.c

```
1 package br.com.projeto.api;  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5  
6 @SpringBootApplication  
7 public class ApiApplication {  
8  
9     public static void main(String[] args) {  
10         SpringApplication.run(primarySource: ApiApplication.class, args);  
11     }  
12 }  
13 }  
14 }
```



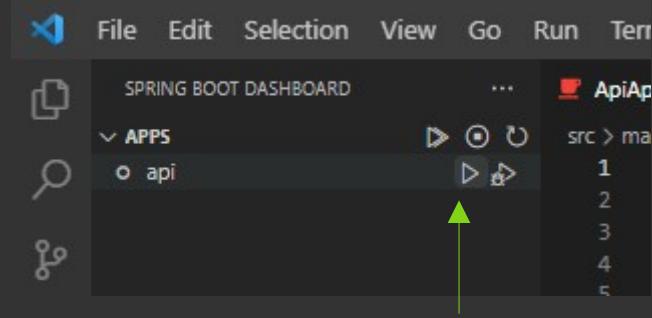
Nessa opção spring boot dashboard

A screenshot of a code editor showing a Spring Boot application structure. The left sidebar shows a project tree with a 'SPRING BOOT DASHBOARD' root, an 'APPS' folder containing an 'api' subfolder, and various icons for file operations like copy, paste, and search. The main editor area displays the 'ApiApplication.java' file:

```
1 package br.com.  
2  
3 import org.spring  
4 import org.spring  
5  
6 @SpringBootApp  
7 public class Ap  
8  
9     Run | Debug  
10    public stat  
11        SpringA  
12    }  
13 }  
14 }
```

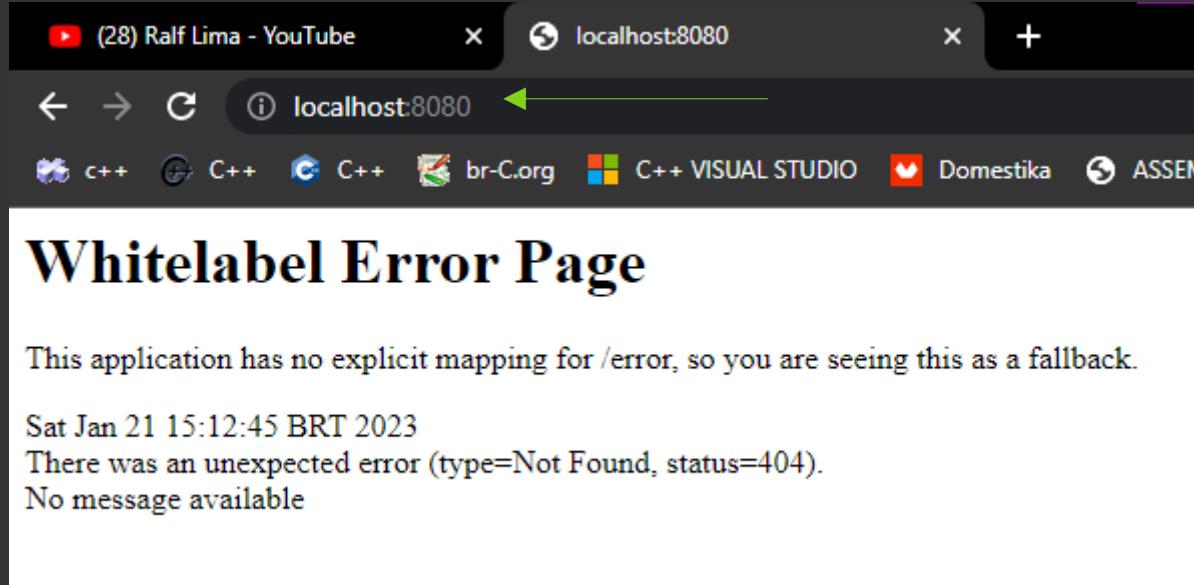
A green arrow points from the bottom of the sidebar towards the power button icon at the bottom center of the interface.

E rodamos o projeto  
Tanto faz qualquer uma das escolhas



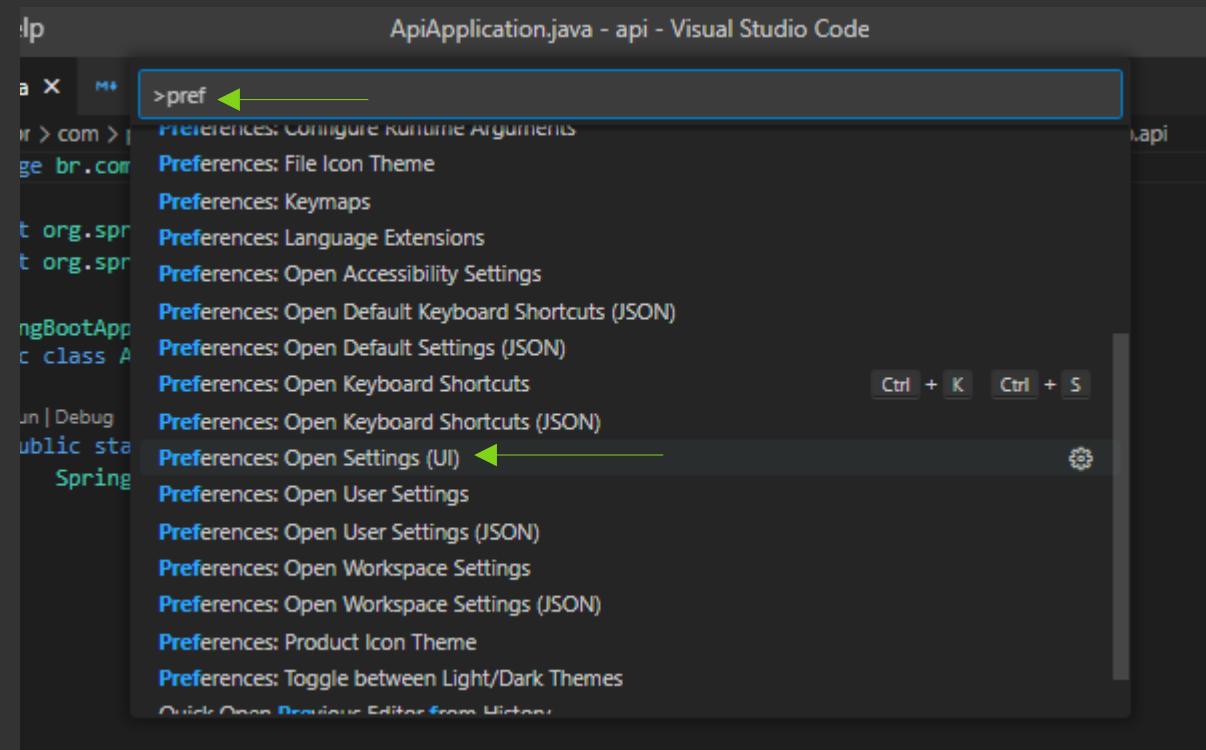
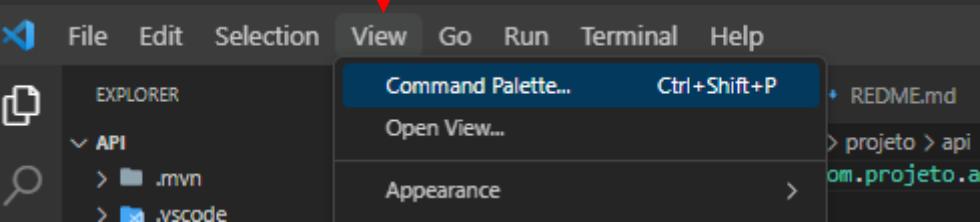
**EXECUTE O PROJETO PARA FAZER UM TESTE**





**ERRO BOM SIGNIFICA QUE ESTA RODANDO**

## **IMPLEMENTANDO O CONTROLLER PARA CORRIGIR O ERRO**





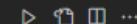
EXPLORER

...

ApiApplication.java

Settings

REDME.md



API  
> .mvn  
> .vscode  
src  
main  
> java  
> resources  
> test  
> target  
↳ .gitignore  
HELP.md  
mvnw  
mvnw.cmd  
pom.xml  
REDME.md

The screenshot shows the Visual Studio Code interface with a Java Spring Boot project open. The Explorer sidebar on the left displays the project structure:

- API
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Agora as pastas aparecem em cascata

The screenshot shows a Java project named "API" in the Explorer sidebar. The project structure includes a ".mvn" folder, a ".vscode" folder, a "src" folder containing "main", "java", "br", "com", and "projeto" subfolders, and an "api" folder. Inside the "api" folder, there is a file named "ApiApplication.java". The code in "ApiApplication.java" is as follows:

```
package br.com.projeto.api;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
@SpringBootApplication  
public class ApiApplication {  
    public static void main(String[] args) {  
        SpringApplication.run(primarySource: ApiApplication.class, args);  
    }  
}
```

A context menu is open over the "api" folder in the Explorer. The menu items are:

- New File...
- New Folder...** (highlighted)
- Reveal in File Explorer Shift+Alt+R
- Open in Integrated Terminal
- Add Folder to Java Source Path
- Remove Folder from Java Source Path
- Find in Folder... Shift+Alt+F

**Crie uma nova pasta dentro da pasta api com o nome controle  
Precisa ser este nome é obrigatorio? não**

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal
- Terminal Tab:** ApiApplication.java
- Explorer Tab:** API folder structure:
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
      - controle
      - resources
      - test
      - target
        - .gitignore
        - HELP.md
        - mvnw
        - mvnw.cmd
        - pom.xml
        - README.md

The screenshot shows the VS Code interface with a dark theme. The Explorer sidebar on the left displays the project structure:

- API
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
        - api
    - controle
  - resources
  - test
  - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - RFDFMF.md

The 'controle' folder is selected. A context menu is open over the 'controle' folder, with the 'New File...' option highlighted in blue.

ApiApplication.java

```
1 package br.com.projeto.api;  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5  
6 @SpringBootApplication  
7 public class ApiApplication {  
8  
9     public static void main(String[] args) {  
10         SpringApplication.run(primarySource:)
```

Criando uma class controle dentro da pasta controle

A screenshot of the Visual Studio Code interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "Controle.java - api - Visual Studio Code". The Explorer sidebar on the left shows a project structure under "API": .mvn, .vscode, src (expanded), main (expanded), java (expanded), br (expanded), com (expanded), projeto (expanded), api (expanded), controle (expanded), Controle.java, and ApiApplication.java. Below the src folder are resources, test, target, .gitignore, HELP.md, mvnw, mvnw.cmd, pom.xml, and README.md. The main editor area displays Java code:

```
src > main > java > br > com > projeto > api > controle > Controle.java > Controle
1 package br.com.projeto.api.controle;
2
3 public class Controle {
4
5 }
```

A code completion dropdown is open at the end of the third line ("public class Controle {"). It lists several suggestions starting with "Controle":

- class Controle
- interface Controle
- enum Controle
- record Controle()
- abstract class Controle
- @interface Controle

A screenshot of the Visual Studio Code interface. The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "Controle.java - api - Visual Studio Code". The Explorer sidebar on the left shows a project structure under "API": .mvn, .vscode, src, main, java, br, com, projeto. The main editor area displays Java code:

```
src > main > java > br > com > projeto > api > controle > Controle.java > Controle
1 package br.com.projeto.api.controle;
2
3 @RestController
4 public RestControllerEndpoint - org.springframework.boot...
5
6 }
7
```

A code completion tooltip is open at line 4, showing the suggestion "public RestControllerEndpoint - org.springframework.boot..." with a tooltip below it containing "org.springframework.boot.actuate.endpoint.web.annotation.RestControllerEndpoint".

Adicione a anotaion **@RestController** e seu import



EXPLORER

- ... ApiApplication.java
- ... Controle.java 1
- ... REDME.md
- API
  - > .mvn
  - > .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java 1

```
src > main > java > br > com > projeto > api > controle > Controle.java > Controle
1 package br.com.projeto.api.controle;
2
3 @RestController
4 public class Controle {
5     ...
6 }
7 }
```

@RestController

- ↳ [RestController - org.springframework.web.bind.annotation.RestController](#)
- ↳ [RestControllerAdvice - org.springframework.web.bind.annotation.RestControllerAdvice](#)
- ↳ [RestControllerEndpoint - org.springframework.boot.autoconfigure.endpoint.RestControllerEndpoint](#)
- ↳ [ResponseStatus - org.springframework.web.bind.annotation.StatusResponse](#)
- ↳ [RegisterReflectionForBinding - org.springframework.web.method.HandlerMethod](#)
- ↳ [RequestScope - org.springframework.web.context.request.RequestScope](#)
- ↳ [RequestMapping - org.springframework.web.bind.annotation.RequestMapping](#)
- ↳ [Repository - org.springframework.stereotype.Repository](#)
- ↳ [ConditionalOnRepositoryType - org.springframework.context.annotation.ConditionalOnRepositoryType](#)
- ↳ [ImportRuntimeHints - org.springframework.context.annotation.ImportRuntimeHints](#)

org.springframework.web.bind.annotation.RestController

A convenience annotation that is itself annotated with `@Controller` and `@ResponseBody`.

Types that carry this annotation are treated as controllers where `@RequestMapping` methods assume `@ResponseBody` semantics by default.

**NOTE:** `@RestController` is processed if an appropriate `HandlerMapping` - `HandlerAdapter` pair is configured such as the `RequestMappingHandlerMapping` - `RequestMappingHandlerAdapter` pair which are the default in the MVC Java config and the MVC namespace.

- **Since:**
  - 4.0
- **Author:**

The screenshot shows a Java project structure in the Explorer sidebar and the corresponding code in the main editor area.

**Project Structure:**

- API** (selected)
- .mvn
- .vscode
- src** (selected)
- main
  - java
    - br
      - com
        - projeto
          - api
            - controle
              - Controle.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Controle.java - api - Visual Studio Code

Controle.java

```
src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java
```

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.RestController;
4
5 @RestController
6 public class Controle {
7
8
9 }
10
```

A screenshot of the Visual Studio Code interface. The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "Controle.java - api - Visual Studio Code". The Explorer sidebar on the left shows a project structure under "API": ".mvn", ".vscode", "src" (with "main" and "java" subfolders), "br" (with "com", "projeto", "api", and "controle" subfolders), "Controle.java" (selected), "ApiApplication.java", "resources", "test", "target", ".gitignore", "HELP.md", and "mvnw". The main editor area displays Java code:

```
2  
3     import org.springframework.web.bind.annotation.RestController;  
4  
5     @RestController  
6     public class Controle {  
7  
8         @Get  
9         public GetExchange - org.springframework.web.service.annotation.GetExchange  
10            GetMapping - org.springframework.web.bind.annotation.GetMapping  
11        }  
12        Generated - jakarta.annotation  
13        Generated - javax.annotation.processing  
14    }
```

A tooltip is open over the "@Get" annotation, providing information about the annotation:

- Shortcut for @HttpExchange for HTTP GET requests.
- Since: 6.0
- Author: Rossen Stoyanchev

Adicione a anotation **@GetMapping** e seu import

The screenshot shows a Java Spring Boot application structure in the Explorer sidebar and its corresponding code in the main editor area.

**File Structure (Explorer):**

- API
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Controle.java - api - Visual Studio Code

```
ApiApplication.java Controle.java x README.md
```

src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping
10    public String mensagem(){
11        return "Hello World";
12    }
13
14 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - api - Visual Studio Code
- Explorer View:** Shows the project structure under API:
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
    - resources
    - test
    - target
      - .gitignore
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
      - REDME.md
- Editor View:** Controle.java file open, showing the following code:

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping("")
10    public String mensagem(){
11        return "Hello World";
12    }
13
14 }
```
- Bottom Status Bar:** Rota especificada dentro dos parentesis ""  
De run e volte no navegador e atualize a pagina

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Title:** Controle.java - api - Visual Studio Code
- Editor Content:** The code for `Controle.java` is displayed, which defines a REST controller for "Hello World".

```
src > main > java > br > com > projeto > api > controle > Controle.java > ...
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping("")
10    public String mensagem(){
11        return "Hello World";
12    }
13
14 }
15
```

- Sidebar:** Shows the project structure under "SPRING BOOT DASHBOARD" with "APPS" expanded and "api" selected.
- Left Panel:** Displays various icons for file operations like Open, Save, Find, and Run.

File Edit Selection View Go Run Terminal Help ApiApplication.java - api - Visual Studio Code

EXPLORER

API

- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java

src > main > java > br > com > projeto > api > ApiApplication.java > ...

```
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ApiApplication {
8     Run | Debug
9     public static void main(String[] args) {
10         SpringApplication.run(primarySource: ApiApplication.class, args);
11     }
12 }
13
14 }
```

RESOLVI O ERRO EXECUTANDO POR AQUI RUN

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
'/actuator'
2023-01-21T16:07:10.769-03:00 INFO 6944 --- [on(1)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing
spatcherServlet'
2023-01-21T16:07:10.784-03:00 INFO 6944 --- [on(1)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializin
2023-01-21T16:07:10.790-03:00 INFO 6944 --- [on(1)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed i
2023-01-21T16:07:10.876-03:00 INFO 6944 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat star
th context path ''
2023-01-21T16:07:11.048-03:00 INFO 6944 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started Api
process running for 8.088)
```

This method cannot be moved, since no possible targets have been found. Only a class which is reachable from within this method can be a valid target. The target must therefore be the declaring class of a parameter or field type. In addition the target must be writable.

Source: Language Support for Java(TM) by Red Hat (Extension)

Ln 14, Col 1 Tab Size: 4 UTF-8 LF {} Java Go Live

Spring Boot-ApiApplication<api> (api)

26°C Nublado



File Edit Selection View Go Run Terminal Help

Controle.java - api - Visual Studio Code



EXPLORER

...

ApiApplication.java

Controle.java X

REDME.md



API

.mvn

.vscode

src

main

java

br

com

projeto

api

controle

Controle.java

ApiApplication.java

resources

test

target

.gitignore

HELP.md

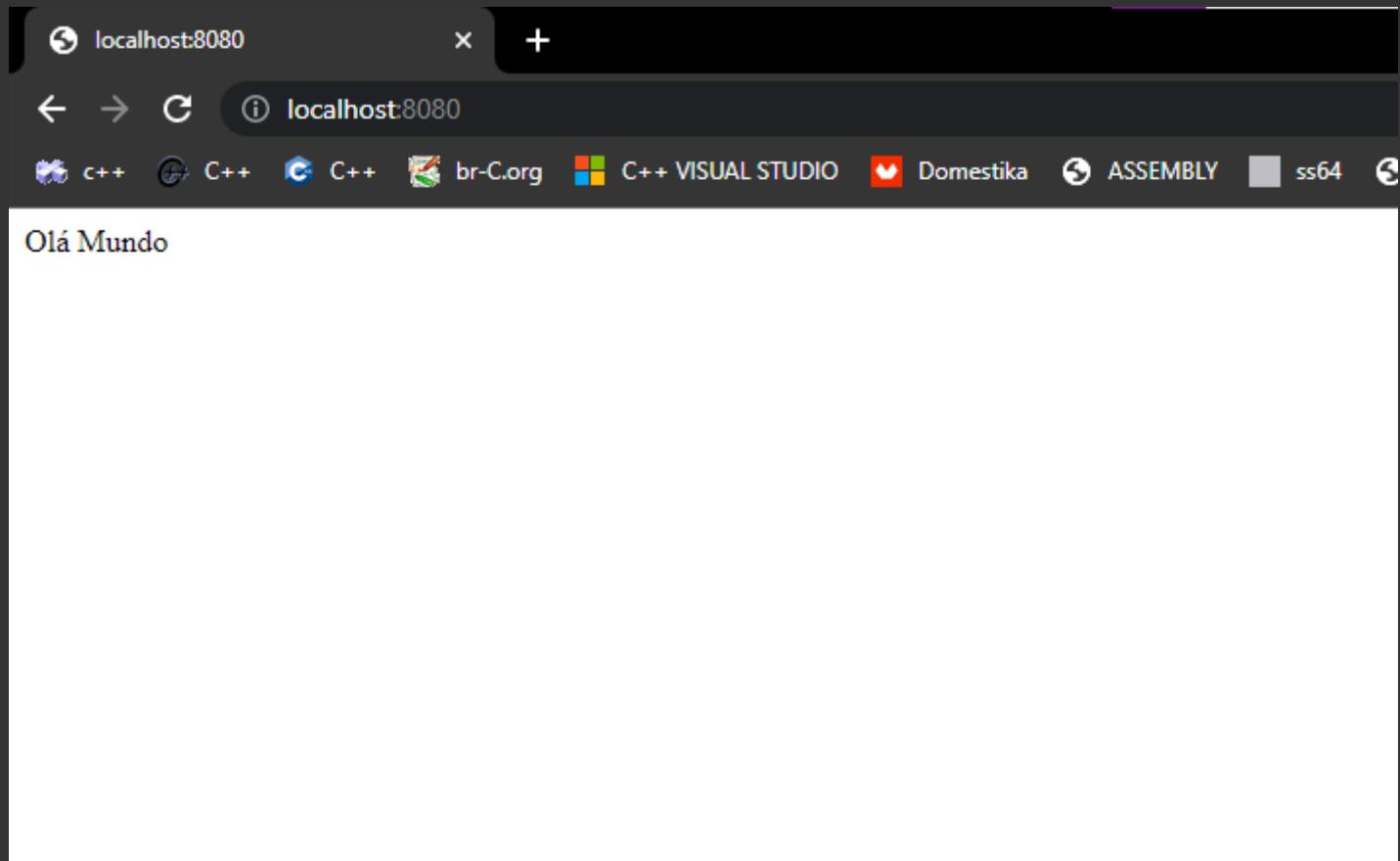
mvnw

mvnw.cmd

pom.xml

REDME.md

```
src > main > java > br > com > projeto > api > controle > Controle.java > ...
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     http://127.0.0.1:8080/
10    @GetMapping("/")
11    public String mensagem(){
12        return "Hello World";
13    }
14}
15
```



```
5  
6 @RestController  
7 public class Controle {  
8  
9     http://127.0.0.1:8080/  
10    @GetMapping("/") ←  
11    public String mensagem(){  
12        return "Hello World";  
13    }  
14 }  
15 }
```

Resolvi adicionando o / na url

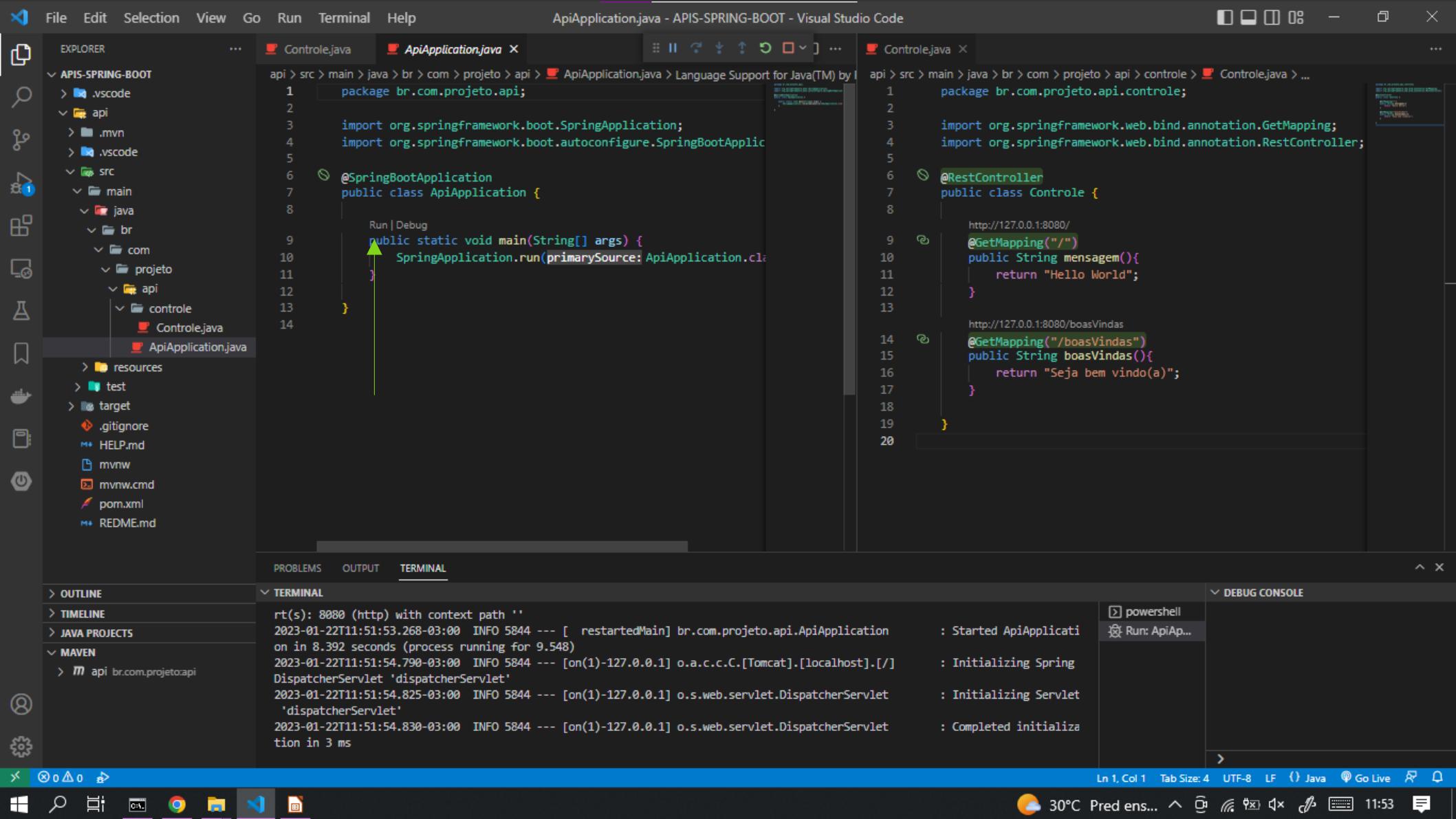
## #6 **@PathVariable**

The screenshot shows the Visual Studio Code interface with the following details:

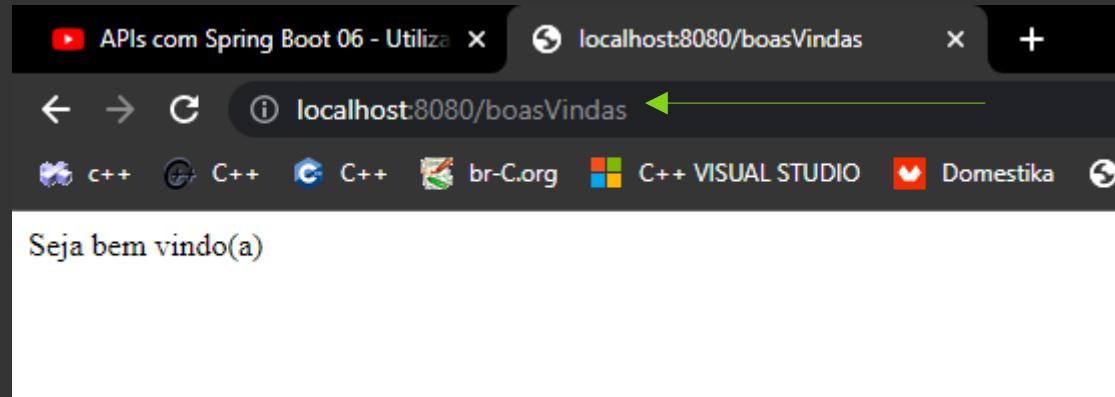
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "src/main/java" folder contains "br.com.projeto.api" which further contains "controle" and "Controle.java". Other files like "ApiApplication.java", ".gitignore", and "pom.xml" are also listed.
- Code Editor:** Displays the "Controle.java" file content:

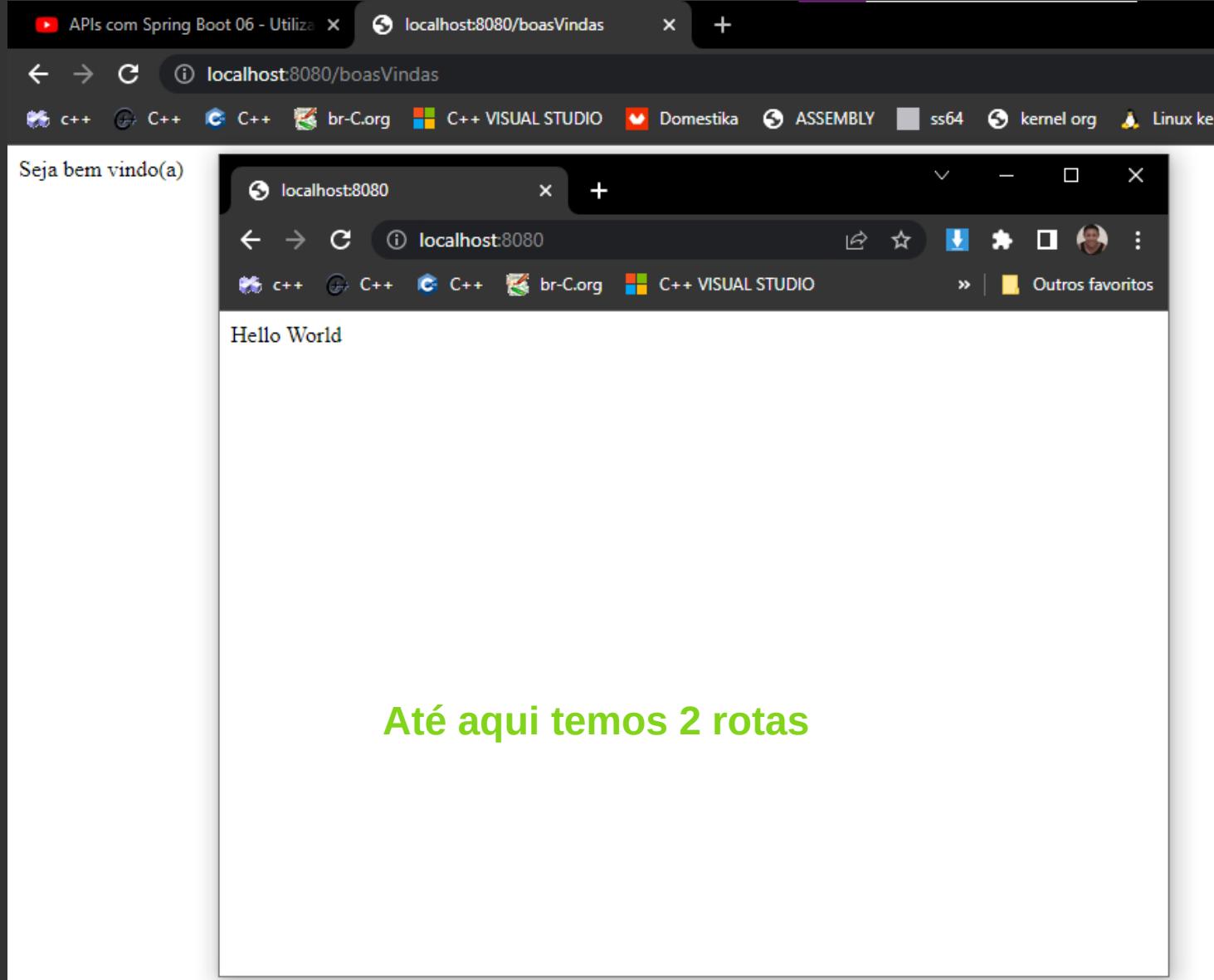
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping("/")
10    public String mensagem(){
11        return "Hello World";
12    }
13
14    @GetMapping("/boasVindas")
15    public String boasVindas(){
16        return "Seja bem vindo(a)";
17    }
18
19 }
20
```
- Terminal:** Shows the command line prompt "PS E:\APIS-SPRING-BOOT>".
- Bottom Status Bar:** Includes tabs for PROBLEMS, OUTPUT, and TERMINAL, with TERMINAL being the active tab.

**Crie uma nova rota  
Depois disso salve va no  
navegador e atualize  
E insira a nova rota**



# rota







EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

&lt; api

&gt; .mvn

&gt; .vscode

&lt; src

&lt; main

&lt; java

&lt; br

&lt; com

&lt; projeto

&lt; api

&lt; controle

Controle.java 3

ApiApplication.java

&gt; resources

&gt; test

&gt; target

&lt; .gitignore

&lt; HELP.md

&lt; mvnw

&lt; mvnw.cmd

&lt; pom.xml

&lt; README.md

Controle.java 3

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > boasVindas()  
1      package br.com.projeto.api.controle;  
2  
3      import org.springframework.web.bind.annotation.GetMapping;  
4      import org.springframework.web.bind.annotation.RestController;  
5  
6      @RestController  
7      public class Controle {  
8  
9          @GetMapping("/")  
10         public String mensagem(){  
11             return "Hello World";  
12         }  
13  
14         @GetMapping("/boasVindas/{nome}")  
15         public String boasVindas(@PathVariable{  
16             return "Seja bem vindo(a)"  
17         }  
18  
19     }
```



Adicione a **@PathVariable** e seu import

• PathParam - jakarta.websocket.server  
• PathVariable - org.springframework.web.bind.annotation  
• PatchExchange - org.springframework.web.service.a...  
• PatchMapping - org.springframework.web.bind.annot...

jakarta.websocket.server

Used to annotate method

endpoints the the ServerF

with a ServerEndpointVal

template.

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio C

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md

Controle.java X

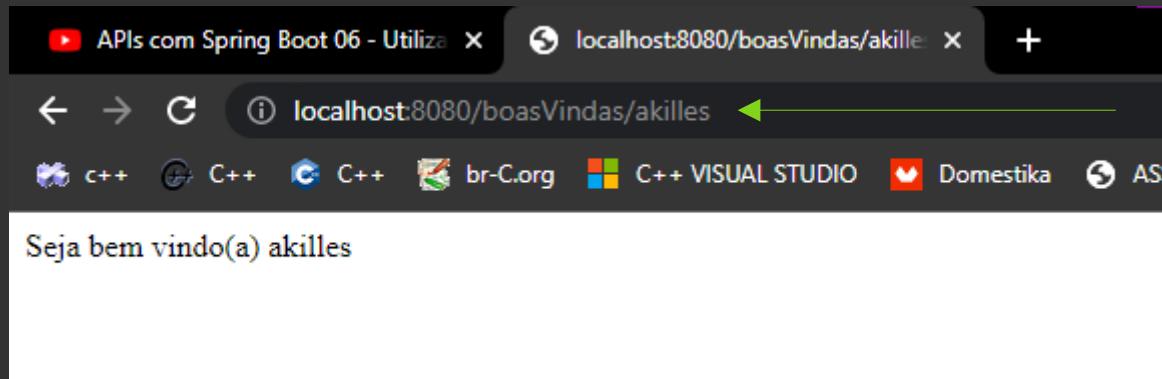
api > src > main > java > br > com > projeto > api > controle > Controle.java > ...

```
2
3     import org.springframework.web.bind.annotation.GetMapping;
4     import org.springframework.web.bind.annotation.PathVariable;
5     import org.springframework.web.bind.annotation.RestController;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14
15        @GetMapping("/boasVindas/{nome}")
16        public String boasVindas(@PathVariable String nome){
17            return "Seja bem vindo(a)";
18        }
19
20    }
```

{nome} faz o spring pegar o nome dinamicamente  
@PathVariable adicionado ao metodo só falta concatenar

```
Controle.java - APIS-SPRING-BOOT - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
APIS-SPRING-BOOT
> .vscode
< api
> .mvn
> .vscode
src
> main
> java
> br
> com
> projeto
> api
> controle
Controle.java
ApiApplication.java
> resources
> test
> target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
README.md
api > src > main > java > br > com > projeto > api > controle > Controle.java > ...
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.PathVariable;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 public class Controle {
9
10    @GetMapping("/")
11    public String mensagem(){
12        return "Hello World";
13    }
14
15    @GetMapping("/boasVindas/{nome}")
16    public String boasVindas(@PathVariable String nome){
17        return "Seja bem vindo(a)" + nome;
18    }
19
20 }
21
```

Concatenando com nome  
Agoar salve volte no navegador e teste





Se não for passado nenhum nome da esse erro  
Mas tá de boa até aqui

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio

EXPLORER

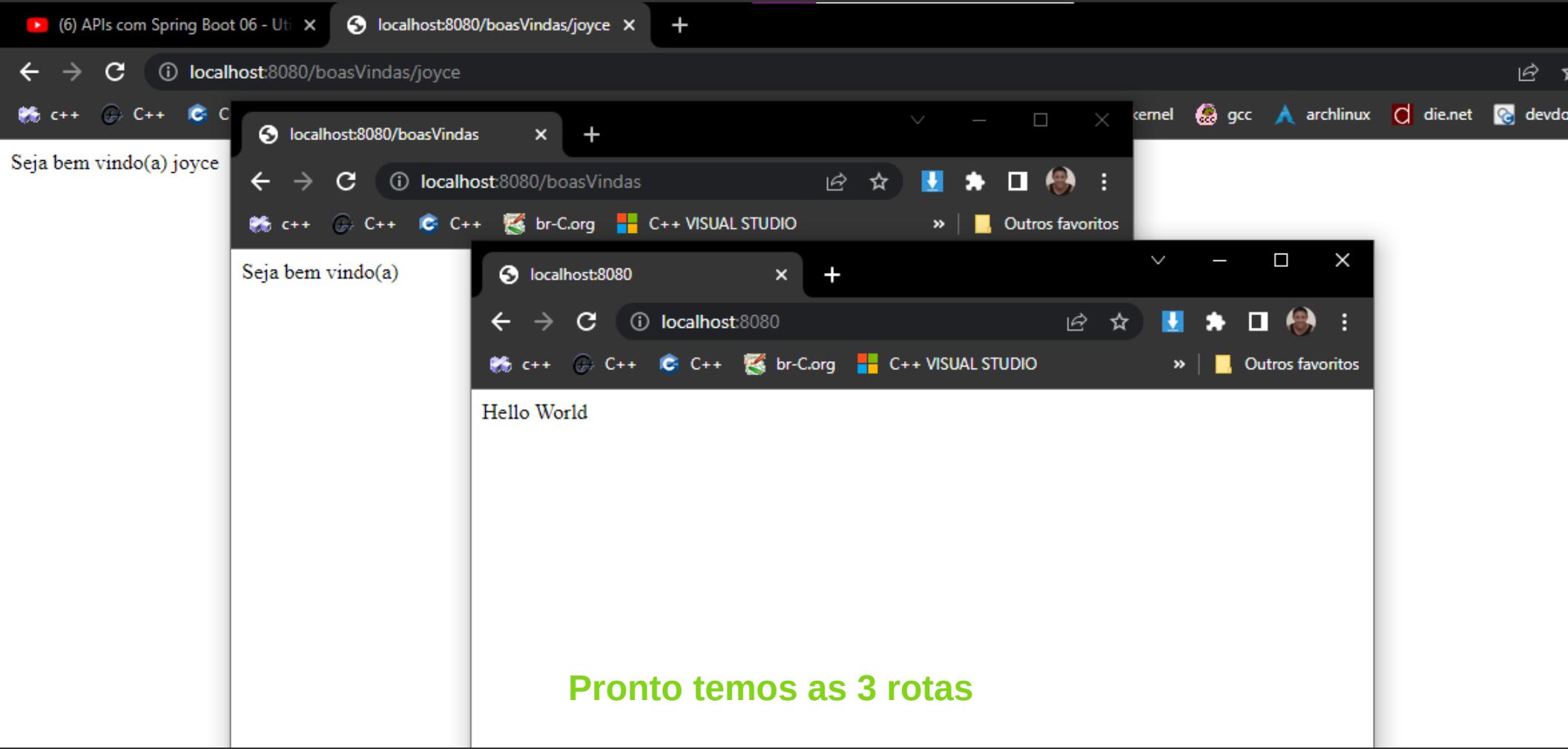
APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api- controle
  - Controle.java
  - ApiApplication.java
- resources
- test
- target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Controle.java

```
2
3     import org.springframework.web.bind.annotation.GetMapping;
4     import org.springframework.web.bind.annotation.PathVariable;
5     import org.springframework.web.bind.annotation.RestController;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14
15        @GetMapping("/boasVindas")
16        public String boasVindas(){
17            return "Seja bem vindo(a) ";
18        }
19
20        @GetMapping("/boasVindas/{nome}")
21        public String boasVindas(@PathVariable String nome){
22            return "Seja bem vindo(a) " + nome;
23        }
24
25    }
```

Podemos resolver o erro adicionando a rota que falta



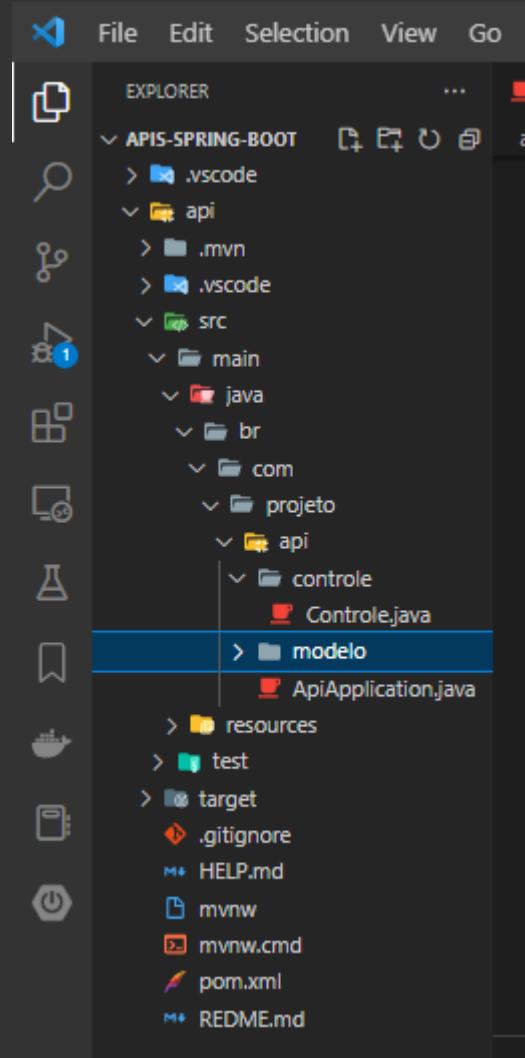
#7  
**MODELOS**

**o que são modelos?  
servem pra capturar os dados vindos de uma url  
e criam tabelas sem precisar escrever o sql**

The screenshot shows the VS Code interface with a Java Spring Boot project named "APIS-SPRING-BOOT". The project structure in the Explorer sidebar includes ".vscode", "api", ".mvn", ".vscode", "src", "main", "java", "br", "com", "projeto", and "api". The "api" folder under "src/main/java" is currently selected. A context menu is open over this folder, with the "New Folder..." option highlighted. Other options in the menu include "New File...", "Reveal in File Explorer", "Open in Integrated Terminal", "Add Folder to Java Source Path", "Remove Folder from Java Source Path", and "Find in Folder...".

```
2
3     import org.springframework.web.bind.annotation.*;
4     import org.springframework.web.bind.annotation.*;
5     import org.springframework.web.bind.annotation.*;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
}
```

Crie uma nova pasta dentro da pasta api chamada modelo



The screenshot shows the Visual Studio Code interface with a Java Spring Boot project named "APIS-SPRING-BOOT". The project structure in the Explorer sidebar includes ".vscode", "api", ".mvn", ".vscode", "src" (containing "main" and "java" folders), and "modelo" (containing "ApiApplication.java"). The "java" folder has subfolders "br", "com", "projeto", "api", and "controle", with "Controle.java" being the active file.

The code in "Controle.java" is:

```
2
3     import org.springframework.web.bind.annotation.*;
4     import org.springframework.web.bind.annotation.RestController;
5     import org.springframework.web.bind.annotation.GetMapping;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14
15        @GetMapping("/boasVindas")
16        public String boasVindas(){
17            return "Bem-vindo ao meu projeto!";
18        }
19    }
```

A context menu is open over the "modelo" folder in the Explorer sidebar, with the "New File..." option highlighted. Other options in the menu include "New Folder...", "Reveal in File Explorer", "Open in Integrated Terminal", "Add Folder to Java Source Path", and "Remove Folder from Java Source Path".

Dentro da pasta **modelo** crie uma classe chamada **Pessoa.java**



File Edit Selection View Go Run Terminal Help

• Pessoa.java - APIS-SPRING-BOOT - Visual Studio



EXPLORER  
APIS-SPRING-BOOT



> .vscode  
< api  
> .mvn  
> .vscode  
< src



< main  
< java  
< br  
< com  
< projeto  
< api  
< controle  
Controle.java



< modelo  
Pessoa.java  
ApiApplication.java



> resources  
> test  
> target  
.gitignore



HELP.md



mvnw



mvnw.cmd



pom.xml



README.md



Controle.java



Pessoa.java



api > src > main > java > br > com > projeto > api > modelo > Pessoa.java

```
1 package br.com.projeto.api.modelo;  
2  
3 public class Pessoa {  
4     |  
5 }  
6
```

class Pessoa  
interface Pessoa  
enum Pessoa  
record Pessoa()  
abstract class Pessoa  
@interface Pessoa

PROBLEMS

OUTPUT

TERMINAL

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoa.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
  - Pessoa.java 2
  - ApiApplication.java
- Code Editor:** The file Pessoa.java is open, showing the following code:

```
1 package br.com.projeto.api.modelo;
2
3 public class Pessoa {
4     private String nome;
5     private String idade;
6 }
7
8 }
```
- Toolbar:** Standard VS Code icons for file operations like Open, Save, Find, and Refresh.

Inicialmente crie esses 2 atributos nome e idade

**Lembrando que uma classe em java serve de modelo para construir objetos  
atravez do seu construtor**

**Lembre-se de deixar os atributos sempre privados pois podemos manipula-  
los atravez dos metodos get e set**

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoajava - APIS-SPRING-BOOT
- Explorer View:** Shows the project structure under 'APIS-SPRING-BOOT'. The 'src' folder contains 'main', 'java', 'br', 'com', 'projeto', 'api', 'controle', 'modelo', 'resources', 'test', and 'target' folders. Inside 'src/main/java', there are 'br.com.projeto.api.modelo' and 'br.com.projeto.api.controle' packages. 'br.com.projeto.api.modelo' contains 'Pessoa.java' and 'ApiApplication.java'. 'br.com.projeto.api.controle' contains 'Controle.java'. 'resources' contains 'application.properties' and 'application.yml'. 'target' contains 'classes', '.mvnw', 'mvnw', 'mvnw.cmd', 'pom.xml', and 'REDME.md' files.
- Editor View:** The file 'Pessoa.java' is open. The code is as follows:

```
1 package br.com.projeto.api.modelo;
2
3 public class Pessoa {
4
5     // Atributos
6     private String nome;
```
- Quick Fix Menu:** A context menu is open over the line 'private String nome;'. It includes options like 'Quick Fix...', 'More Actions...', and several generate options:
  - Generate Getter and Setter for 'nome'
  - Generate Getter for 'nome'
  - Generate Setter for 'nome'
  - Generate Constructors...
  - Add final modifier for 'nome'
- Bottom Status Bar:** Shows 'PROBLEMS 2', 'OUTPUT', and 'TERMINAL'.

**Selecione toda a linha do nome e click na lampada e escolha gerar os metodos getters e setters**



File Edit Selection View Go Run Terminal Help

Pessoa.java - APIS-SPRING-BOOT - Visual St



EXPLORER

APIS-SPRING-BOOT

> .vscode

< api

> .mvn

> .vscode

< src

< main

< java

< br

< com

< projeto

< api

< controle

< Controle.java

< modelo

< Pessoa.java

< ApiApplication.java

> resources

> test

> target

< .gitignore

< HELP.md

< mvnw

< mvnw.cmd

< pom.xml

< README.md

Controle.java

Pessoa.java X

PROBLEMS OUTPUT TERMINAL

```
api > src > main > java > br > com > projeto > api > modelo > Pessoa.java > ...
1  package br.com.projeto.api.modelo;
2
3  public class Pessoa {
4
5      // Atributos
6      private String nome;
7      private String idade;
8
9
10     // Get e Set
11    public String getNome() {
12        return nome;
13    }
14    public void setNome(String nome) {
15        this.nome = nome;
16    }
17    public String getIdade() {
18        return idade;
19    }
20    public void setIdade(String idade) {
21        this.idade = idade;
22    }
23
24
25
26 }
```

Repita o procedimento para idade

**Obs: quando gerei os metodos os metodos da idade ficaram por cima do nome  
dai como no original era o nome por cima eu coloquei  
Tente preservar a ordem dos attributos e coloque os getters e setters nessa  
mesma ordem**

File Edit Selection View Go Run Terminal Help

Pessoajava - APIS-SPRING-BOOT - Visual S

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- > api
- > .mvn
- > .vscode
- > src
- > main
- > java
- > br
- > com
- > projeto
- > api
- > controle
- > Controle.java
- > modelo
- > Pessoa.java
- > ApiApplication.java

Controle.java Pessoajava X

```
api > src > main > java > br > com > projeto > api > modelo > Pessoajava > ...
1 package br.com.projeto.api.modelo;
2
3 public class Pessoa {
4
5     // Atributos
6     private String nome;
7     private String idade;
8
9
10    // Get e Set
11    public String getNome() {
12        return nome;
13    }
14    public String getIdade() {
15        return idade;
16    }
17    public void setNome(String nome) {
18        this.nome = nome;
19    }
20    public void setIdade(String idade) {
21        this.idade = idade;
22    }
23
24 }
25
```

PROBLEMS OUTPUT TERMINAL

Colocados na ordem manualmente

#8

## Vinculando modelos e controles

**Devemos criar uma rota que retorne obrigatoriamente um nome e uma idade  
Pois tenho essas características no modelo pessoa**

File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle

Controle.java 3 Pessoajava

```
2
3     import org.springframework.web.bind.annotation.GetMapping;
4     import org.springframework.web.bind.annotation.PathVariable;
5     import org.springframework.web.bind.annotation.RestController;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14
15        @GetMapping("/boasVindas")
16        public String boasVindas(){
17            return "Seja bem vindo(a) ";
18        }
19
20        @GetMapping("/boasVindas/{nome}")
21        public String boasVindas(@PathVariable String nome){
22            return "Seja bem vindo(a) " + nome;
23        }
24
25        public Pes
26    }
```

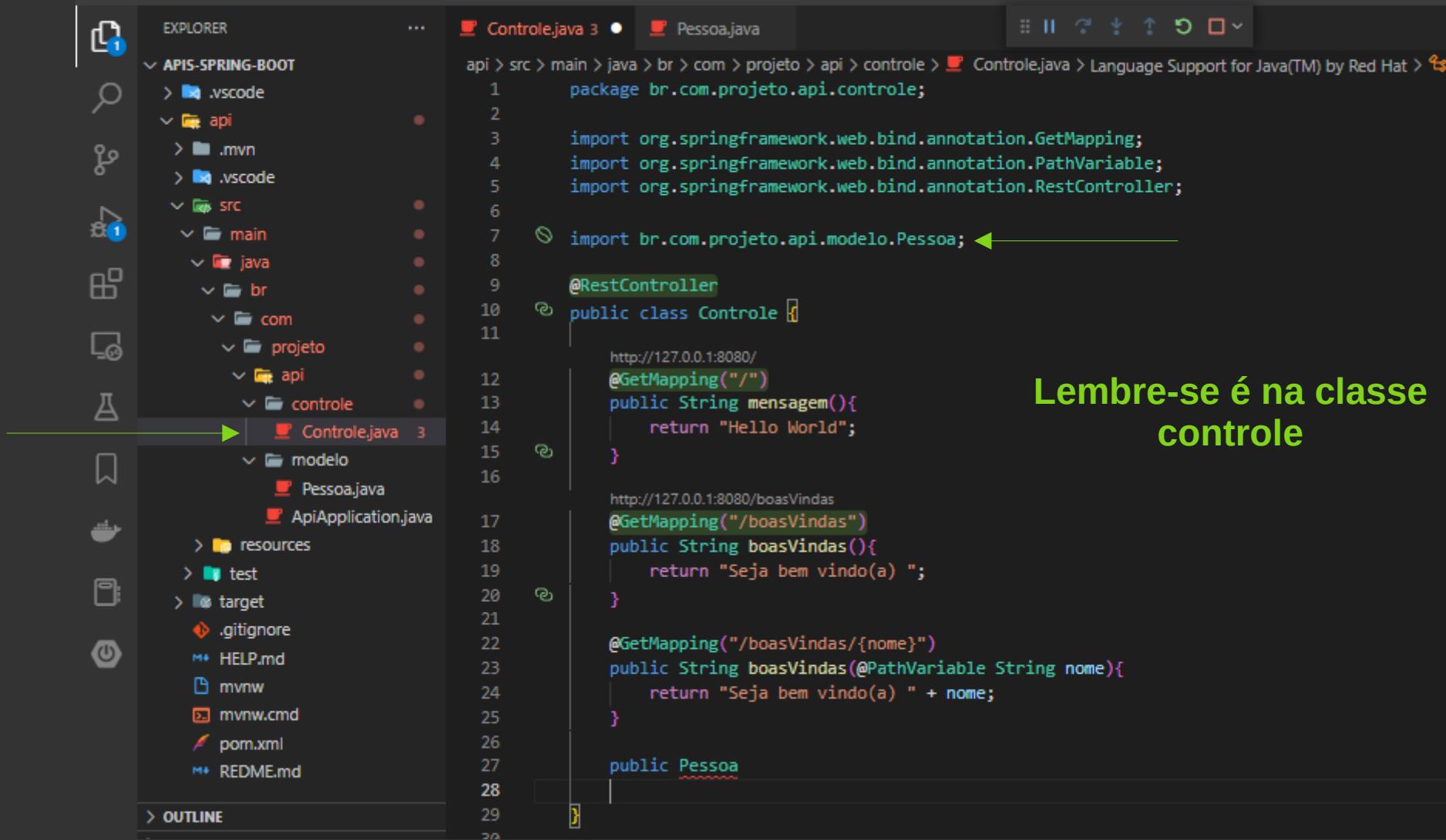
PROBLEMS 3 OUTPUT TERMINAL

Crie a rota e importe Pessoa e adicione seu importe

br.com.projeto.api.modelo.Pessoa

Pessoajava

linksResolver : Exposing 1 endpoint(  
TomcatWebServer : Tomcat started on port  
Application : Started ApiApplicati  
on in 1.686 seconds (proc  
PersistentValve - org.apache.catalina.valves  
INFO 5844 --- [ restarted main] .ConditionnevaluationueltaLoggingListener : Condition evaluation



# Lembre-se é na classe controle



File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio



EXPLORER

APIS-SPRING-BOOT

.vscode

api

.mvn

.vscode

src

main

java

br

com

projeto

api

controle

Controle.java 1

modelo

Pessoa.java

ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

README.md

Controle.java 1 X

Pessoa.java

Controle.java - APIS-SPRING-BOOT - Visual Studio

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java
      import org.springframework.web.bind.annotation.RestController;
      import br.com.projeto.api.modelo.Pessoa;
      @RestController
      public class Controle {
          http://127.0.0.1:8080/
          @GetMapping("/")
          public String mensagem(){
              return "Hello World";
          }
          http://127.0.0.1:8080/boasVindas
          @GetMapping("/boasVindas")
          public String boasVindas(){
              return "Seja bem vindo(a) ";
          }
          @GetMapping("/boasVindas/{nome}")
          public String boasVindas(@PathVariable String nome){
              return "Seja bem vindo(a) " + nome;
          }
          public Pessoa pessoa(){
          }
      }
```

A ideia é pegar o nome e a  
idade mais não via url  
devemos usar o requestbody

O nome da rota é pessoa

EXPLORER

- APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md
  - OUTLINE
  - TIMELINE
  - JAVA PROJECTS

Controle.java 3 ● Pessoa.java

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > pessoa()  
5     import org.springframework.web.bind.annotation.RestController;  
6  
7     import br.com.projeto.api.modelo.Pessoa;  
8  
9     @RestController  
10    public class Controle {  
11  
12        @GetMapping("/")  
13        public String mensagem(){  
14            return "Hello World";  
15        }  
16  
17        @GetMapping("/boasVindas")  
18        public String boasVindas(){  
19            return "Seja bem vindo(a) ";  
20        }  
21  
22        @GetMapping("/boasVindas/{nome}")  
23        public String boasVindas(@PathVariable String nome){  
24            return "Seja bem vindo(a) " + nome;  
25        }  
26  
27        public Pessoa pessoa(@Request){  
28            RequestAttribute - org.springframework.web.bind.a...  
29        }  
30  
31    }
```

## Adicione a anotation e seu import

org.springframework.web.bind.annotation.RequestAttribute x  
n.RequestAttribute

Annotation to bind a method parameter to a request attribute.

The main motivation is to provide convenient access to request attributes from a controller method with an optional/required check and a cast to the target method parameter type.

- Since:
  - 4.3
- Author:
  - Rossen Stoyanchev
- See Also:
  - RequestMapping
  - SessionAttribute

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
  - > .mvn
  - > .vscode
- < src
  - < main
    - < java
      - < br
        - < com
          - < projeto
            - < api
              - < controle
                - Controle.java 3
  - > resources
  - > test
  - > target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - README.md

Controle.java 3 • Pessoajava

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(

```
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.PathVariable;
5 import org.springframework.web.bind.annotation.RequestBody;
6 import org.springframework.web.bind.annotation.RestController;
7
8 import br.com.projeto.api.modelo.Pessoa;
9
10 @RestController
11 public class Controle {
12
13     http://127.0.0.1:8080/
14     @GetMapping("/")
15     public String mensagem(){
16         return "Hello World";
17     }
18
19     http://127.0.0.1:8080/boasVindas
20     @GetMapping("/boasVindas")
21     public String boasVindas(){
22         return "Seja bem vindo(a) ";
23     }
24
25     @GetMapping("/boasVindas/{nome}")
26     public String boasVindas(@PathVariable String nome){
27         return "Seja bem vindo(a) " + nome;
28     }
29
30     public Pessoa pessoa(@RequestBody){}
31 }
```

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- < controle
- Controle.java
- < modelo
- Pessoa.java
- ApiApplication.java
- < resources
- < test
- < target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

OUTLINE

```
    7
    8     import br.com.projeto.api.modelo.Pessoa;
    9
   10    @RestController
   11    public class Controle {
   12
   13        http://127.0.0.1:8080/
   14        @GetMapping("/")
   15        public String mensagem(){
   16            return "Hello World";
   17
   18        http://127.0.0.1:8080/boasVindas
   19        @GetMapping("/boasVindas")
   20        public String boasVindas(){
   21            return "Seja bem vindo(a) ";
   22
   23        @GetMapping("/boasVindas/{nome}")
   24        public String boasVindas(@PathVariable String nome){
   25            return "Seja bem vindo(a) " + nome;
   26
   27
   28        public Pessoa pessoa(@RequestBody Pessoa p){
   29            return p;
   30
   31
   32
   33    }
```

Pessoa é o tipo de dado que espero

Nome desse objeto do tipo Pessoa  
p

O que retornamos

**Quando usamos o @RequestBody não podemos usar o @GetMapping  
Pois ele não tem suporte ao @RequestBody**

Dai usamo o  
**@PostMapping**

**Lembrando que todos os verbos http tem seus POST GET DELETE UPDATE ETC...**

File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code jakarta.annotation.PostConstruct

EXPLORER ... Controle.java 1 Pessoajava

APIS-SPRING-BOOT .vscode api .mvn .vscode src main java br com projeto api controle Controle.java Language Support

```
    7
    8     import br.com.projeto.api.modelo.Pessoa;
    9
   10    @RestController
   11    public class Controle {
   12
   13        http://127.0.0.1:8080/
   14        @GetMapping("/")
   15        public String mensagem(){
   16            return "Hello World";
   17
   18        http://127.0.0.1:8080/boasVindas
   19        @GetMapping("/boasVindas")
   20        public String boasVindas(){
   21            return "Seja bem vindo(a) ";
   22
   23        @GetMapping("/boasVindas/{nome}")
   24        public String boasVindas(@PathVariable String nome){
   25            return "Seja bem vindo(a) " + nome;
   26
   27
   28        @Post
   29        public void PostConstruct - jakarta.annotation
   30
   31
   32
   33
   34
   35 }
```

PROBLEMS 1 OUTPUT

TERMINAL 2023-01-22T12:43:17.507+00:00 [INFO] 3044 --- [ restartedMain] br.com.projeto.apispringbootApplication : Started ApispringbootApplication in 1.861 seconds (process running for 3094.247)

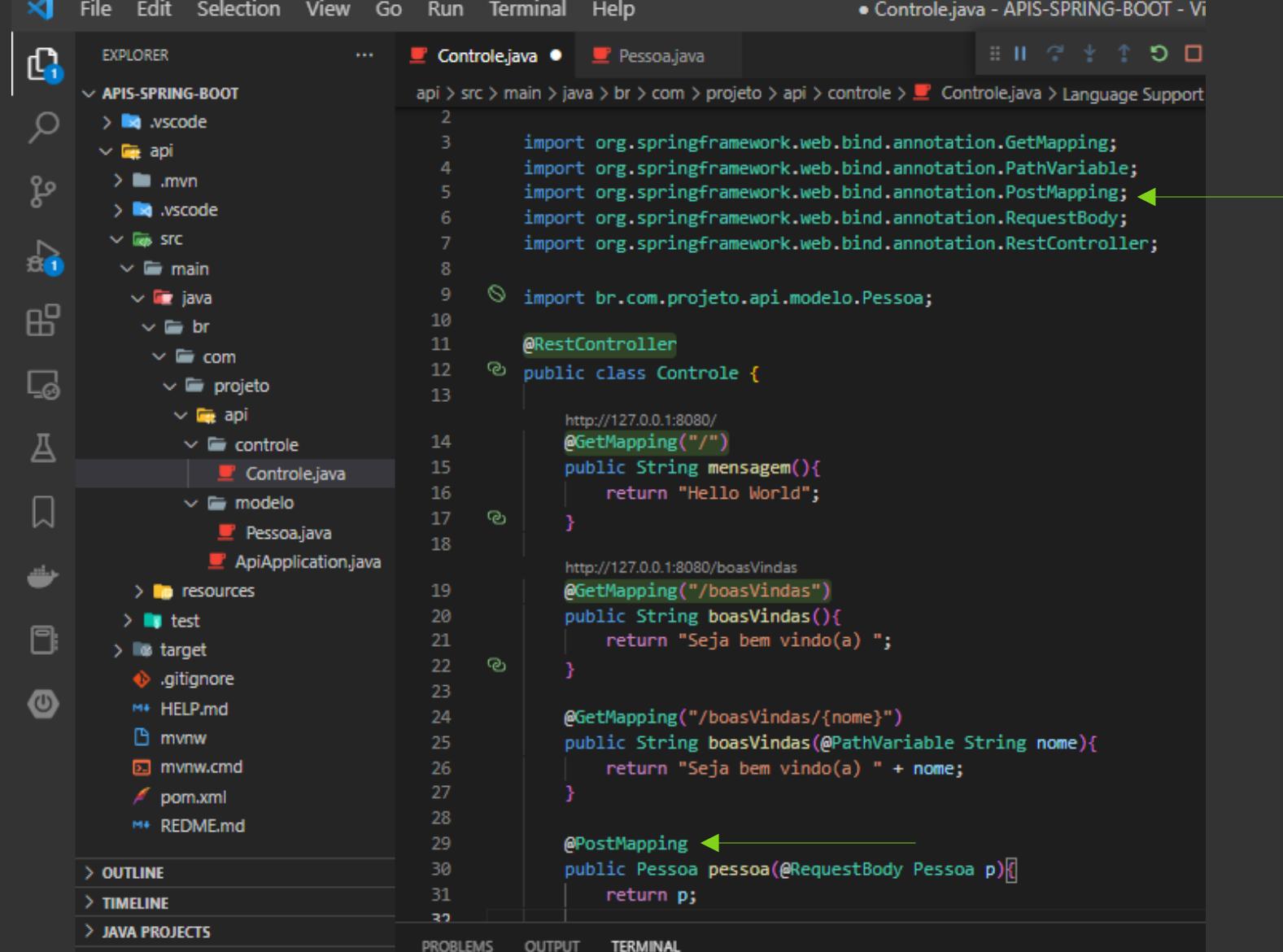
# Adicionamos O post e seu import

The `PostConstruct` annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization. This method must be invoked before the class is put into service. This annotation must be supported on all classes that support dependency injection. The method annotated with `PostConstruct` must be invoked even if the class does not request any resources to be injected. Only one method in a given class can be annotated with this annotation. The method on which the `PostConstruct` annotation is applied must fulfill all of the following criteria:

- The method must not have any parameters except in the case of interceptors in which case it takes an `InvocationContext` object as defined by the Jakarta Interceptors specification.
- The method defined on an interceptor class or superclass of an interceptor class must have one of the following signatures:
  - `void (InvocationContext)`
  - `Object (InvocationContext) throws Exception`

Note: A `PostConstruct` interceptor method must not throw application exceptions, but it may be declared to throw checked exceptions including the `java.lang.Exception` if the same interceptor method interposes on business or timeout methods in addition to lifecycle events. If a `PostConstruct` interceptor method returns a value, it is ignored by the container.

- The method defined on a non-interceptor class must have the following signature:



The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure under "APIS-SPRING-BOOT".
  - src folder contains:
    - main folder contains:
      - java folder contains:
        - br folder contains:
          - com folder contains:
            - projeto folder contains:
              - api folder contains:
                - controle folder contains: Controle.java
  - CONTROLE.JAVA:** The current file being edited.

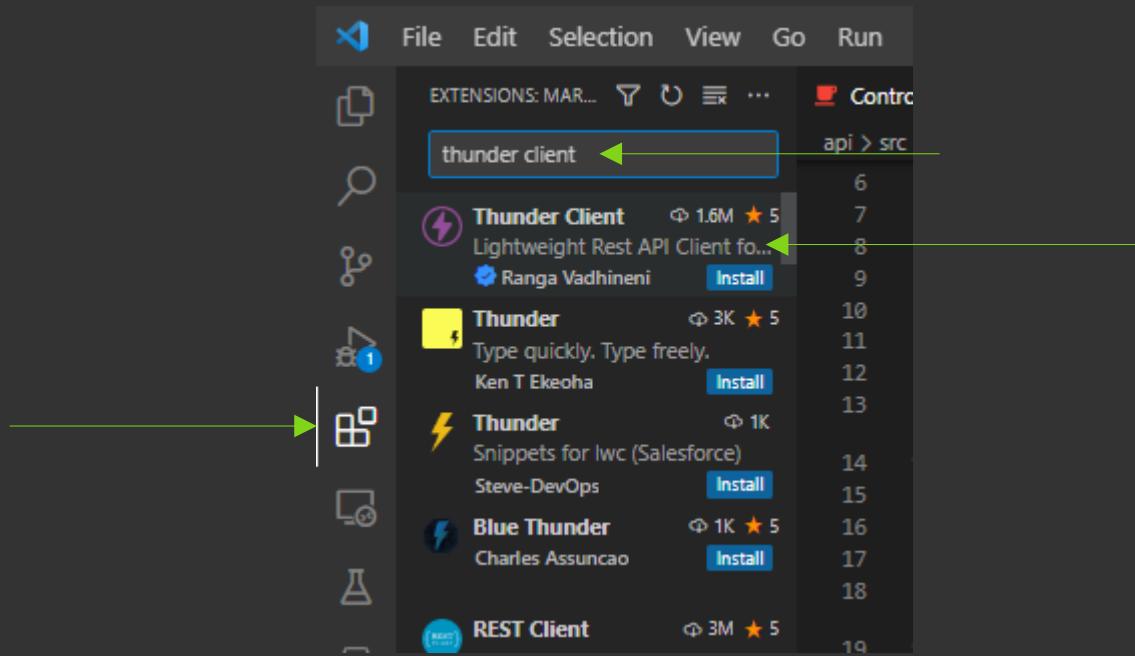
```
6 import org.springframework.web.bind.annotation.RequestBody;
7 import org.springframework.web.bind.annotation.RestController;
8
9 import br.com.projeto.api.modelo.Pessoa;
10
11 @RestController
12 public class Controle {
13
14     @GetMapping("/")
15     public String mensagem(){
16         return "Hello World";
17     }
18
19     @GetMapping("/boasVindas")
20     public String boasVindas(){
21         return "Seja bem vindo(a) ";
22     }
23
24     @GetMapping("/boasVindas/{nome}")
25     public String boasVindas(@PathVariable String nome){
26         return "Seja bem vindo(a) " + nome;
27     }
28
29     @PostMapping("/pessoa")
30     public Pessoa pessoa(@RequestBody Pessoa p){
31         return p;
32     }
33
34 }
```
  - OUTPUT:** Shows the status message "No output has been produced yet."
  - TERMINAL:** Shows the command "code ."

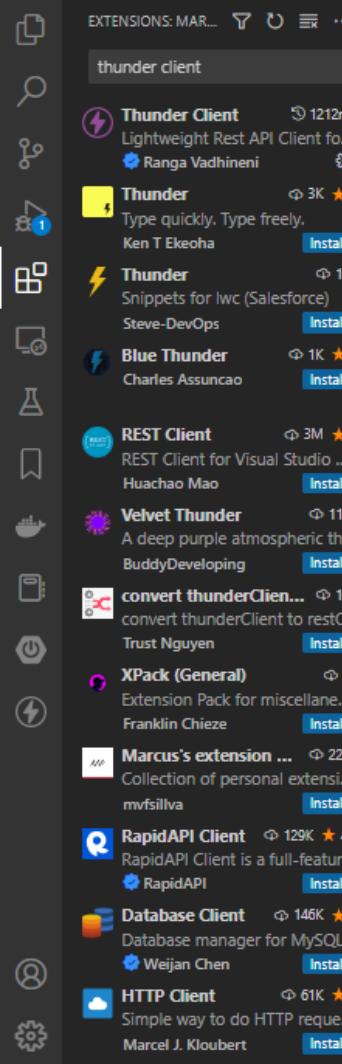
Por fim criamos a rota  
/pessoa

#9  
**Testando o projeto com thunder**

**Quando utilizamos o @PostMapping não podemos utilizar o navegador pois ele não tem suporte precisamos baixar uma extenção por isso vamos baixar o thunder client**

**Se não quisermos poderemos usar o postman ou o insomnia**





Extension: Thunder Client X Pessoajava

# Thunder Client v2.3.4

Ranga Vadhineni | 1,637,530 | ★★★★★(155) | Sponsor

Lightweight Rest API Client for VS Code

Disable Uninstall ⚙️

This extension is enabled globally.

## Pronto instalamos

### Details Feature Contributions Changelog Runtime Status

## Thunder Client

Thunder Client is a lightweight Rest API Client Extension for Visual Studio Code, hand-crafted by [Ranga Vadhineni](#) with simple and clean design.

- Voted as #10 Product of the day on Product Hunt
- Website - [www.thunderclient.com](http://www.thunderclient.com)
- Follow Twitter for updates - [twitter.com/thunder\\_client](https://twitter.com/thunder_client)
- Support: [github.com/rangav/thunder-client-support](https://github.com/rangav/thunder-client-support)

### Story behind Thunder Client

- Read Launch Blog Post on [Medium](#)

### Usage

- Install the Extension, Click Thunder Client icon on the Action Bar.
- From Sidebar click [New Request](#) button to test API

### TERMINAL

```
2023-03-00 INFO 5844 --- [ restartedMain] br.com.projeto.api.ApiApplication      : Started ApiApplication (process running for 4041.403)
2023-03-00 INFO 5844 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation
```

### Categories

Programming Languages Snippets

Testing

### Extension Resources

Marketplace Repository License Ranga Vadhineni

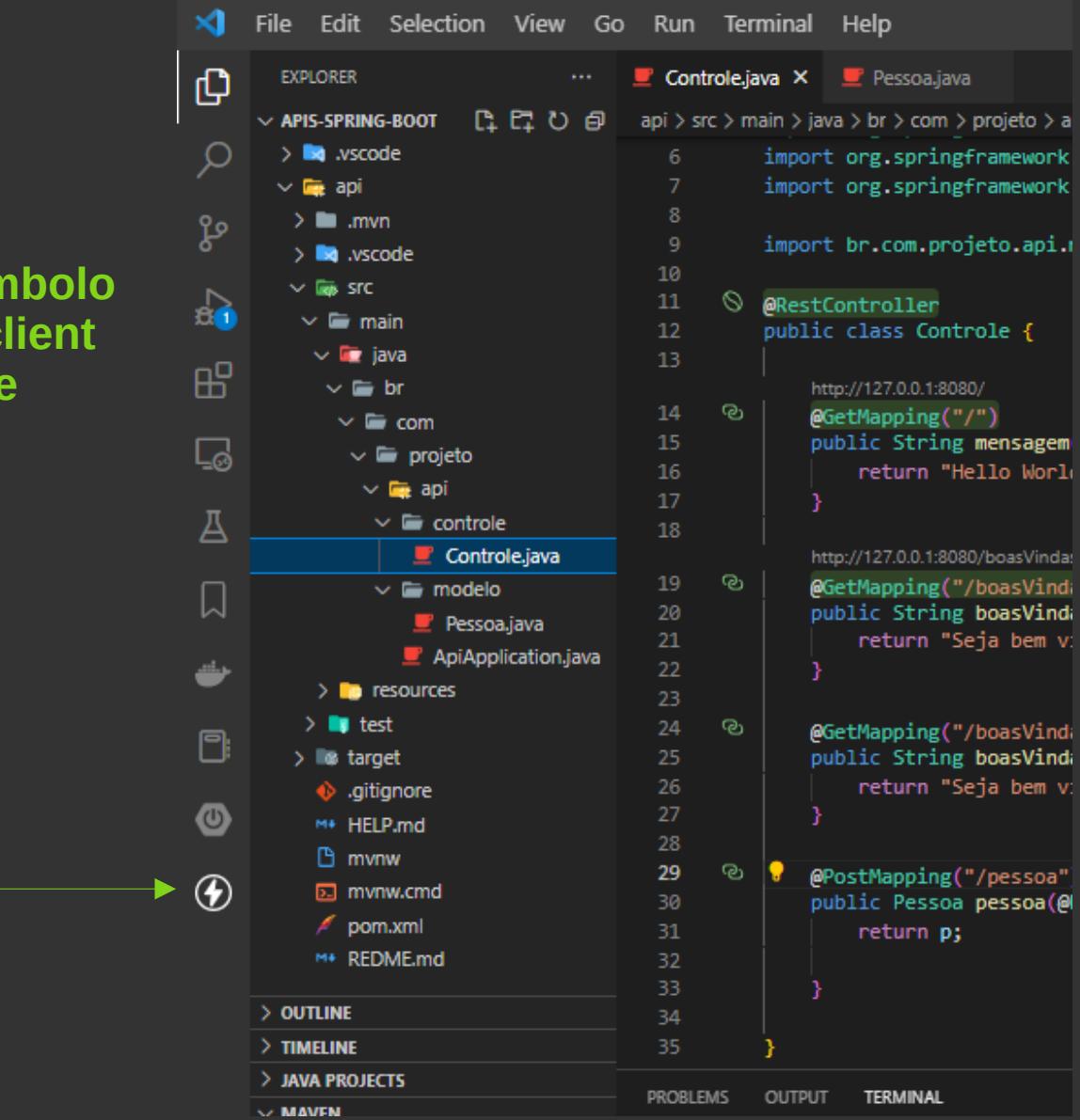
### More Info

Published 3/30/2021, 09:59:45  
Last released 1/22/2023, 05:38:57  
Identifier rangav.vscode-thunder-client

DEBUG CONSOLE

powershell  
Run: ApiAp...

Repare no simbolo  
do thunder client  
Click nele

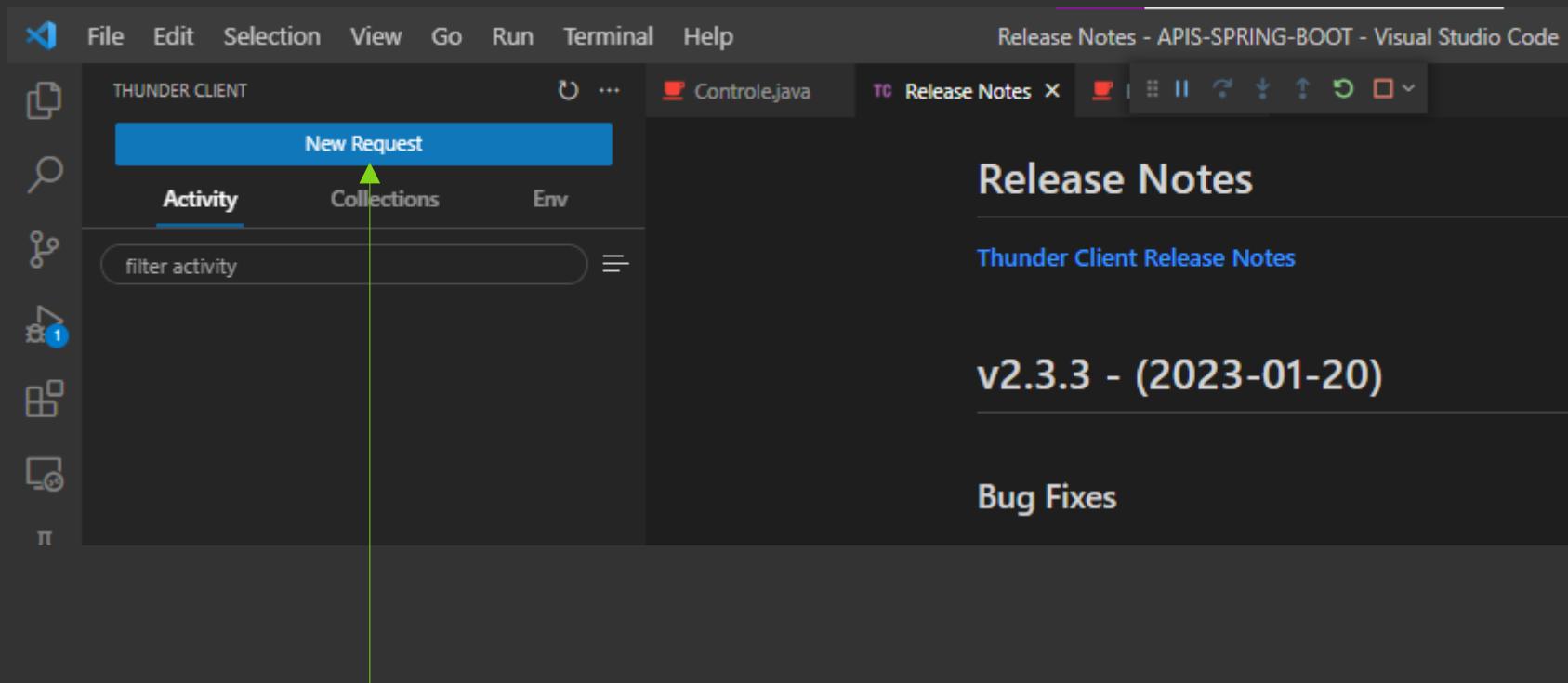


The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
  - src folder contains main, java, br, com, projeto, api, controle, modelo, and resources.
  - java folder contains Controle.java, Pessoa.java, and ApiApplication.java.
  - resources folder contains application.properties.
  - test folder.
  - target folder.
  - gitignore file.
  - HELP.md file.
  - mvnw file.
  - mvnw.cmd file.
  - pom.xml file.
  - README.md file.
- Editor:** The file "Controle.java" is open, showing Java code for a Spring Controller.

```
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.PostMapping;
8 import br.com.projeto.api.modelo.Pessoa;
9 import br.com.projeto.api.servico.ServicoPessoa;
10
11 @RestController
12 public class Controle {
13
14     @GetMapping("/")
15     public String mensagem() {
16         return "Hello World";
17     }
18
19     @GetMapping("/boasVindas")
20     public String boasVindas() {
21         return "Seja bem vindo!";
22     }
23
24     @GetMapping("/boasVindas")
25     public String boasVindas() {
26         return "Seja bem vindo!";
27     }
28
29     @PostMapping("/pessoa")
30     public Pessoa pessoa(@RequestBody Pessoa p) {
31         return p;
32     }
33 }
```
- Status Bar:** Shows the status "APIs-Spring-Boot" and the number "1" in the status bar.

# Click em new request



File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env filter activity

Controle.java Release Notes

GET https://www.thunderclient.com/welcome Send

Status: Size: Time:

Query Headers 2 Auth Body Tests Pre Run New Response Headers Cookies Results Docs

Query Parameters

parameter value

Welcome to Thunder Client

Your activity will appear here...

Send Request Ctrl + Enter

Import Curl Ctrl + U

Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

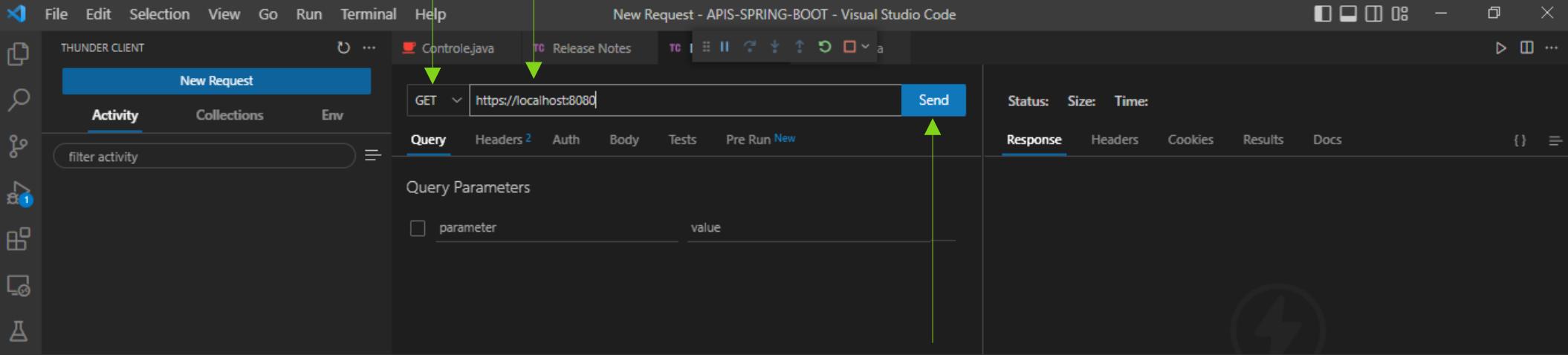
TERMINAL

```
2023-01-22T12:59:05.123-03:00 INFO 5844 --- [ restartedMain] br.com.projeto.api.ApiApplication      : Started ApiApplication in 0.798 seconds (process running for 4041.403)
2023-01-22T12:59:05.139-03:00 INFO 5844 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
```

DEBUG CONSOLE

powershell Run: ApiAp...

31°C Parc ens... 13:09

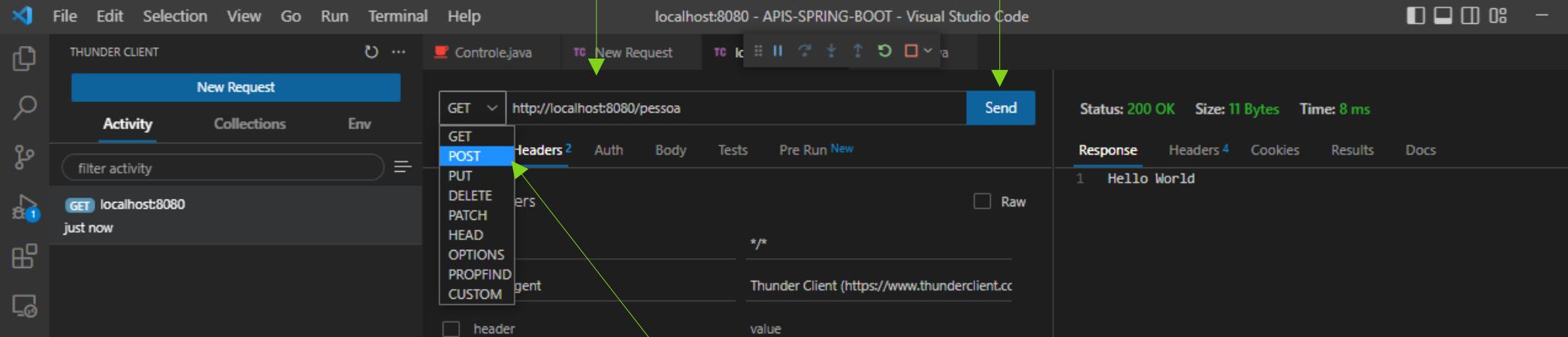


**Fazemos um teste na rota hello world digite a url com o metodo get e click em send**

A screenshot of the Thunder Client extension in Visual Studio Code. The interface shows a request being sent to `http://localhost:8080/`. The response status is `200 OK`, size is `11 Bytes`, and time is `8 ms`. The response body contains the text `Hello World`.

The Thunder Client sidebar shows a recent activity: a `GET` request to `localhost:8080` just now.

Como deu erro copie e coleia a própria url do navegador dai deu certo



Para acessar a rota /pessoa troque o metodo para POST cole a url e de send

File Edit Selection View Go Run Terminal Help localhost:8080 - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080 just now

Controle.java New Request

POST http://localhost:8080/pessoa Send

Headers 2 Query Auth Body Tests Pre Run New

Http Headers Raw

Accept \*/\*

User-Agent Thunder Client (https://www.thunderclient.cc)

header value

Status: 400 Bad Request Size: 5.83 KB Time: 57 ms

Response Headers 4 Cookies Results Docs

```
1 {
2   "timestamp": "2023-01-22T16:16:37.356+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "trace": "org.springframework.http.converter.HttpMessageNotReadableException: Required request body is missing: public br.com.projeto.api.modelo.Pessoa br.com.projeto.api.controle.Controle.pessoa(br.com.projeto.api.modelo.Pessoa)\r\n\r\nat org.springframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodProcessor.readWithMessageConverters(RequestResponseBodyMethodProcessor.java:166)\r\n\r\nat org.springframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodProcessor.resolveArgument(RequestResponseBodyMethodProcessor.java:136)\r\n\r\nat org.springframework.web.method.support.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:122)\r\n\r\nat org.springframework.web.method.support.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:181)\r\n\r\nat org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:148)\r\n\r\nat org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:117)\r\n\r\nat org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:884)\r\n\r\nat org.springframework.web.servlet.mvc.method.annotation
```

Deu esse erro pois essa rota espera uma informação via body

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
ted initialization in 1 ms
2023-01-22T13:16:37.351-03:00  WARN 5844 --- [nio-8080-exec-8] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.http.converter.HttpMessageNotReadableException: Required request body is missing: public br.com.projeto.api.modelo.Pessoa br.com.projeto.api.controle.Controle.pessoa(br.com.projeto.api.modelo.Pessoa)]
```

DEBUG CONSOLE

powershell Run: ApiApp...

31°C Parc ens... 13:16

# Click em body

Lembre-se que json não é o único formato temos xml entre outros mais o mais utilizado é jason veja qual eles estão pedindo nesse caso é json e na grande maioria sera

A screenshot of the Thunder Client extension in Visual Studio Code. The interface shows a 'New Request' dialog with the following details:

- Method: POST
- Endpoint: http://localhost:8080/pessoa
- Body tab selected
- Content type: Json
- JSON Content:

```
1 {  
2     "nome": "Cristiano",  
3     "idade": 40  
4 }
```

The 'Send' button is highlighted with a blue arrow pointing towards it. To the right, the response area is visible with tabs for Response, Headers, Cookies, Results, and Docs. A large green arrow points from the bottom text towards the 'Send' button.

No body crie seu objeto json  
E depois click em send

File Edit Selection View Go Run Terminal Help localhost:8080 - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ApiApplication.java localhost:8080

New Request

Activity Collections Env

filter activity

POST localhost:8080 just now

POST http://localhost:8080/pessoa Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "idade": 40  
4 }
```

Status: 200 OK Size: 33 Bytes Time: 187 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "nome": "Cristiano",  
3   "idade": "40"  
4 }
```

Conseguimos passar o objeto via body  
Desse jeito os dados não são passados  
pela url e você pode até passar senhas

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
(s): 8080 (http) with context path ''  
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication  
in 18.894 seconds (process running for 22.481)  
2023-01-22T13:41:52.193-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]  
spatcherServlet 'dispatcherServlet'  
2023-01-22T13:41:52.194-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet  
dispatcherServlet:  
2023-01-22T13:41:52.200-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet  
on in 4 ms
```

DEBUG CONSOLE

: Started ApiApplication  
: Initializing Spring Di  
: Initializing Servlet '  
: Completed initializati

powershell Run: ApiAp...

31°C Parc ens... 13:45

#10  
**Criando a base de dados**

**Nesse passo podemos usar o mysql ou qualquer outro tipo para criar a base de dados**  
**Nesse caso usarei uma extensão para mysql no próprio vscode**

File Edit Selection View Go Run Terminal Help

Extension: MySQL - APIS-SPRING-BOOT - Visual Studio Code

EXTENSIONS: MARKETPLACE

mysql

- MySQL** v1.1M ★ 3.5  
MySQL management tool  
Jun Han [Install](#)
- MySQL** v767K ★ 4  
Database manager for MySQL/Maria...  
WeiJian Chen [Install](#)
- MySQL Syntax** v430K ★ 3  
MySQL syntax highlighting support  
Jake Bathman [Install](#)
- SQLTools MySQL/Mari...** v519K ★ 5  
SQLTools MySQL/MariaDB  
Matheus Teixeira [Install](#)
- mysql-inline-decorator** v108K ★ 3  
Add color coding to inline MySQL st...  
odubuc [Install](#)
- MySQL Autocomplete** v24K ★ 5  
MySQL Sintax Autocomplete for Vis...  
nospinozacr [Install](#)
- MySQL Statement Scr...** v41K ★ 5  
Easy mysql statement running with s...  
Jared Black [Install](#)
- ES7 JavaScript/Node/...** v38K ★ 5  
Simple extension for Node, javascript...  
abrahamwilliam007 [Install](#)
- SQLite & MySQL Snipp...** v27K ★ 5  
A snippet for MySQL and SQLite, to ...  
Rohit Chouhan [Install](#)
- MySQL Tool** v2K ★ 4.5

Extension: MySQL X

MySQL v6.1.3

Weijan Chen | 767,140 | ★★★★☆(191)

Database manager for MySQL/MariaDB, PostgreSQL, SQLite, Redis and ElasticSearch.

[Install](#) [⚙️](#)

[Details](#) [Feature Contributions](#) [Changelog](#)

## Database Client

This project is no longer maintain for free and requires payment to unlock all features. At the same time I will put more effort into developing this project.

The online document is migrate to [doc.database-client.com](#).

vscode marketplace v6.1.3 installs 767k stars 2k rating 4.1/5 (191) license MIT

This project is a database client for VSCode, supports manager MySQL/MariaDB, PostgreSQL, SQLite, Redis, ClickHouse, 达梦, and ElasticSearch, and works as an SSH client, boost your maximum productivity!

Project site: [vscode-database-client](#), 中文文档

PROBLEMS OUTPUT TERMINAL

TERMINAL

DEBUG CONSOLE

Categories

Programming Languages Keymaps

Extension Resources

Marketplace Repository License Weijan Chen

More Info

Published 5/5/2020, 05:03:00  
Last released 1/13/2023, 19:54:23  
Identifier cweijan.vscode-mysql-client?

Va em extensões e instale essa extensão para o mysql

EXTENSIONS: MARKETPLACE

mysql

- MySQL** v1.1M ★ 3.5  
MySQL management tool  
Jun Han [Install](#)
- MySQL** Database manager for MySQL/Maria...  
Weijan Chen [Install](#)
- MySQL Syntax** v430K ★ 3  
MySQL syntax highlighting support  
Jake Bathman [Install](#)
- SQLTools MySQL/Mari...** v519K ★ 5  
SQLTools MySQL/MariaDB  
Matheus Teixeira [Install](#)
- mysql-inline-decorator** v108K ★ 3  
Add color coding to inline MySQL st...  
odubuc [Install](#)
- MySQL Autocomplete** v24K ★ 5  
MySQL Syntax Autocomplete for Vis...  
nospinozacr [Install](#)
- MySQL Statement Scr...** v41K ★ 5  
Easy mysql statement running with s...  
Jared Black [Install](#)
- ES7 JavaScript/Node/...** v38K ★ 5  
Simple extension for Node, javascript...  
abrahamwilliam007 [Install](#)
- SQLite & MySQL Snipp...** v27K ★ 5  
A snippet for MySQL and SQLite, to ...  
Rohit Chouhan [Install](#)
- MySQL Tool** v2K ★ 4.5  
MySQL management tool is a unifie...  
wycliffe Pepela [Install](#)
- MySQL Inline Decorator...** v1K ★ 5  
Ilyas Akin [Install](#)
- Go MySQL Highlighter** v5K ★ 5  
Syntax highlights MySQL strings in G...  
Stirling Marketing Group [Install](#)

Extension: MySQL X



**MySQL** v6.1.3

Weijan Chen | 767,140 | ★★★★★(191)

Database manager for MySQL/MariaDB, PostgreSQL, SQLite, Redis and ElasticSearch.

[Disable](#) [Uninstall](#) [Settings](#)

This extension is enabled globally.

[Details](#) [Feature Contributions](#) [Changelog](#)

## Database Client

This project is no longer maintain for free and requires payment to unlock all features. At the same time I will put more effort into developing this project.

The online document is migrate to [doc.database-client.com](http://doc.database-client.com).

vscode marketplace v6.1.3 installs 767k stars 2k rating 4.1/5 (191) license MIT

This project is a database client for VSCode, supports manager MySQL/MariaDB, PostgreSQL, SQLite, Redis, ClickHouse, and ElasticSearch, and works as an SSH client, boost your maximum productivity!

PROBLEMS OUTPUT TERMINAL

TERMINAL

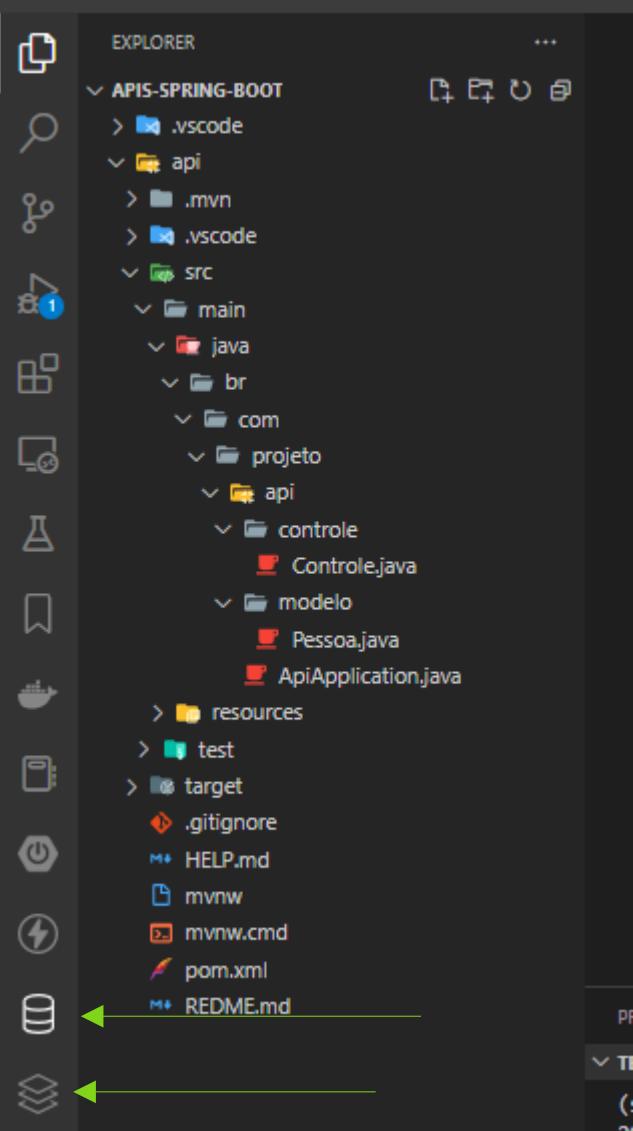
```
(s): 8080 (http) with context path ""
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApp...
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sp...
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Se...
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initi...
```

DEBUG CONSOLE

powershell

Run: ApiAp...

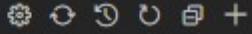
File Edit Selection View Go Run Terminal



Ela cria 2 atalhos click no database



DATABASE



You haven't created any connections [learn more.](#)



Create Connection



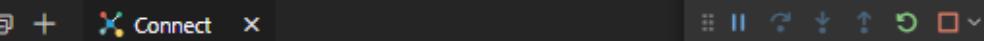
PROBLE

TERMIN

(s): 8







Deixei os dados como  
estão só mudei a senha  
Depois click em connect



## Connect to server

Name

Connection Name

Connection Scope

Global

Premium Only  
Workspace

Group

Group

Connect Timeout

5000

Read Only

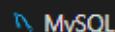
Save Password

Forever

▼

Note: This extension needs to be [pay](#) to unlock all features.

### Server Type



MySQL



MariaDB



PostgreSQL



SQL Server



Oracle



SQLite



DM8



ClickHouse



SSH



Docker



Redis



ElasticSearch



MongoDB



S3/Minio



FTP



Neo4j



JDBC(Beta)

\* Host

127.0.0.1

\* Port

3306

\* Username

root

\* Password



Database

Connection database

Advance Option

SSH Tunnel



Use SSL



Socket Path

e.g. /var/run/mysql/mysqld.sock

Use



Save



Connect



Close

PROBLEMS OUTPUT TERMINAL

TERMINAL

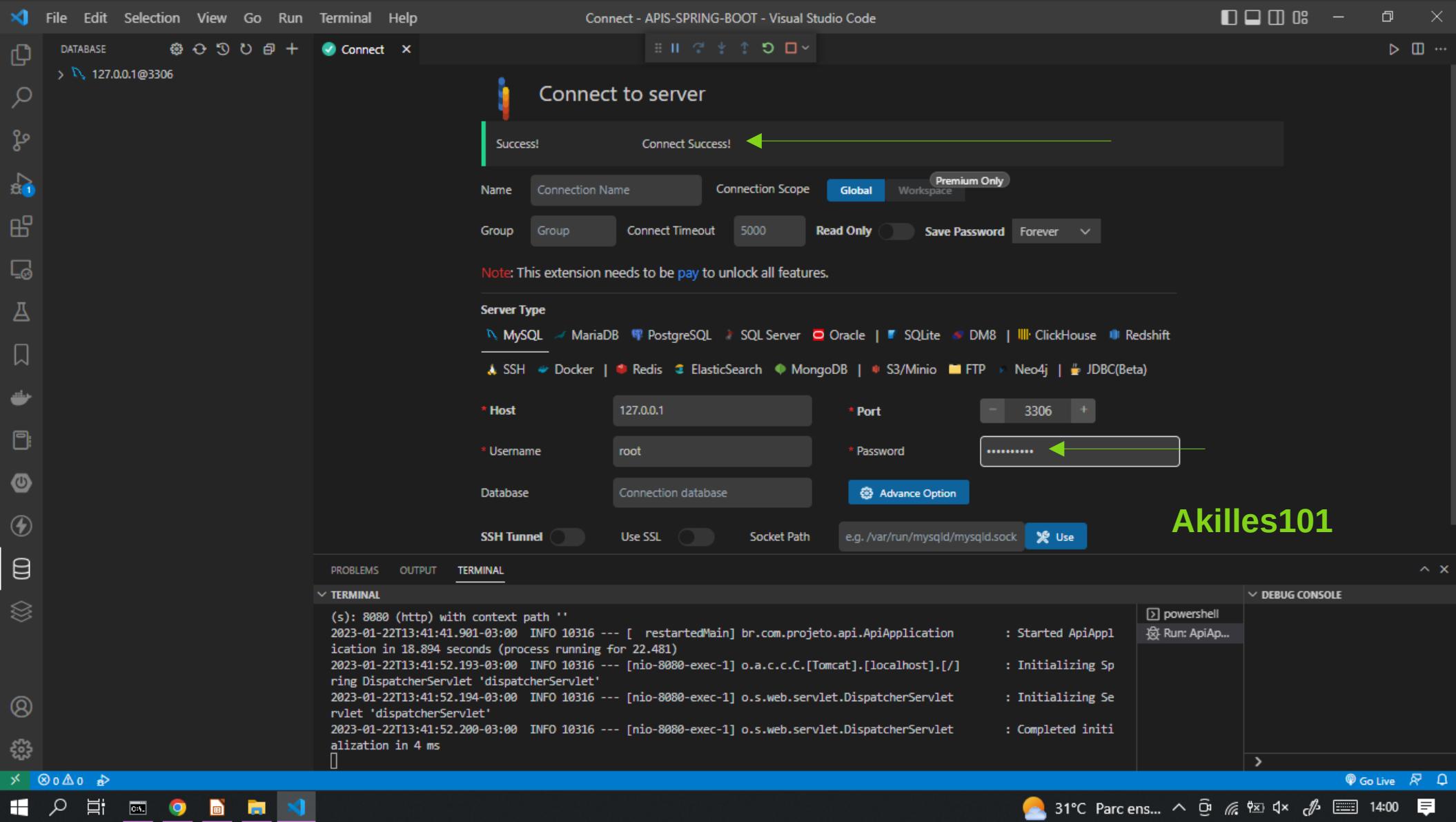
(s): 8080 (http) with context path ''

Akilles101

A senha é a verdadeira do mysql

DEBUG CONSOLE

powershell



Aqui estão as bases de dados do mysql

File Edit Selection View Go Run Terminal Help Connect - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 + -

Connect to server

Success! Connect Success!

Name Connection Name Connection Scope Global Premium Only

Group Group Connect Timeout 5000 Read Only Save Password Forever

Note: This extension needs to be pay to unlock all features.

Server Type

MySQL MariaDB PostgreSQL SQL Server Oracle SQLite DM8 ClickHouse Redshift

SSH Docker Redis ElasticSearch MongoDB S3/Minio FTP Neo4j JDBC(Beta)

\* Host 127.0.0.1 \* Port 3306

\* Username root \* Password

Database Connection database Advance Option

SSH Tunnel Use SSL Socket Path e.g. /var/run/mysql/mysqld.sock Use

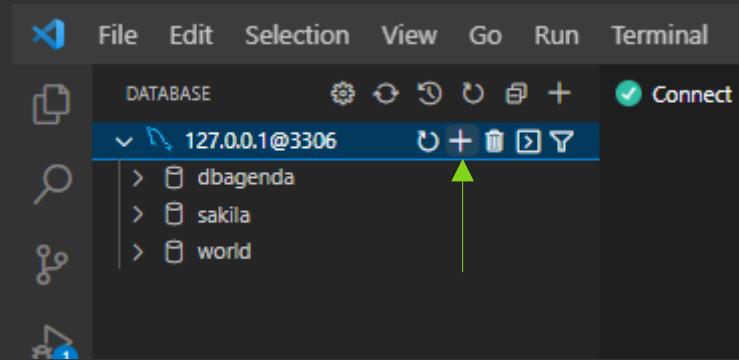
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

TERMINAL

```
(s): 8080 (http) with context path ""
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiAppl
ication in 18.894 seconds (process running for 22.481)
2023-01-22T13:41:52.193-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sp
ring DispatcherServlet 'dispatcherServlet'
2023-01-22T13:41:52.194-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Se
rvlet 'dispatcherServlet'
2023-01-22T13:41:52.200-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initi
alization in 4 ms
[]
```

powershell Run: ApiAp...

31°C Parc ens... Go Live 14:01



**Click no sinal de + para criar o banco de dados**

The screenshot shows the MySQL Workbench application window. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the current file is "create-db-template.sql - APIS-SPRING-BOOT - Visual Studio". The left sidebar has icons for Database, Schema, Table, View, Function, and Procedure. The Database section shows a connection to "127.0.0.1@3306" with three databases listed: dbagenda, sakila, and world. The main pane displays the contents of the "create-db-template.sql" file:

```
C: > Users > brito > AppData > Roaming > Code > User > globalStorage > cweijan.vscode-mysql-client2 > 16
    ⚡ Active Connection
    ↴ 1   -- Active: 1674406792944@127.0.0.1@3306 MySQL
        ▷ Execute
    2   CREATE DATABASE |
    3   |   DEFAULT CHARACTER SET = 'utf8mb4';
```

The screenshot shows the MySQL Workbench application. On the left, there's a sidebar with icons for File, Edit, Selection, View, Go, Run, Terminal, and Help. Below that is a database browser pane showing a connection to '127.0.0.1@3306' with databases 'dbagenda', 'sakila', and 'world'. The main area is a terminal window titled 'create-db-template.sql' which contains the following SQL code:

```
1 Active Connection
2 Active: 1674406792944@127.0.0.1@3306 MySQL
3 CREATE DATABASE api_spring
4     DEFAULT CHARACTER SET = 'utf8mb4';
```

**Depois de dar nome api\_spring click em execute**

File Edit Selection View Go Run Terminal Help • create-db-template.sql - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25 Connect create-db-template.sql

C:\> Users > brito > AppData > Roaming > Code > User > globalStorage > cweijan.vscode-mysql-client2 > 1674406792944@127.0.0.1@3306 > create-db-template.sql > ...

Active Connection  
1 Active: 1674406792944@127.0.0.1@3306 MySQL  
Execute  
2 CREATE DATABASE api\_spring  
3 DEFAULT CHARACTER SET = 'utf8mb4';

Data

CREATE DATABASE api\_spring  
DEFAULT CHARACTER SET = 'utf8mb4'

+ Input to filter result Free 1 Cost: 536ms < >

CREATE DATABASE api\_spring DEFAULT CHARACTER SET = 'utf8mb4'

AffectedRows : 1

Veja que o banco de dados foi criado

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
(s): 8080 (http) with context path ""
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication      : Started ApiAppl
ication in 18.894 seconds (process running for 22.481)
2023-01-22T13:41:52.193-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]      : Initializing Sp
ring DispatcherServlet 'dispatcherServlet'
2023-01-22T13:41:52.194-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      : Initializing Se
rvlet 'dispatcherServlet'
2023-01-22T13:41:52.200-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      : Completed initi
alization in 4 ms
```

DEBUG CONSOLE

powershell  
Run: ApiAp...

Ln 3, Col 39 Spaces: 4 UTF-8 CRLF SQL Go Live

31°C Parc ens... 14:05

**#11**  
**IMPLEMENTANDO DEPENDENCIAS**

**Feche o servidor mysql e o servidor  
Quando o servidor está funcionando posso atualizar ou baixar dependencias? não**

EXPLORER

- APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
    - resources
      - static
      - templates
      - application.properties
    - test
    - target
      - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - README.md

ApiApplication.java X

```
api > src > main > java > br > com > projeto > api > ApiApplication.java > ...
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ApiApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(primarySource: ApiApplication.class, args);
11     }
12 }
13
14 }
```

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
on port(s): 8080 (http) with context path ''
2023-01-23T11:16:20.014-03:00  INFO 11124 --- [ restartedMain] br.com.projeto.api.ApiApplication
ication in 24.163 seconds (process running for 28.408)
```

```
ApiApplication.java - APIS-SPRING-BOOT - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ApiApplication.java X
api > src > main > java > br > com > projeto > api > ApiApplication.java > ...
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ApiApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(primarySource: ApiApplication.class, args);
11     }
12 }
13
14 }
```

Qualquer uma das 2 opções finaliza o servidor

EXPLORER

APIS-SPRING-BOOT

- .vscode
- src
  - main
  - java
    - br
      - com
      - projeto
      - api
        - controle
          - Controle.java
        - modelo
          - Pessoa.java
        - ApiApplication.java
  - resources
    - static
    - templates
    - application.properties
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md

api > 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>3.0.2</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>br.com.projeto</groupId>
  <artifactId>api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>api</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>

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

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

Todas as dependencias ficam no arquivo pom.xml

PROBLEMS OUTPUT TERMINAL

OUTLINE

TIMELINE

JAVA PROJECTS

MAVEN

api br.com.projeto:api

port(s): 8080 (http) with context path ''  
2023-01-23T11:23:14.611-03:00 INFO 6540 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApplica  
tion in 11.71 seconds (process running for 14.239)  
2023-01-23T11:23:15.963-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sprin  
g DispatcherServlet 'dispatcherServlet'  
2023-01-23T11:23:16.063-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializing Servl  
et 'dispatcherServlet'  
2023-01-23T11:23:16.087-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed initiali  
zation in 2 ms  
PS E:\APIS-SPRING-BOOT>

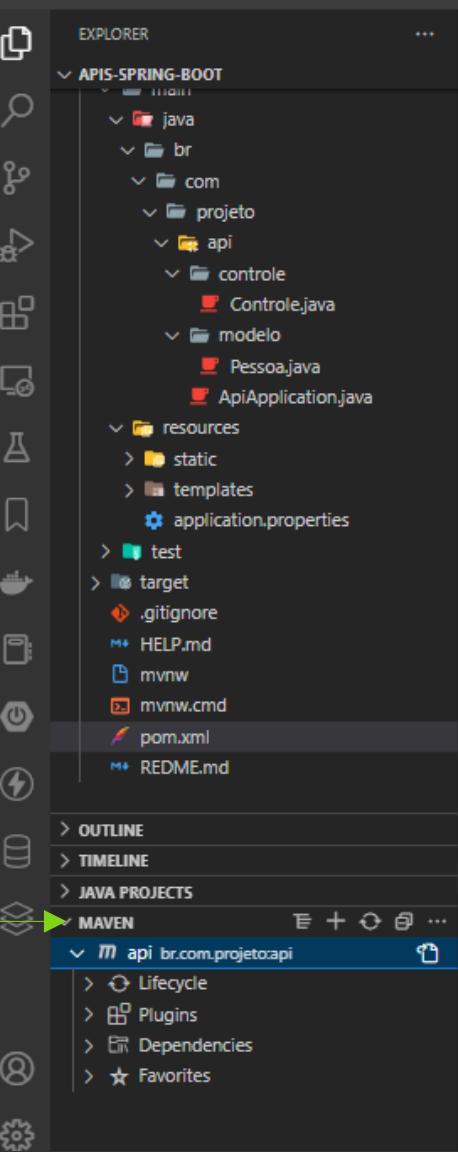
DEBUG CONSOLE

Run: ApiAp...

Run: ApiAp...

## **ADICIONANDO AS DEPENDÊNCIAS**

File Edit Selection View Go Run

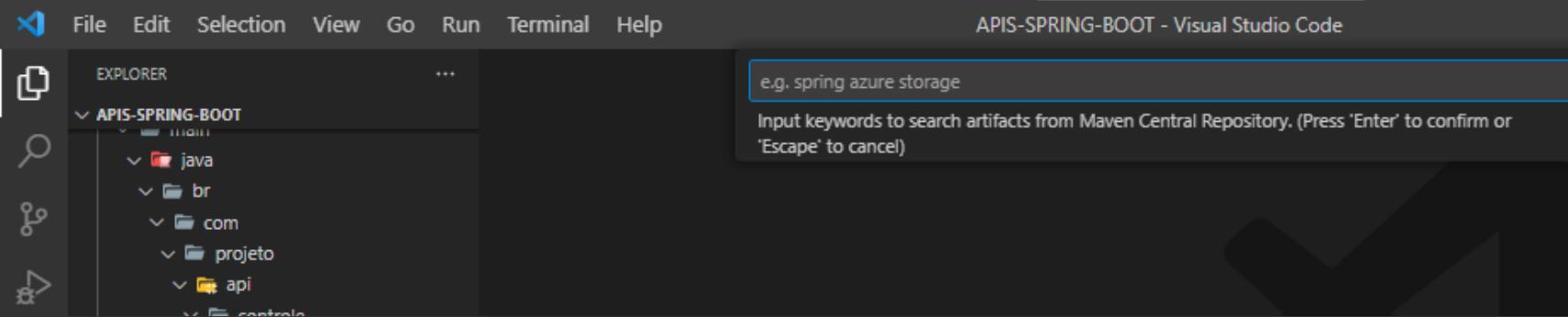


# O QUE É O MAVEN?

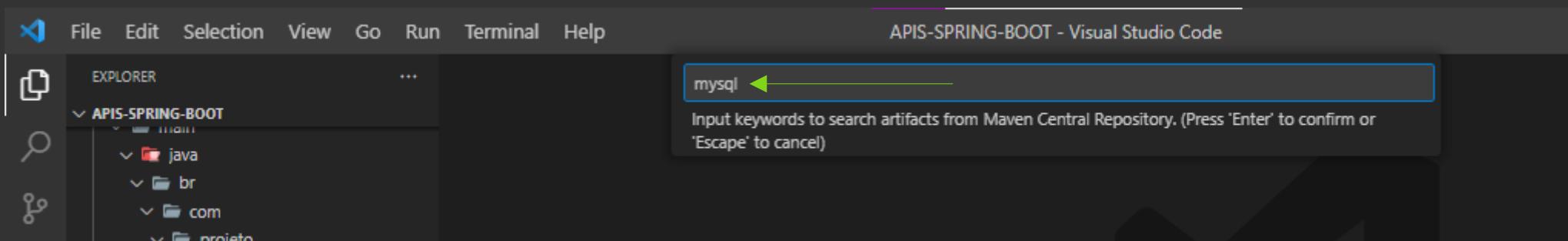
## Responsável por gerenciar os pacotes do spring

Click no sinal de +

**Abre um popup que pede keyords  
O que são keywords? Palavras chaves**

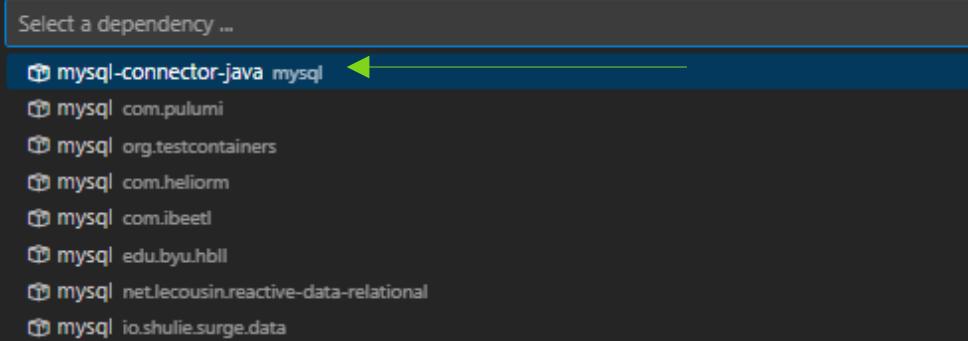
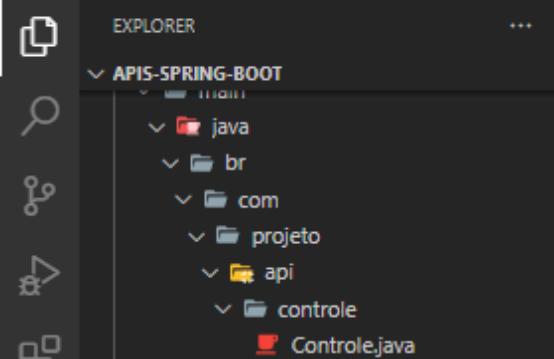


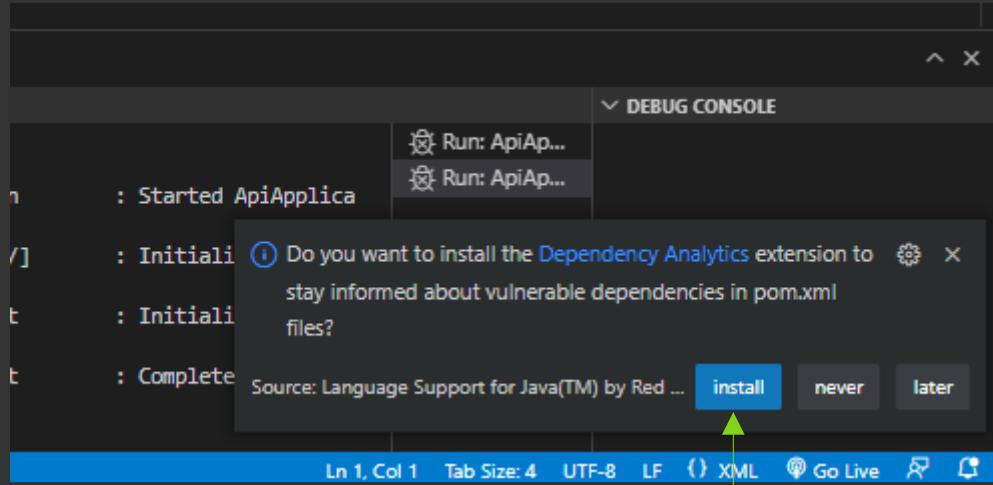
Digite mysql e tecle enter



File Edit Selection View Go Run Terminal Help

APIS-SPRING-BOOT - Visual Studio Code





## EXPLORER pom.xml

APIS-SPRING-BOOT

- java
- br
- com
  - projeto
    - api
      - Controle.java
    - controle
      - Controle.java
    - modelo
      - Pessoa.java
      - ApiApplication.java
  - resources
    - static
    - templates
    - application.properties
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

api > 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>3.0.2</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>br.com.projeto</groupId>
  <artifactId>api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>api</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>

    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.32</version>
    </dependency>
  </dependencies>

```

Dependencia adicionada  
ela aparece no arquivo  
pom.xml

PROBLEMS 1 OUTPUT TERMINAL

## TERMINAL

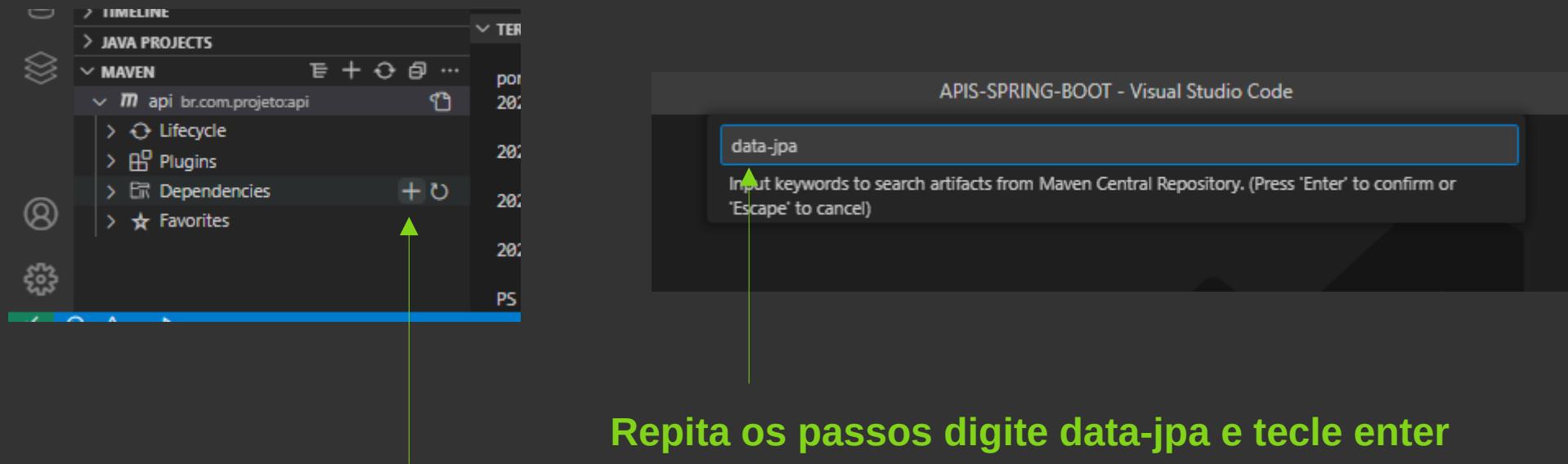
```
port(s): 8080 (http) with context path ''
2023-01-23T11:23:14.611-03:00  INFO 6540 --- [ restartedMain] br.com.projeto.api.ApiApplication
: Started ApiApplication in 11.71 seconds (process running for 14.239)
2023-01-23T11:23:15.963-03:00  INFO 6540 --- [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-23T11:23:16.063-03:00  INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet
: Initializing Servlet 'dispatcherServlet'
2023-01-23T11:23:16.087-03:00  INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet
: Completed initialization in 2 ms
PS E:\APIS-SPRING-BOOT>
```

Run: ApiAp...

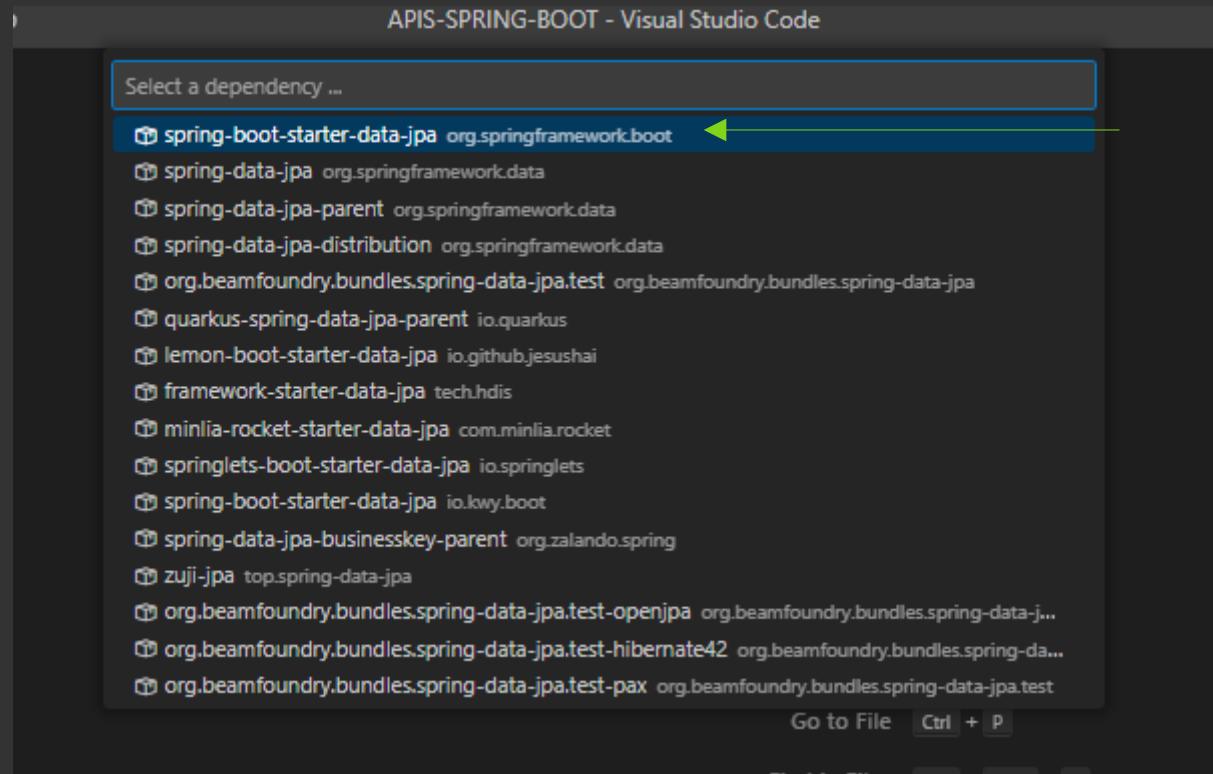
Run: ApiAp...

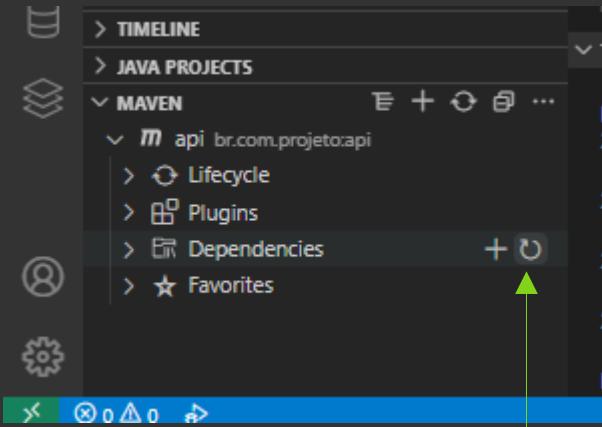
## DEBUG CONSOLE

**Essa dependencia serve para dar acesso ao banco de dados  
Vamos agora instalar a dependencia jpa**



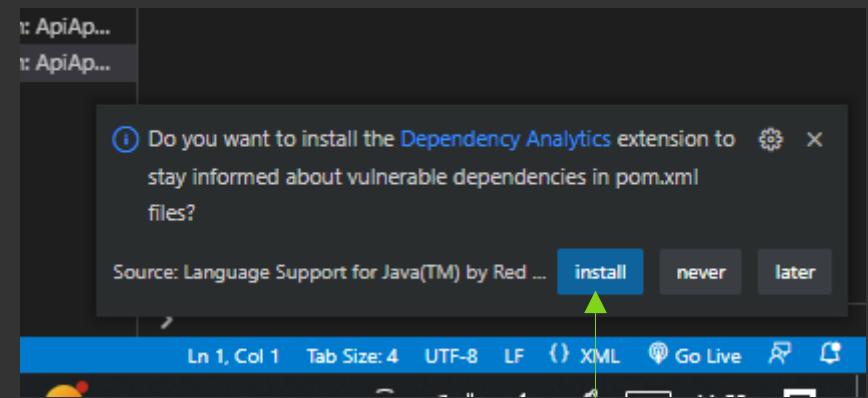
Repita os passos digite data-jpa e tecle enter





Se não estiver baixando tente um refresh

Repita os passos para baixar a dependencia depois do refresh



File Edit Selection View Go Run Terminal Help • pom.xml - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER API-SPRING-BOOT

api > 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>3.0.2</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>br.com.projeto</groupId>
  <artifactId>api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>api</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
      <version>3.0.2</version>
    </dependency>
  </dependencies>

```

Se deu tudo ok a dependencia aparece no pom.xml

OUTLINE

PROBLEMS

TERMINAL

MAVEN

MAVEN

port(s): 8080 (http) with context path ''  
2023-01-23T11:23:14.611-03:00 INFO 6540 --- [ restartedMain] br.com.projeto.api.ApiApplication  
: Started ApiApplication in 11.71 seconds (process running for 14.239)  
2023-01-23T11:23:15.963-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[]/  
: Initializing Spring DispatcherServlet 'dispatcherServlet'  
2023-01-23T11:23:16.063-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet  
: Initializing Servlet 'dispatcherServlet'  
2023-01-23T11:23:16.087-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet  
: Completed initialization in 2 ms

Ln 1, Col 1 Tab Size: 4 UTF-8 LF { XML Go Live

**Existe outro jeito de instalar dependencias?**

**Sim sopie e cole elas dentro do arquivo pom.xml**

**Ex:**

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
    <version>3.0.2</version>
  </dependency>
```

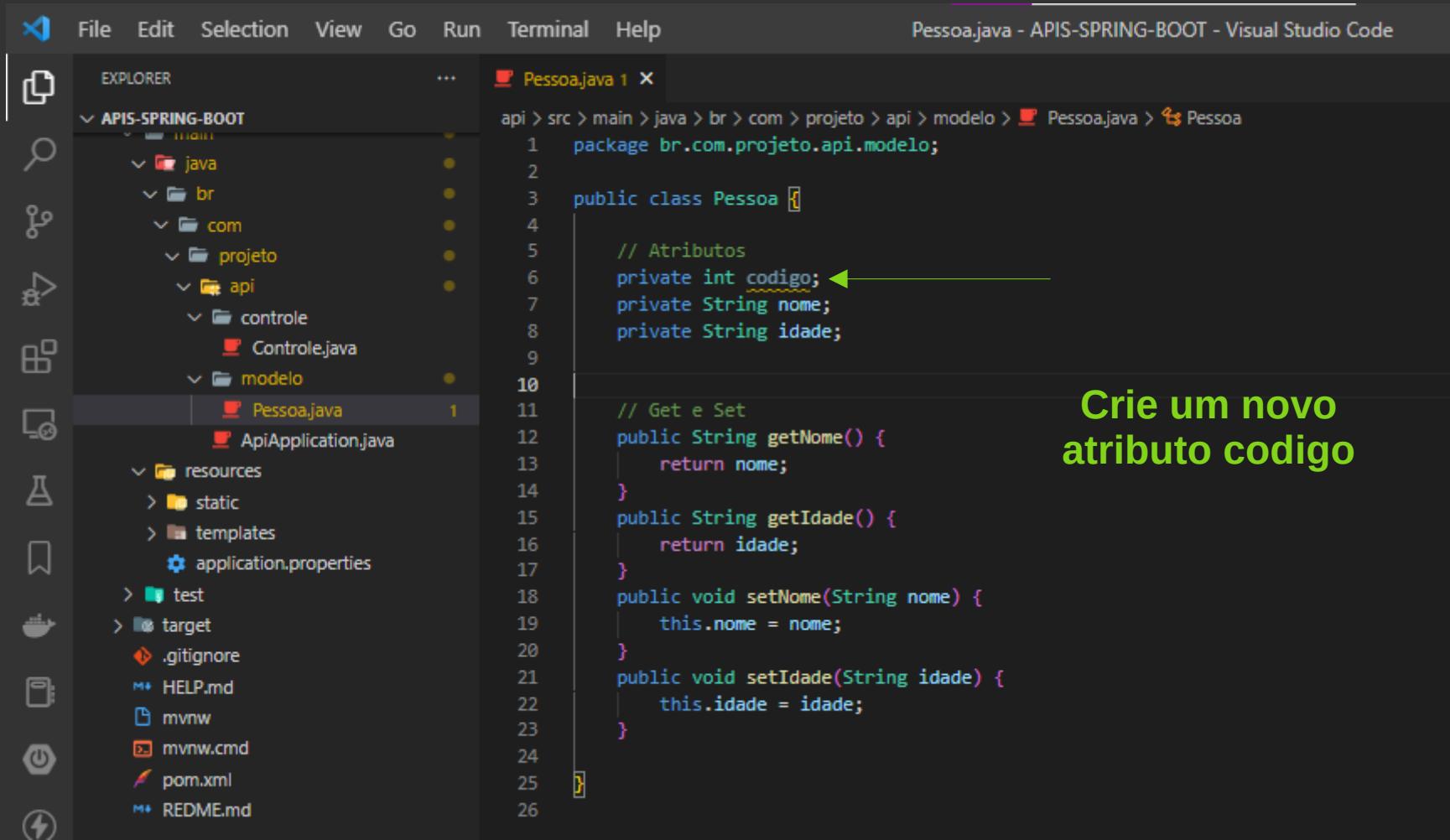
**Adicionando assim temos ela instalada**

**Não esqueça de salvar o arquivo pom.xml depois de instalar a dependencia**

**Essas duas dependencias já nos da o acesso ao banco de dados e ao jpa**

#12

Configurando o modelo pessoa pra gerar tabelas



## Crie um novo atributo código

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoa.java - APIS-SPRING
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including main, java, br, com, projeto, api, controle, modelo, resources, static, templates, application.properties, test, target, and editorconfig.
- Editor View:** The file Pessoa.java is open at line 1. The code is as follows:

```
1 package br.com.projeto.api.modelo;
2
3 public class Pessoa {
4
5     // Atributos
6     private int codigo;
```
- Context Menu (Quick Fix...):** A context menu is open over the line "private int codigo;". It contains the following options:
  - Remove 'codigo', keep assignments with side effects
  - More Actions...
  - Generate Getter and Setter for 'codigo' (selected)
  - Generate Getter for 'codigo'
  - Generate Setter for 'codigo'
  - Generate Constructors...
  - Add final modifier for 'codigo'

Selecione a linha todo e click na lampada pra gerar os getters e setters desse atributo

Run Terminal Help

Pessoa.java - APIS-SPR

...

Pessoa.java X

api > src > main > java > br > com > projeto > api > modelo > Pessoa.java

```
6     private int codigo;
7     private String nome;
8     private String idade;
9
10
11    // Get e Set
12    public String getNome() {
13        return nome;
14    }
15    public String getIdade() {
16        return idade;
17    }
18    public void setNome(String nome) { ←
19        this.nome = nome;
20    }
21    public void setIdade(String idade) {
22        this.idade = idade;
23    }
24    public int getCodigo() {
25        return codigo;
26    }
27    public void setCodigo(int codigo) {
28        this.codigo = codigo;
29    }
30
31 }
```

**Recortei e colei no lugar certo  
Pois ele aparece logo abaixo da próxima  
vou prestar atenção onde o cursor esta  
ficando**

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Code Editor:** The file "Pessoa.java" is open. The code is as follows:

```
1 package br.com.projeto.api.modelo;
2
3 @Entity
4 public class Pessoa {
5     // ...
6 }
7
8
9
10
11
12
13
14
15
16
```

A tooltip is displayed over the "@Entity" annotation, providing information about the Jakarta Persistence Entity annotation:

- jakarta.persistence.Entity**
- Description:** Specifies that the class is an entity. This annotation is applied to the entity class.
- Since:** 1.0

Adicione a anotation e seu import

The screenshot shows the Visual Studio Code interface with the following details:

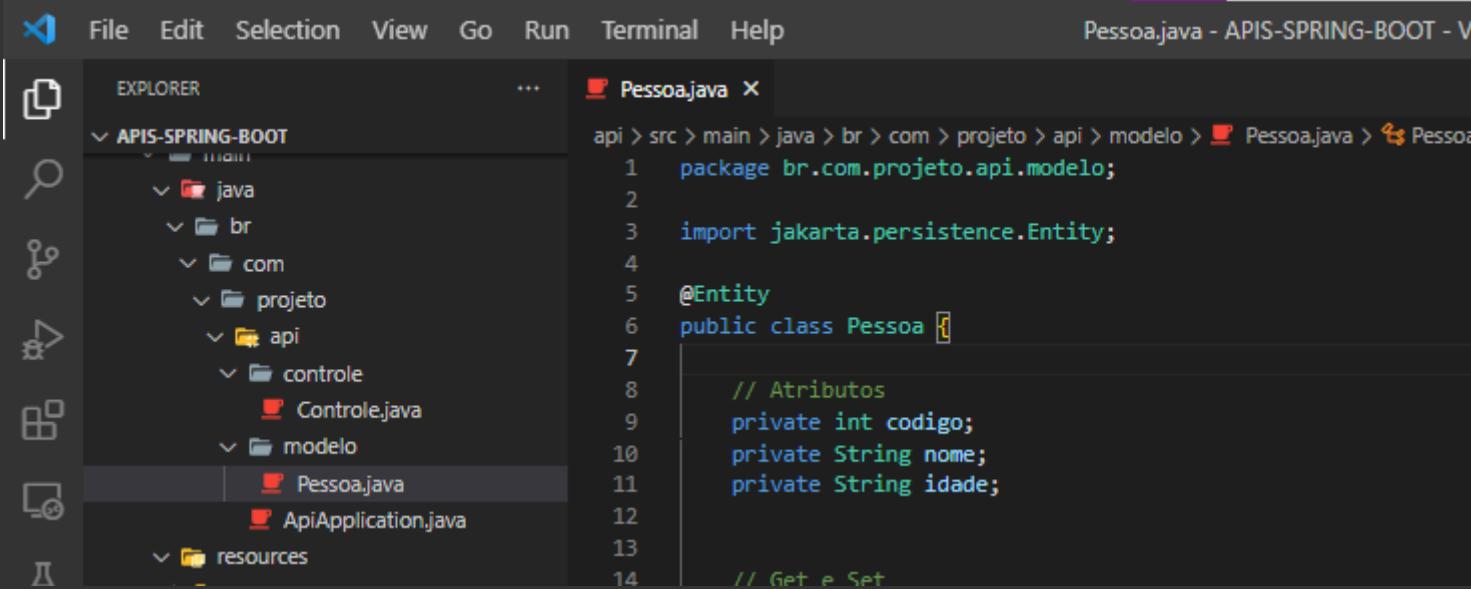
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `Pessoa.java` from the `APIS-SPRING-BOOT` project. The code is as follows:

```
3 import jakarta.persistence.Entity; ←
4
5 @Entity ←
6 public class Pessoa {←
7
8     // Atributos
9     private int codigo;
10    private String nome;
11    private String idade;
12
13    // Get e Set
14    public String getName() {←
15
```

**Explorer:** Shows the project structure:

- `APIS-SPRING-BOOT` (selected)
- `src`
  - `java`
    - `br`
      - `com`
        - `projeto`
          - `api`
            - `controle`
              - `Controle.java`
            - `modelo`
              - `Pessoa.java` (selected)
              - `ApiApplication.java`
    - `resources`

# Pra que serve o @Entity? Ele especifica a criação da tabela



The screenshot shows a Java IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Title:** Pessoa.java - APIS-SPRING-BOOT - Vi
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - src
  - main
  - java
  - br
  - com
  - projeto
  - api
  - controle
  - Controle.java
  - modelo
  - Pessoa.java
  - ApiApplication.java
- Code Editor:** Displays the Pessoa.java code:

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4
5 @Entity
6 public class Pessoa {
7
8     // Atributos
9     private int codigo;
10    private String nome;
11    private String idade;
12
13    // Get e Set
14 }
```

Nesse caso ele criaria uma tabela com os campos **codigo nome idade**  
Por padrão a tabela segue o nome da classe **Pessoa**

**E se eu quiser trocar o nome dessa tabela posso?**

**Sim adicionando o @Table e seu importe e passando dentro dos parentesis o nome da  
tabela que você quer**

**@Table(name = "pessoas")**



EXPLORER

APIS-SPRING-BOOT

```
src
  main
    java
      br
        com
          projeto
            api
              controle
                Controle.java
              modelo
                Pessoa.java
                ApiApplication.java
            resources
              static
              templates
              application.properties
    test
    target
    .gitignore
    HELP.md
    mvnw
```

Pessoa.java 1

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4
5 @Entity
6 @Table
7 public class Pessoa {
8     @Table(name = "pessoas")
9     @Id
10    @GeneratedValue(strategy = GenerationType.IDENTITY)
11    private Long id;
12
13    @Column(name = "nome")
14    private String nome;
15
16    @Column(name = "idade")
17    private Integer idade;
18
19    public String getIdade() {
20        return idade;
21    }
22
23    public void setNome(String nome) {
24        this.nome = nome;
25    }
26}
```

jakarta.persistence.Table

Specifies the primary table for the annotated entity. Additional tables may be specified using SecondaryTable or SecondaryTables annotation.

If no Table annotation is specified for an entity class, the default values apply.

Example:

```
@Entity
@Table(name="CUST", schema="RECORDS")
public class Customer { ... }
```

Since:

1.0

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoajava - APIS-SPRING-BOOT - Visual Studio
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT":
  - src
  - main
  - java
  - br
  - com
  - projeto
  - api
  - controle
  - Controle.java
  - modelo
  - Pessoajava (selected)
  - ApiApplication.java
  - resources
  - static
- Code Editor:** Displays the content of Pessoajava.java:

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Table; ←
5
6 @Entity
7 @Table(name = "pessoas") ←
8 public class Pessoa {
9
10    // Atributos
11    private int codigo;
12    private String nome;
13    private String idade;
14}
```

A yellow bracket highlights the annotations @Entity and @Table.

Com essa annotation damo o nome de pessoas a nossa tabela  
E caso não queria mudar o nome da tabela não precisaria dessa  
linha

**Vamos adicionar a chave primaria**

**Adicione a annotation @Id e seu importe lembre que é do pacote javax.persistence**



EXPLORER

APIS-SPRING-BOOT

java  
br  
com  
projeto  
api  
controle  
Controle.java  
modelo  
Pessoa.java  
ApiApplication.java  
resources  
static  
templates  
application.properties  
test  
target  
.gitignore

Pessoa.java 1

```
api > src > main > java > br > com > projeto > api > modelo > Pessoa.java > Pessoa > codigo
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Table;
5
6 @Entity
7 @Table(name = "pessoas")
8 public class Pessoa {
9
10     // Atributos
11     @Id
12     private jakarta.persistence.Id id - jakarta.persistence.Id
13     private org.springframework.data.annotation.Id id - org.springframework.data.annotation.Id
14     private jakarta.persistence.IdClass idClass - jakarta.persistence.IdClass
15         private org.hibernate.annotations.IdGeneratorType idGeneratorType - org.hibernate.annotations.IdGeneratorType
16         private org.hibernate.annotations.RowId rowId - org.hibernate.annotations.RowId
17         private org.hibernate.annotations.Index index - org.hibernate.annotations.Index
18     public jakarta.persistence.IndexColumn indexColumn - org.hibernate.annotations.IndexColumn
19         private jakarta.persistence.Index index - jakarta.persistence.Index
20     } private org.springframework.stereotype.Indexed indexed - org.springframework.stereotype.Indexed
21     public com.fasterxml.jackson.annotation.JsonIdentityInfo jsonIdentityInfo - com.fasterxml.jackson.annotation.JsonIdentityInfo
```

jakarta.persistence.Id

Specifies the primary key of an entity. The field or property to which the `Id` annotation is applied should be one of the following types: any Java primitive type; any primitive wrapper type; `String`; `java.util.Date`; `java.sql.Date`; `java.math.BigDecimal`; `java.math.BigInteger`.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoajava - API-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT":
  - main
  - java
    - br
      - com
        - projeto
          - api
            - controle
              - Controle.java
            - modelo
              - Pessoajava
              - ApiApplication.java
    - resources
      - static
      - templates
      - application.properties
  - Code Editor:** Displays the content of the Pessoajava.java file.

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.Id;
5 import jakarta.persistence.Table;
6
7 @Entity
8 @Table(name = "pessoas")
9 public class Pessoa {
10
11     // Atributos
12     @Id
13     private int codigo;
14     private String nome;
15     private String idade;
16
17 }
```

Podemos pedir um auto-increment  
Adicionando a annotation `@GeneratedValue` e seu importe e  
dentro passamos o strategy = `GenerationType.AUTO`

```
11 // Atributos
12 @Id
13 @Generated
14 private int o Generated - jakarta.annotation
15 private String o Generated - javax.annotation.processing
16 private String o Generated - org.hibernate.annotations
17     o GeneratedColumn - org.hibernate.annotations
18     o GeneratedColumn - org.hibernate.annotations.Diale...
19 // Get e o GeneratedColumns - org.hibernate.annotations.Dial...
20 public String o GeneratedValue - jakarta.persistence
21     return o GeneratorType - org.hibernate.annotations
22 }
23 public String getIdade() {
24     return idade;
25 }
```

The `Generated` annotation is used to mark source code that has been generated. It can also be used to differentiate user written code from generated code in a single file.

The `value` element must have the name of the code generator. The recommended convention is to use the fully qualified name of the code generator in the value field, for example `com.company.package.classname`.

Pessoa.java ●

api > src > main > java > br > com > projeto > api > modelo > Pessoa.java > Pesso... > Pesso... > cod...

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue; ←
5 import jakarta.persistence.Id;
6 import jakarta.persistence.Table;
7
8 @Entity
9 @Table(name = "pessoas")
10 public class Pessoa {
11
12     // Atributos
13     @Id
14     @GeneratedValue
15     private int codigo;
16     private String nome;
17     private String idade;
18 }
```

api > src > main > java > br > com > projeto > api > modelo > Pessoajava > Pessoa > codigo

```
2  
3 import jakarta.persistence.Entity;  
4 import jakarta.persistence.GeneratedValue;  
5 import jakarta.persistence.GenerationType; ←  
6 import jakarta.persistence.Id;  
7 import jakarta.persistence.Table;  
8  
9 @Entity  
10 @Table(name = "pessoas")  
11 public class Pessoa {  
12  
13     // Atributos  
14     @Id  
15     @GeneratedValue(strategy = GenerationType.)  
16     private int codigo;  
17     private String nome;  
18     private String idade;  
19  
20     // Get e Set  
21     public String getNome() {  
22         return nome;  
23     }  
24     public String getIdade() {  
25         return idade;  
26     }  
27     public void setNome(String nome) {
```

➤ AUTO : GenerationType  
➤ IDENTITY : GenerationType  
➤ SEQUENCE : GenerationType  
➤ TABLE : GenerationType  
➤ UUID : GenerationType  
↳ class : Class<jakarta.persistence.GenerationType>  
□ cast  
□ var

GenerationType.AUTO : GenerationType x

Indicates that the persistence provider should pick an appropriate strategy for the particular database. The AUTO generation strategy may expect a database resource to exist, or it may attempt to create one. A vendor may provide documentation on how to create such resources in the event that it does not support schema generation or cannot create the schema resource at runtime.

Pessoa.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
              - modelo
                - Pessoa.java
                - ApiApplication.java

Pessoa.java

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8
9 @Entity
10 @Table(name = "pessoas")
11 public class Pessoa {
12
13     // Atributos
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY) ←
16     private int codigo;
17     private String nome;
18     private String idade;
19
20
21     // Get e Set
22     public String getNome() {
23         return nome;
24     }
25     public String getIdade() {
26         return idade;
27     }
28 }
```

Adicione as anotações e seus imports que serve para auto increment

# #13

## Conexão com mysql

**Para configurar a conexão temos que usar o arquivo application.properties**

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active Editor:** application.properties - API
- Explorer:** Shows the project structure:
  - API-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
      - resources
        - static
        - templates
  - application.properties
  - test
  - target
    - .gitignore
    - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Active File:** application.properties - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
      - resources
        - static
        - templates
    - application.properties
    - test
    - target
      - .gitignore
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
      - REDME.md
- Code Editor:** Displays the content of the application.properties file:

```
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.
2 spring.jpa.hibernate.ddl-auto=update
3
4 # Acesso ao banco de dados
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:3306/nome_base_de_dados
6
7 # Usuário do banco de dados
8 spring.datasource.username=usuário
9
10 # Senha do banco de dados
11 spring.datasource.password=senha
12
```
- Bottom Center Text:** **Digite essa estrutura dentro do arquivo**

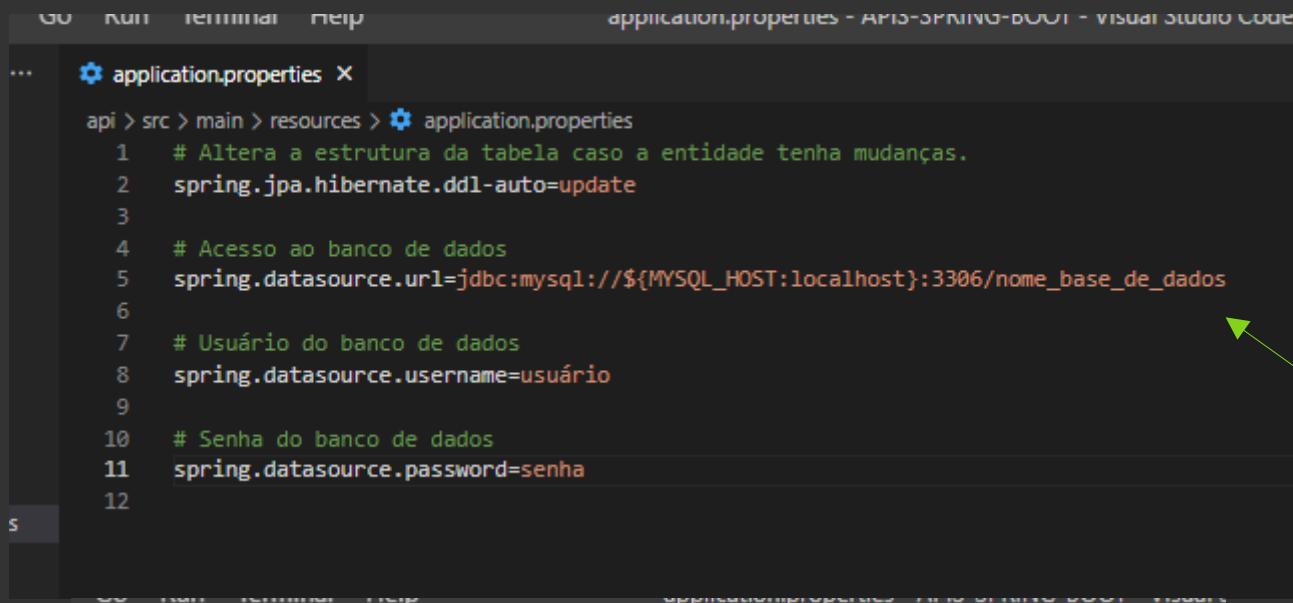
**Se o arquivo pom.xml marcar um erro feche o vscode e abra novamente**

The screenshot shows the Visual Studio Code interface with the application.properties file open. The file contains configuration for a Spring Boot application, specifically for a MySQL database connection. The code is as follows:

```
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.
2 spring.jpa.hibernate.ddl-auto=update
3
4 # Acesso ao banco de dados
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:3306/nome_base_de_dados
6
7 # Usuário do banco de dados
8 spring.datasource.username=usuário
9
10 # Senha do banco de dados
11 spring.datasource.password=senha
12
```

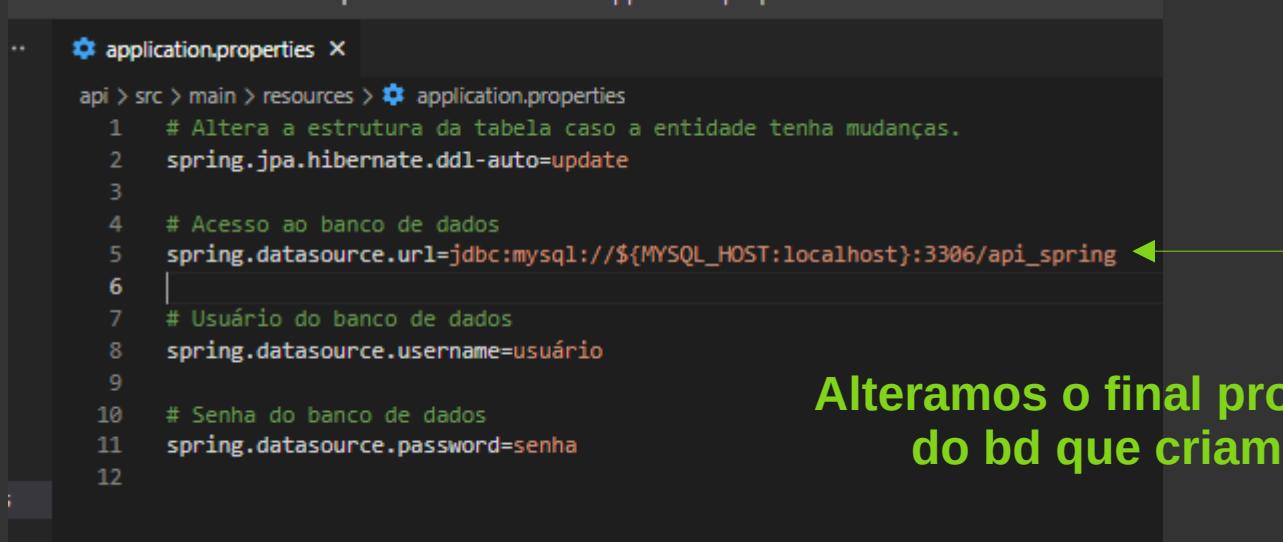
Annotations are present in the bottom right corner of the code editor:

- A green arrow points from the word "senha" to the line "spring.datasource.password=senha".
- A green arrow points from the word "usuário" to the line "spring.datasource.username=usuário".
- A green arrow points from the word "String de conexão" to the line "spring.datasource.url=jdbc:mysql://\${MYSQL\_HOST}:3306/nome\_base\_de\_dados".



application.properties - API-SPRING-BOOT - visual studio code

```
... application.properties X  
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:localhost}:3306/nome_base_de_dados  
6  
7 # Usuário do banco de dados  
8 spring.datasource.username=usuário  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=senha  
12
```



application.properties X

```
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:localhost}:3306/api_spring ←  
6 |  
7 # Usuário do banco de dados  
8 spring.datasource.username=usuário  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=senha  
12
```

Alteramos o final pro nome  
do bd que criamos

```
.. application.properties X  
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:localhost}:3306/api_spring  
6  
7 # Usuário do banco de dados  
8 spring.datasource.username=usuário ←  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=senha  
12
```

v Go Run Terminal Help application.properties - APIS-SPRING-BOOT - Visual Studio Code

```
.. application.properties X  
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:localhost}:3306/api_spring  
6  
7 # Usuário do banco de dados  
8 spring.datasource.username=root ←  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=senha  
12
```

es

Lembrando que esses dados  
são do mysql quando  
inicialmente configuramos  
Nele o usuário é root

Connect X

Connect to server

Name Connection Name Connection Scope Global Premium Only

Group Group Connect Timeout 5000 Read Only Save Password Forever

Note: This extension needs to be [pay](#) to unlock all features.

Server Type

MySQL MariaDB PostgreSQL SQL Server Oracle SQLite DM8 ClickHouse Redshift

SSH Docker Redis ElasticSearch MongoDB S3/Minio FTP Neo4j JDBC(Beta)

\* Host 127.0.0.1 \* Port 3306

\* Username root \* Password

Database Connection database Advance Option

SSH Tunnel Use SSL Socket Path e.g. /var/run/mysql/mysqld.sock Use

Save Connect Close

Desses dados que  
estou falando

Senha:  
Akilles101

```
application.properties X  
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql:// ${MYSQL_HOST}:localhost}:3306/api_spring  
6  
7 # Usuário do banco de dados  
8 spring.datasource.username=root  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=senha ←  
12
```

Go Run Terminal Help • application.properties - APIS-SPRING-BOOT - Visual Studio Code

```
application.properties ●  
api > src > main > resources > application.properties  
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.  
2 spring.jpa.hibernate.ddl-auto=update  
3  
4 # Acesso ao banco de dados  
5 spring.datasource.url=jdbc:mysql:// ${MYSQL_HOST}:localhost}:3306/api_spring  
6  
7 # Usuário do banco de dados  
8 spring.datasource.username=root  
9  
10 # Senha do banco de dados  
11 spring.datasource.password=Akilles101 ←  
12
```

Akilles101

The screenshot shows the MySQL Workbench application. On the left, there's a sidebar with icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and a Database browser. The Database browser shows a connection to '127.0.0.1@3306' with databases 'dbagenda', 'sakila', and 'world'. The main area has tabs for 'Connect' and 'create-db-template.sql'. The 'create-db-template.sql' tab is active, displaying the following SQL code:

```
C: > Users > brito > AppData > Roaming > Code > User > globalStorage > cweijan.vscode-mysql-client  
      ⚡ Active Connection  
      -- Active: 1674406792944@127.0.0.1@3306 MySQL  
      ▷ Execute  
2 CREATE DATABASE api_spring ←  
3   DEFAULT CHARACTER SET = 'utf8mb4';
```

Pra não deixar duvidas sobre o nome do banco de dados `api_spring`

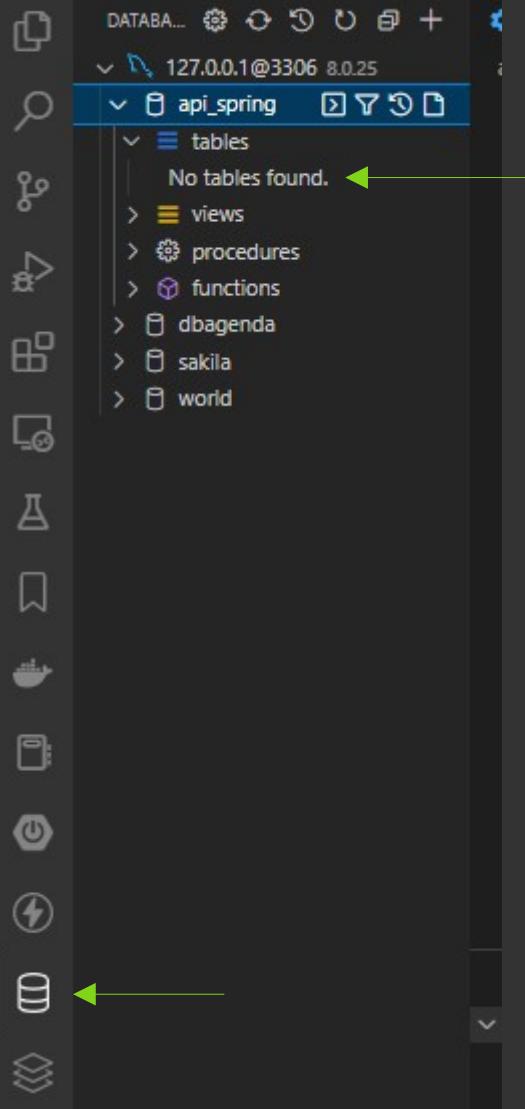
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** application.properties - API-SPRING-BOOT - Visual
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
      - resources
        - static
        - templates
    - application.properties
    - test
    - target
      - .gitignore
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
      - README.md
- Editor View:** Displays the content of the application.properties file.

```
api > src > main > resources > application.properties
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.
2 spring.jpa.hibernate.ddl-auto=update
3
4 # Acesso ao banco de dados
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:3306/api_spring
6
7 # Usuário do banco de dados
8 spring.datasource.username=root
9
10 # Senha do banco de dados
11 spring.datasource.password=Akilles101
12 |
```

Salve os dados

File Edit Selection View Go

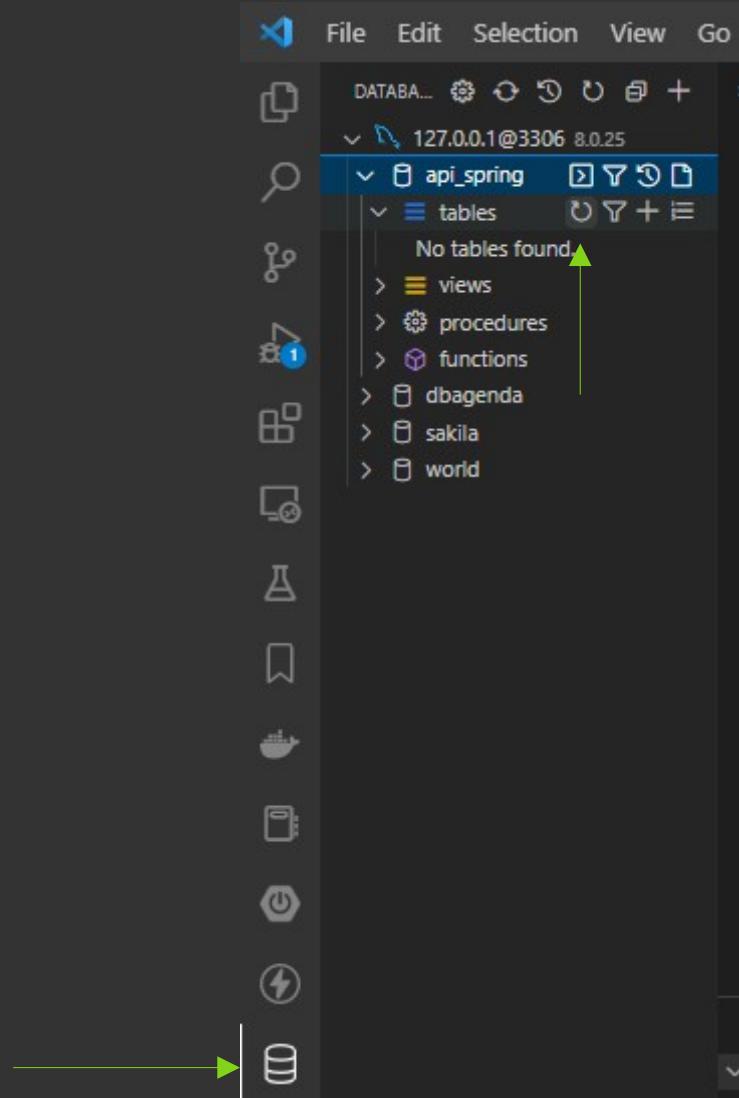


Abra o bad pra ver que ainda não existem tabelas

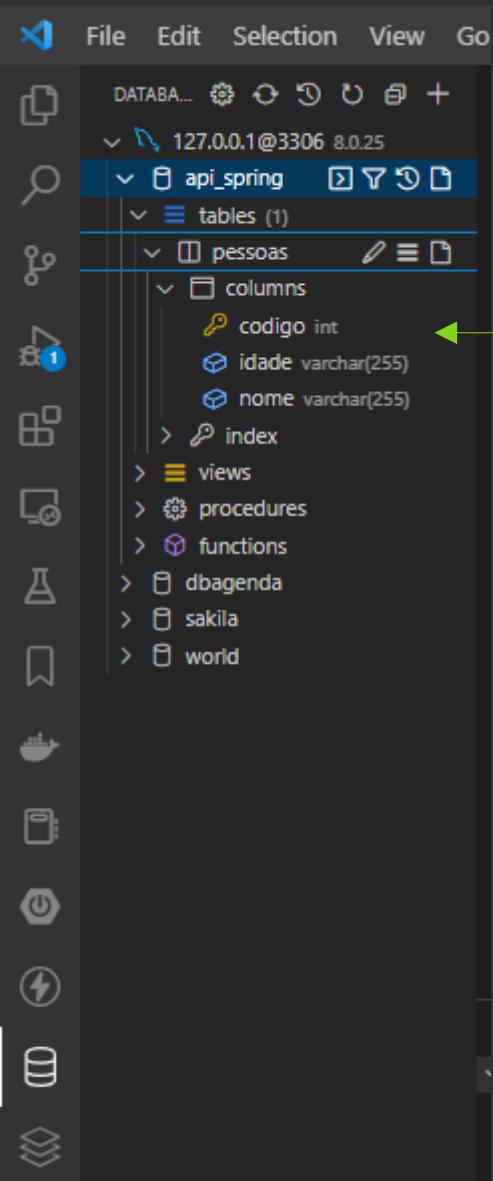
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Title:** application.properties - APIS-SPRING-BOOT - Visual Studio Code
- Left Sidebar:** Shows the project structure under "APPS": "api".
- Right Editor Area:** Displays the content of the application.properties file:

```
1 # Altera a estrutura da tabela caso a entidade tenha mudanças.
2 spring.jpa.hibernate.ddl-auto=update
3
4 # Acesso ao banco de dados
5 spring.datasource.url=jdbc:mysql://${MYSQL_HOST}:3306/api_spring
6
7 # Usuário do banco de dados
8 spring.datasource.username=root
9
10 # Senha do banco de dados
11 spring.datasource.password=Akilles101
12
```
- Bottom Center:** A large green button with the text "Execute o projeto" (Execute the project).
- Bottom Left:** A vertical toolbar with icons for file operations (New, Open, Save, Find, Copy, Paste, Delete, Undo, Redo), a search icon, a refresh icon, and a power icon.



Depois de executar  
atualize refresh



Os dados foram criados

#14  
**Criando repositorio**

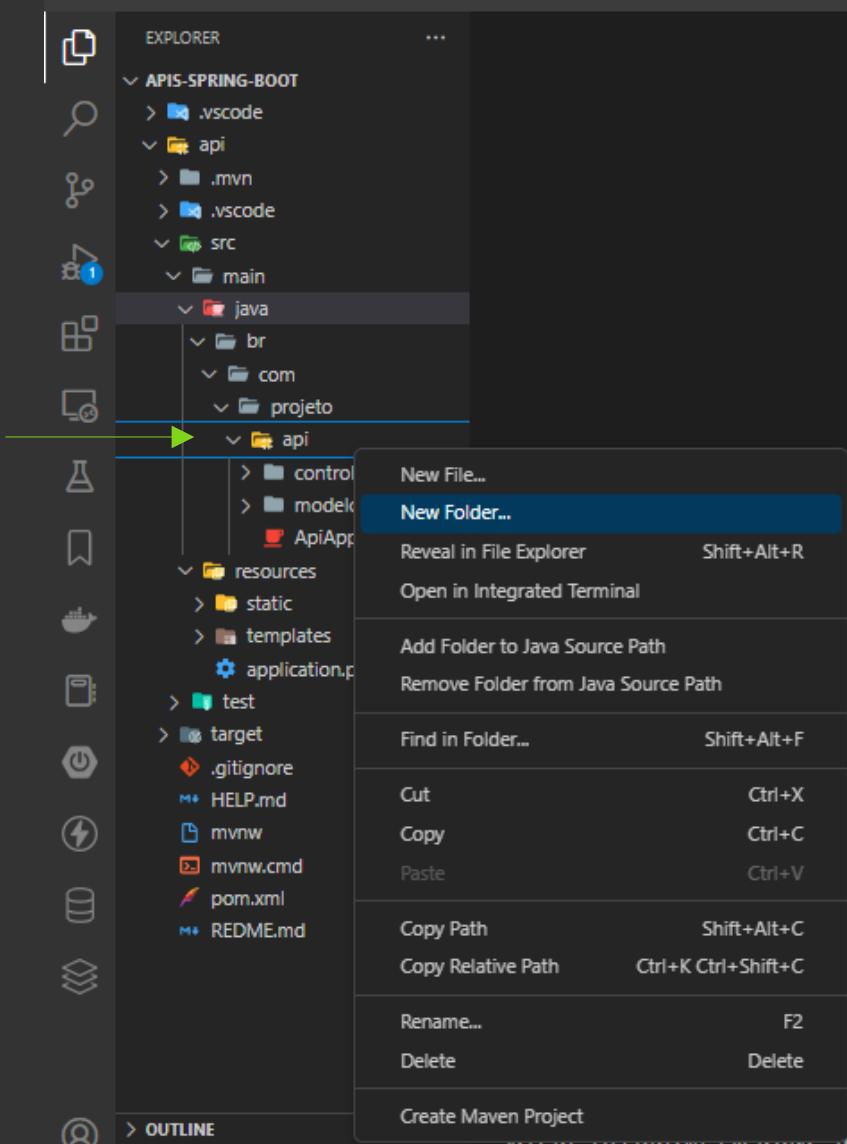
**o que é um repositorio?**

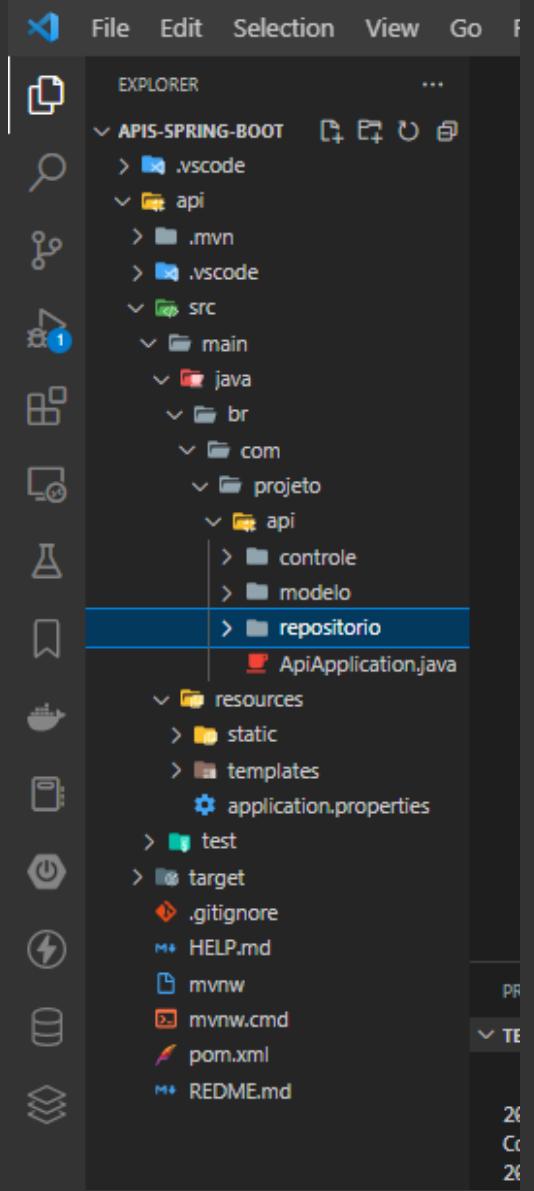
**Uma camada de persistencia**

**O que é uma camada de persistencia?**

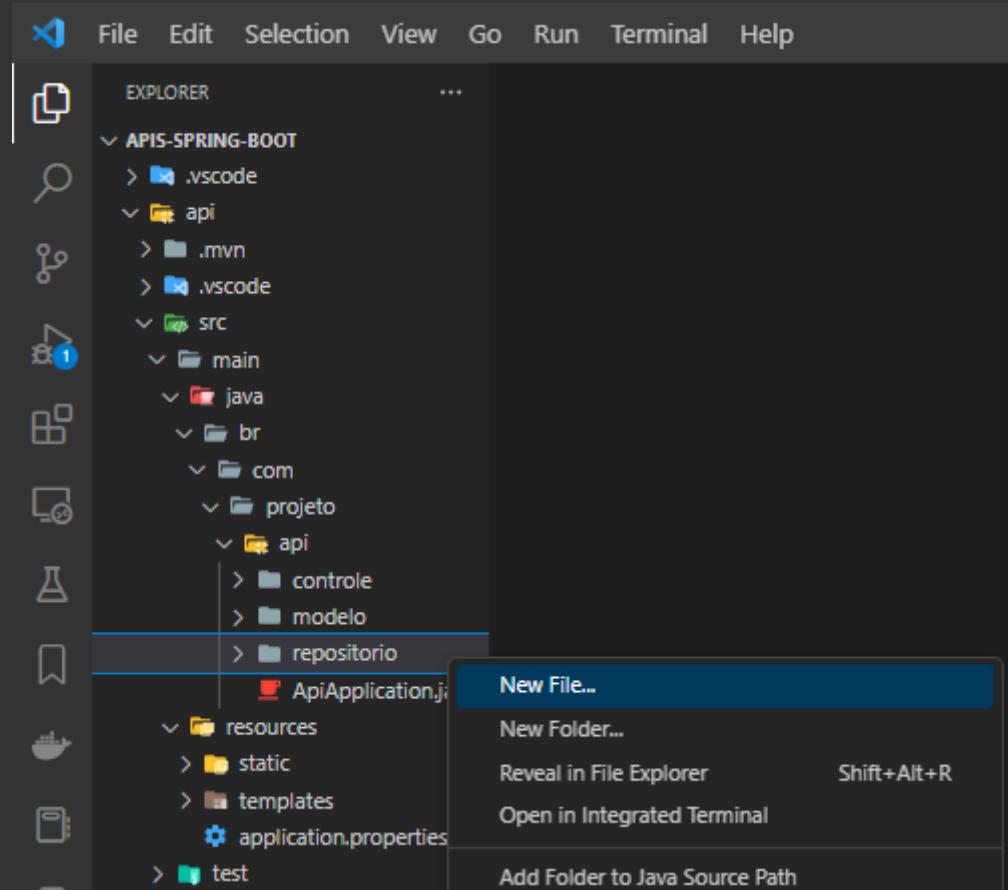
**Parte do projeto onde fazemos as ações do banco de dados**

**Os chamados cruds**





Crie a pasta  
repositorio



File Edit Selection View Go Run Terminal Help

• Repositorio.java - APIS-SPRING-BOOT - Visual Studio C

REPOSITORY

APIS-SPRING-BOOT

.vscode .mvn .vscode

api

src

main

java

br

com

projeto

api

controle

modelo

repositorio

Repositorio.java

ApiApplication.java

resources

static

templates

application.properties

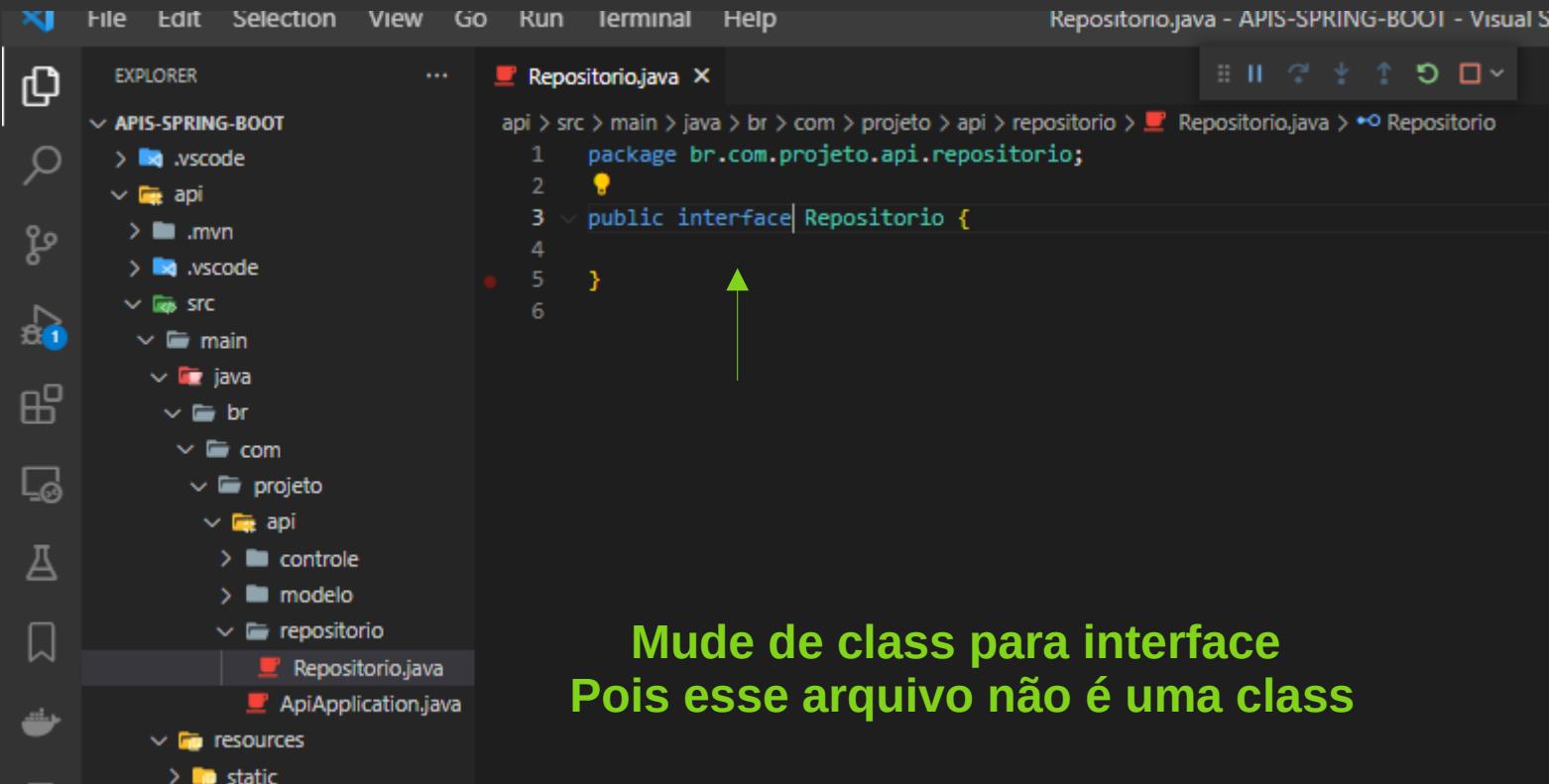
test

Repositorio.java

```
1 package br.com.projeto.api.repositorio;
2
3 public class Repositorio {
4
5 }
```

class Repositorio  
interface Repositorio  
enum Repositorio  
record Repositorio()  
abstract class Repositorio  
@interface Repositorio

Dentro de repositorio crie  
Repositorio.java



```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Repositorio
1 package br.com.projeto.api.repositorio;
2
3 public interface Repositorio {
4
5 }
6
```

Mude de class para interface  
Pois esse arquivo não é uma class

Para o spring saber que é um repositorio adicionamos a  
anotation @Repository e seu import

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT": .vscode, api, src (with main, java, br, com, projeto, repository), and .mvn, .vscode.
- Code Editor:** Displays the file "Repository.java".

```
1 package br.com.projeto.api.repository;
2
3 @Reposito
4 public <--> Repository - org.springframework.stereotype.Repository
5
6 }
```
- Tooltip:** A tooltip is displayed over the "@Reposito" part of the code, listing the stereotype's methods:
  - o Repository - org.springframework.stereotype.Repository
  - o RepositoryDefinition - org.springframework.data.r...
  - o NoRepositoryBean - org.springframework.data.repos...
  - o EnableJpaRepositories - org.springframework.data...
  - o ConditionalOnRepositoryType - org.springframework...
- Description:** Below the tooltip, there are two descriptive paragraphs:
  - Indicates that an annotated class is a "Repository", originally defined by Domain-Driven Design (Evans, 2003) as "a mechanism for encapsulating storage, retrieval, and search behavior which emulates a collection of objects".
  - Teams implementing traditional Jakarta EE patterns such as "Data Access Object" may also apply this stereotype to DAO classes, though care should be taken to understand the distinction between Data Access Object and DDD-style repositories before...

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Bar:** File Edit Selection View Go Run Terminal Help
- Title Bar:** Repositorio.java - API-SPRING-BOOT
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT":
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
    - control
- Code Editor:** The file "Repositorio.java" is open, showing the following code:

```
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public interface Repositorio {
7
8 }
9
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including .vscode, api, src (main/java(br/com/projeto/api/repositorio), resources, static, templates, application.properties, test, and target).
- Code Editor:** Displays the following Java code:

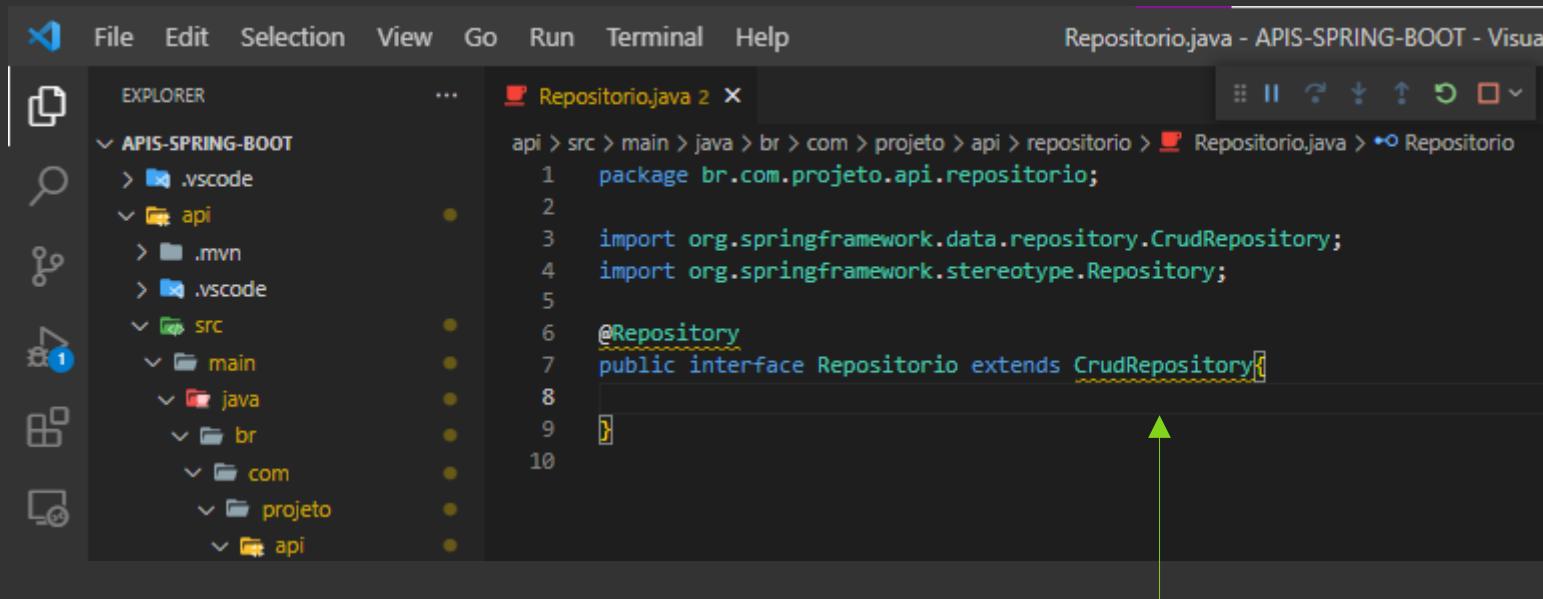
```
api > src > main > java > br > com > projeto > api > repositorio > Repository.java > Repository
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public interface Repository extends Crud[
```
- IntelliJ IDEA Completion Screenshot:** A modal window shows completion suggestions for the word "Crud". The suggestions include:
  - o CrudMethodMetadata - org.springframework.data.jpa...
  - o CrudMethods - org.springframework.data.repository...
  - o CrudRepository - org.springframework.data.repository...
  - o CriteriaUpdate - jakarta.persistence.criteria...
  - o CurrentTenantIdentifierResolver - org.hibernate.c...
  - o CoroutineCrudRepository - org.springframework.dat...
  - o CriteriaBuilder - jakarta.persistence.criteria...
  - o CreateViewOrBuilder - com.mysql.cj.x.protobuf.Mys...
  - o CachedResultSetMetaData - com.mysql.cj.jdbc.result...
  - o ContributableDatabaseObject - org.hibernate.boot...
  - o ConstructorBinding - org.springframework.boot.con...
  - o ConstructorBinding - org.springframework.boot.con...
- Right Panel:** Shows the JavaDoc for the Crud interface, listing its methods and authors.

Interface to abstract CrudMethodMetadata that provide the LockModeType to be used for query execution.

• Author:

  - Oliver Gierke
  - Thomas Darimont
  - Christoph Strobl
  - Mark Paluch
  - Jens Schauder
  - Greg Turnquist

Usando extends criamos uma herança do metodo crudrepository que disponibiliza vários metodos para usarmos



The screenshot shows a dark-themed interface of Visual Studio Code. In the top bar, the file 'Repositorio.java' is open in the editor. The code itself is as follows:

```
1 package br.com.projeto.api.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 @Repository
7 public interface Repositorio extends CrudRepository<T> {
8
9 }
10
```

A yellow arrow points upwards from the bottom of the code area towards the interface name 'Repositorio'. This visual cue is likely used to highlight the specific error or context being discussed in the accompanying text.

Esse erro é gerado pois temos que passar para o crud o modelo que queremos trabalhar e o tipo de dado que é a nossa chave primaria vamos fazer

# Primeiro adicionamos o modelo Pessoa e importamos

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under API-SPRING-BOOT:
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
              - controle
              - modelo
              - repository
                - Repository.java
                - ApiApplication.java
  - Code Editor:** Repository.java (Line 7: public interface Repository extends CrudRepository<Pessoa>{) has a red underline under "Pessoa". A floating search results panel shows suggestions for "Pessoa":
    - Pessoa - br.com.projeto.api.modelo
    - PessimisticLockingFailureException - org.springframework
    - PessimisticReadSelectLockingStrategy - org.hibernate
    - PessimisticReadUpdateLockingStrategy - org.hibernate
    - PessimisticWriteUpdateLockingStrategy - org.hibernate
    - PersistentSortedMap - org.hibernate.collection.spi
    - PessimisticWriteSelectLockingStrategy - org.hibernate
    - PermissionDeniedDataAccessException - org.springframework
    - PagedResourcesAssembler - org.springframework.dat...
    - PagedResourcesAssemblerArgumentResolver - org.spr...
    - PrinterMessageFromOperator - javax.print.attribut...
    - PessimisticForceIncrementLockingStrategy - org.hib...
  - Status Bar:** Shows the file path: br.com.projeto.api.modelo.Pessoa.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `Repositorio.java`, located at `api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java`. The code is as follows:

```
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 import br.com.projeto.api.modelo.Pessoa;
7
8 @Repository
9 public interface Repositorio extends CrudRepository<Pessoa, Long> {
10
11 }
```

O próximo dado é o tipo que sempre vai ter que ser uma classe que definimos no nosso modelo como int e o crudrepository não aceita tipos primitivos  
Por isso utilizamos a classe interger vamos fazer

File Edit Selection View Go Run Terminal Help

REPOSITORY.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
        - api
          - controle
          - modelo
        - repositorio
- Repositorio.java
- ApiApplication.java

resources

- static
- templates
- application.properties

Repositorio.java

```
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 import br.com.projeto.api.modelo.Pessoa;
7
8 @Repository
9 public interface Repositorio extends CrudRepository<Pessoa, Integer>{
10
11 }
```

The Integer class wraps a value of the primitive type int in an object. An object of type Integer contains a single field whose type is int.

In addition, this class provides several methods for converting an int to a String and a String to an int, as well as other constants and methods useful when dealing with an int.

This is a value-based class; programmers should treat instances that are equal as interchangeable and should not use instances for synchronization.

Integer - java.lang  
IntegerDataType - com.mysql.cj.protocol.a.NativeC...  
IntegerDescriptor - org.hibernate.internal.util.t...  
IntegerDeserializer - com.fasterxml.jackson.databa...  
IntegerJavaType - org.hibernate.type.descriptor.j...  
IntegerJdbcType - org.hibernate.type.descriptor.j...  
IntegerPrimitiveArrayType - org.hibernate.type.typ...  
IntegerProperty - com.mysql.cj.conf  
IntegerPropertyDefinition - com.mysql.cj.conf  
IntegerSerializer - com.fasterxml.jackson.databind...  
IntegerSyntax - javax.print.attribute  
IntegerTokenConverter - ch.qos.logback.core.rol...  
...

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer (Left Panel):** Shows the project structure of "APIS-SPRING-BOOT".
  - Root: .vscode, .mvn, .vscode
  - api: .vscode
  - src: main (java, br, com, projeto, api, controle, modelo, repositorio)
  - resources: static, templates, application.properties
  - test
  - target, .gitignore
  - HELP.md, mvnw, mvnw.cmd, pom.xml, README.md
- Editor (Center Panel):** The file "Repositorio.java" is open.

```
1 package br.com.projeto.api.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 import br.com.projeto.api.modelo.Pessoa;
7
8 @Repository
9 public interface Repositorio extends CrudRepository<Pessoa, Integer> {
10
11 }
12
```
- Terminal (Bottom Panel):** Shows the output of a recent command.

```
2023-01-23T13:32:20.116-03:00 INFO 9796 --- [ restartedMain] o.s.b.d.a.OptionalLiveRelo
LiveReload server is running on port 35729
2023-01-23T13:32:20.147-03:00 INFO 9796 --- [ restartedMain] o.s.b.a.web.EndpointLinks

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                - modelo
              - repository
    - resources
      - static
      - templates
      - application.properties
    - test
    - target
      - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md
- Editor:** Displays the file "Repositorio.java 1" with the following code:

```
1 package br.com.projeto.api.repository;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 import br.com.projeto.api.modelo.Pessoa;
7
8 @Repository
9 public interface Repositorio extends CrudRepository<Pessoa, Integer>{
10
11 }
```
- Terminal:** Shows the output of a command run in the terminal tab.

Pela cor e sublinhado temos erros no  
 @repository vamos resolver  
 Abrimos o arquivo pom.xml

File Edit Selection View Go Run Terminal Help pom.xml - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- .mvn
- .vscode
- api
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                - modelo
                - repositorio
                  - Repositorio... 1
                  - ApiApplication.java
  - resources
    - static
    - templates
    - application.properties
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

PROBLEMS 1 OUTPUT TERMINAL

TERMINAL

```
2023-01-23T13:36:43.358-03:00 INFO 9796 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-01-23T13:36:43.365-03:00 INFO 9796 --- [ restartedMain] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint(s) beneath base path '/actuator'
2023-01-23T13:36:43.403-03:00 INFO 9796 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-01-23T13:36:43.411-03:00 INFO 9796 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApplication in 1.815 seconds (process running for 2111.557)
2023-01-23T13:36:43.414-03:00 INFO 9796 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
```

DEBUG CONSOLE

  - Run: ApiApplication
  - Run: ApiApplication

Do you want to install the Dependency Analytics extension to stay informed about vulnerable dependencies in pom.xml files?

Source: Language Support for Java(TM) by Red Hat

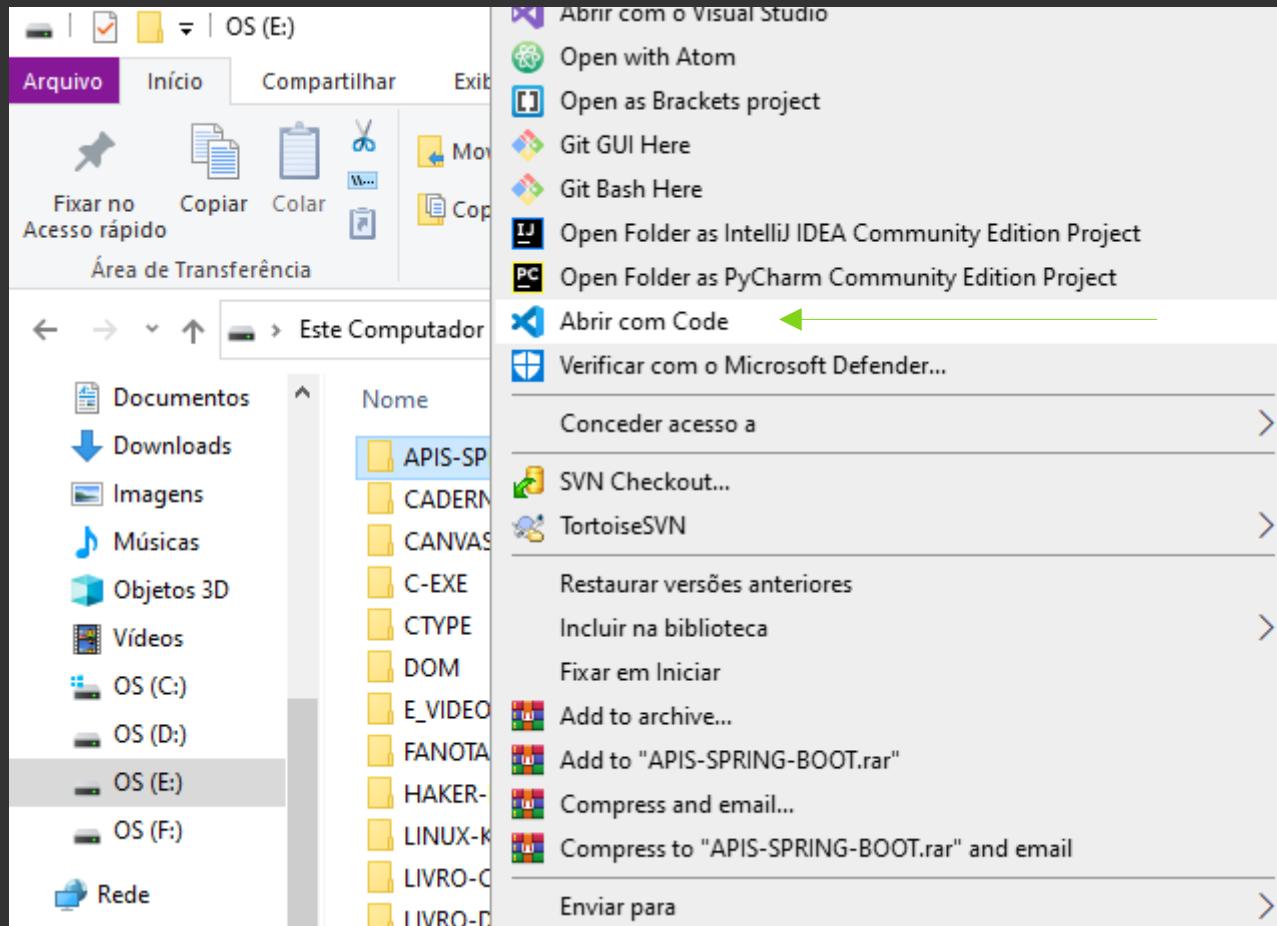
install never later

Ln 22, Col 56 Tab Size: 4 UTF-8 { XML Go Live

Vemos se ele tem essas dependencias. Se não tive adicionamos e depois clicamos ali

**Essa solução não funcionou nem tirando a versão da dependencia  
Lendo a pagina mandaram fechar o spring e abrir novamente vamos tentar fechar  
e reabrir**

# Fechei e abri de novo o vscode



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under 'APIS-SPRING-BOOT'. The 'src' folder contains 'main' and 'test'. 'main' has 'java', 'br', 'com', 'projeto', 'api', 'controle', 'modelo', and 'repositorio' subfolders. 'repositorio' contains 'Repositorio.java' and 'ApiApplication.java'. 'resources' contains 'static', 'templates', and 'application.properties'. Other files shown include '.gitignore', 'HELP.md', 'mvnw', 'mvnw.cmd', 'pom.xml', and 'README.md'.
- Code Editor:** Displays the content of 'Repositorio.java'. The code defines a public interface 'Repositorio' that extends 'CrudRepository<Pessoa, Integer>'. The code is as follows:

```
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.stereotype.Repository;
5
6 import br.com.projeto.api.modelo.Pessoa;
7
8 @Repository
9 public interface Repositorio extends CrudRepository<Pessoa, Integer>{
10
11 }
```

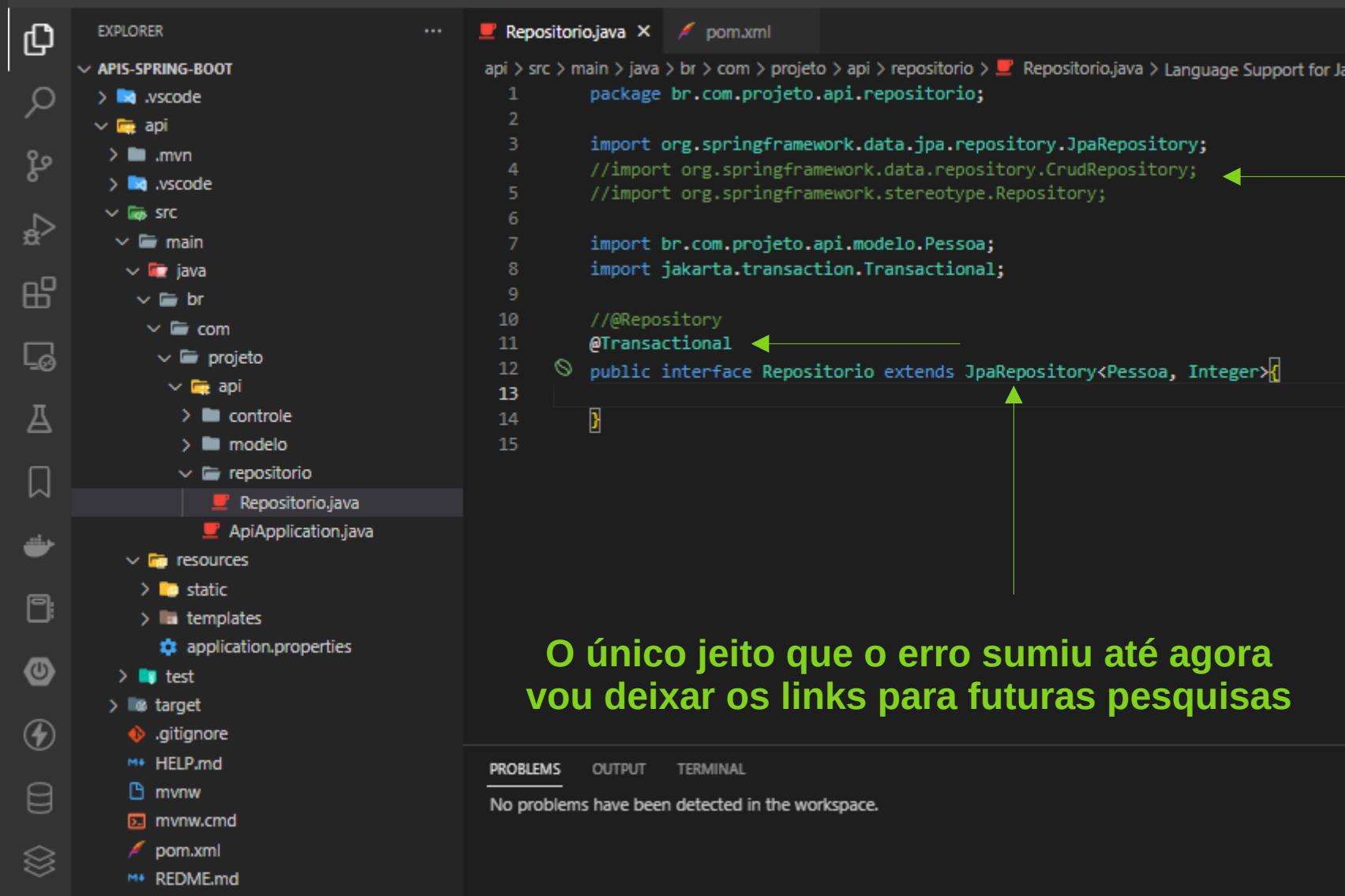
- Terminal:** Shows a PowerShell session with the following output:

```
Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6
-
PS E:\APIS-SPRING-BOOT>
* History restored
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

```
Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6
PS E:\APIS-SPRING-BOOT>
```
- Status Bar:** Ln 12, Col 1, Spaces: 4, UTF-8, CRLF, Java, Go Live, and other icons.

**E o erro sumiu voltou o erro**



<https://spring.io/blog/2011/02/10/getting-started-with-spring-data-jpa>

<https://www.youtube.com/watch?v=ReZG1Cv60Bs>

Podemos ir na pagina do tutorial e na pagina da api  
Qualquer erro podemos voltar a versao antiga mas o próprio professor disse não existir  
problemas

**#15**  
**ANNOTATION @Autowired**

**Pra que ela serve a @AutoWired?**

**Para o desenvolvedor não precisar instanciar objetos pra ter acesso a determinados  
metodos isso é chamado de injeção de dependencias**

# Adicione o @AutoWired e seu import

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including .vscode, api, .mvn, .vscode, src, main, java, br, com, projeto, api, controle, Controle.java, modelo, repositorio, Repositorio.java, ApiApplication.java, resources, static, templates, application.properties, test, target, .gitignore, and HPI P.mld.
- Code Editor:** The file Controle.java is open. The code defines a REST controller with a single endpoint that returns a welcome message. A tooltip for the @Auto annotation is displayed, providing information about its purpose and usage.
- Completion Panel:** A floating panel shows the tooltip for the @Auto annotation, which includes:
  - Auto-configuration specific variant of Spring Framework's @Order annotation. Allows auto-configuration classes to be ordered among themselves without affecting the order of configuration classes passed to AnnotationConfigApplicationContext.register(Class).
  - As with standard @Configuration classes, the order in which auto-configuration classes are applied only affects the order in which their beans are defined. The order in which those beans are subsequently created is unaffected and is determined by each bean's dependencies and any @DependsOn relationships.

```
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.RestController;

import br.com.projeto.api.modelo.Pessoa;

@RestController
public class Controle {

    @AutoConfigureOrder - org.springframework.boot.autoconfigure.AutoConfigureOrder
    @Autowired - org.springframework.beans.factory.annotation.Autowired
    @AutoConfiguration - org.springframework.boot.autoconfigure.AutoConfiguration
    @AutoConfigurationPackage - org.springframework.boot.autoconfigure.AutoConfigurationPackage
    @AutoConfigureAfter - org.springframework.boot.autoconfigure.AutoConfigureAfter
    @AutoConfigureBefore - org.springframework.boot.autoconfigure.AutoConfigureBefore
    @NoAutoStart - ch.qos.logback.core.joran.spi.NoAutoStart
    @JsonAutoDetect - com.fasterxml.jackson.annotation.JsonAutoDetect
    @AttributeOverride - jakarta.persistence.AttributeOverride
    @AttributeOverrides - jakarta.persistence.AttributeOverrides
    @EnableAutoConfiguration - org.springframework.boot.autoconfigure.EnableAutoConfiguration
    @ImportAutoConfiguration - org.springframework.boot.autoconfigure.ImportAutoConfiguration
    public String boasVindas(@PathVariable String nome){
        return "Seja bem vindo(a) " + nome;
    }

    @PostMapping("/pessoa")
    public Pessoa pessoa(@RequestBody Pessoa p){
        return null;
    }
}
```

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code



REPOSITORY Repository.java Controle.java ●

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > mensagem  
1 package br.com.projeto.api.controle;  
2  
3 import org.springframework.beans.factory.annotation.Autowired; ←  
4 import org.springframework.web.bind.annotation.GetMapping;  
5 import org.springframework.web.bind.annotation.PathVariable;  
6 import org.springframework.web.bind.annotation.PostMapping;  
7 import org.springframework.web.bind.annotation.RequestBody;  
8 import org.springframework.web.bind.annotation.RestController;  
9  
10 import br.com.projeto.api.modelo.Pessoa;  
11  
12 @RestController  
13 public class Controle {  
14     @Autowired  
15     @GetMapping("/")  
16     public String mensagem(){  
17         return "Hello World";  
18     }  
19 }  
20
```

File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
          - controle
            - Controle.java
- modelo
- repositorio
  - Repositorio.java
  - ApiApplication.java
- resources
  - static
  - templates
    - application.properties
- test
- target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Repositorio.java Controle.java 2

```
2
3     import org.springframework.beans.factory.annotation.Autowired;
4     import org.springframework.web.bind.annotation.GetMapping;
5     import org.springframework.web.bind.annotation.PathVariable;
6     import org.springframework.web.bind.annotation.PostMapping;
7     import org.springframework.web.bind.annotation.RequestBody;
8     import org.springframework.web.bind.annotation.RestController;
9
10    import br.com.projeto.api.modelo.Pessoa;
11
12    @RestController
13    public class Controle {
14
15        @Autowired
16        private Repositorio
17        @GetMapping("/")
18        public String me
19            return "Hell"
20        }
21
22        @GetMapping("/bo
23        public String bo
24            return "Seja"
25        }
26
27        @GetMapping("/bo
28        public String boasVindas(@PathVariable String nome){
29            return "Seja bem vindo(a) " + nome;
30
31    }
```

PROBLEMS 2 OUTPUT TERMINAL

Estamos criando um  
objeto de forma privada

org.springframework.data.repository.support.Repositories

Wrapper class to access repository instances obtained from a ListableBeanFactory.

- Author:

- Oliver Gierke
- Thomas Darimont
- Thomas Eizinger
- Christoph Strobl
- Alessandro Nistico
- Johannes Englmeier

CONSOLE

File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER Repository.java Controle.java 1 X

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle
- Controle.java
- modelo
- repositorio
- Repositorio.java
- ApiApplication.java

resources

- static
- templates
- application.properties

test

target

- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

PROBLEMS 1 OUTPUT TERMINAL

TERMINAL

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos os direitos reservados.  
Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>  
PS E:\APIS-SPRING-BOOT> [ ]

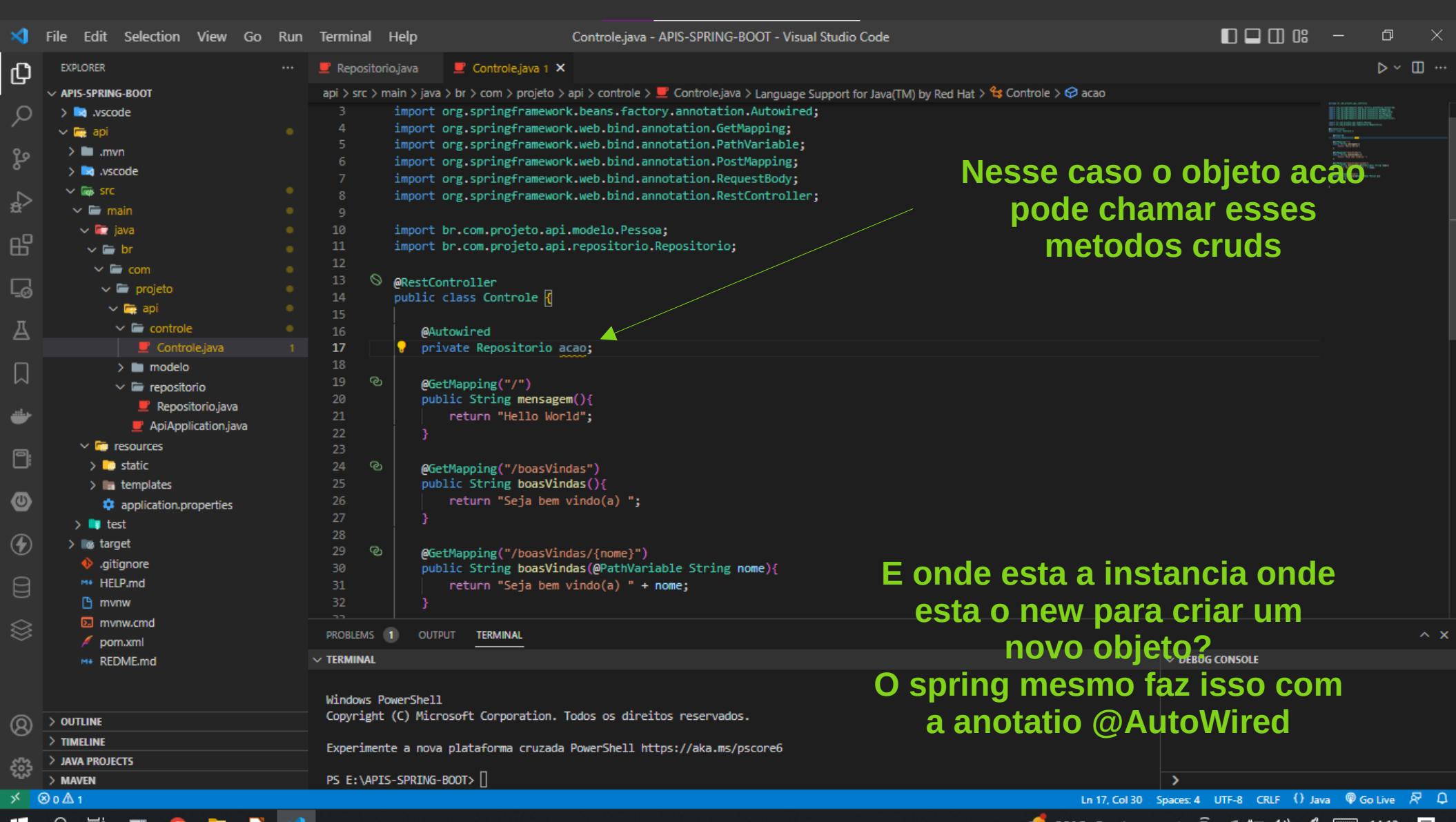
Ln 17, Col 30 Spaces: 4 UTF-8 CRLF Java Go Live

```
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio; ←
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao; ←
18
19     @GetMapping("/")
20     public String mensagem(){
21         return "Hello World";
22     }
23
24     @GetMapping("/boasVindas")
25     public String boasVindas(){
26         return "Seja bem vindo(a) ";
27     }
28
29     @GetMapping("/boasVindas/{nome}")
30     public String boasVindas(@PathVariable String nome){
31         return "Seja bem vindo(a) " + nome;
32     }
33 }
```

Criamos o objeto e adicionamos seu import

**Qual o tipo do objeto criado?  
É do tipo Repository**

**E porque tivemos que criar um obejto do tipo repositorio?  
Porque ele contem as ações do nosso banco de dados  
Quais são essas ações?  
Selecionar cadastrar deletar excluir updat o chamado crud**



Nesse caso o objeto acao pode chamar esses metodos cruds

E onde esta a instancia onde esta o new para criar um novo objeto?

O spring mesmo faz isso com a anotatio @AutoWired

```
File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code
```

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > acao
```

```
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repository.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao; // Autowired annotation
18
19     @GetMapping("/")
20     public String mensagem(){
21         return "Hello World";
22     }
23
24     @GetMapping("/boasVindas")
25     public String boasVindas(){
26         return "Seja bem vindo(a) ";
27     }
28
29     @GetMapping("/boasVindas/{nome}")
30     public String boasVindas(@PathVariable String nome){
31         return "Seja bem vindo(a) " + nome;
32     }
33 }
```

PROBLEMS 1 OUTPUT TERMINAL

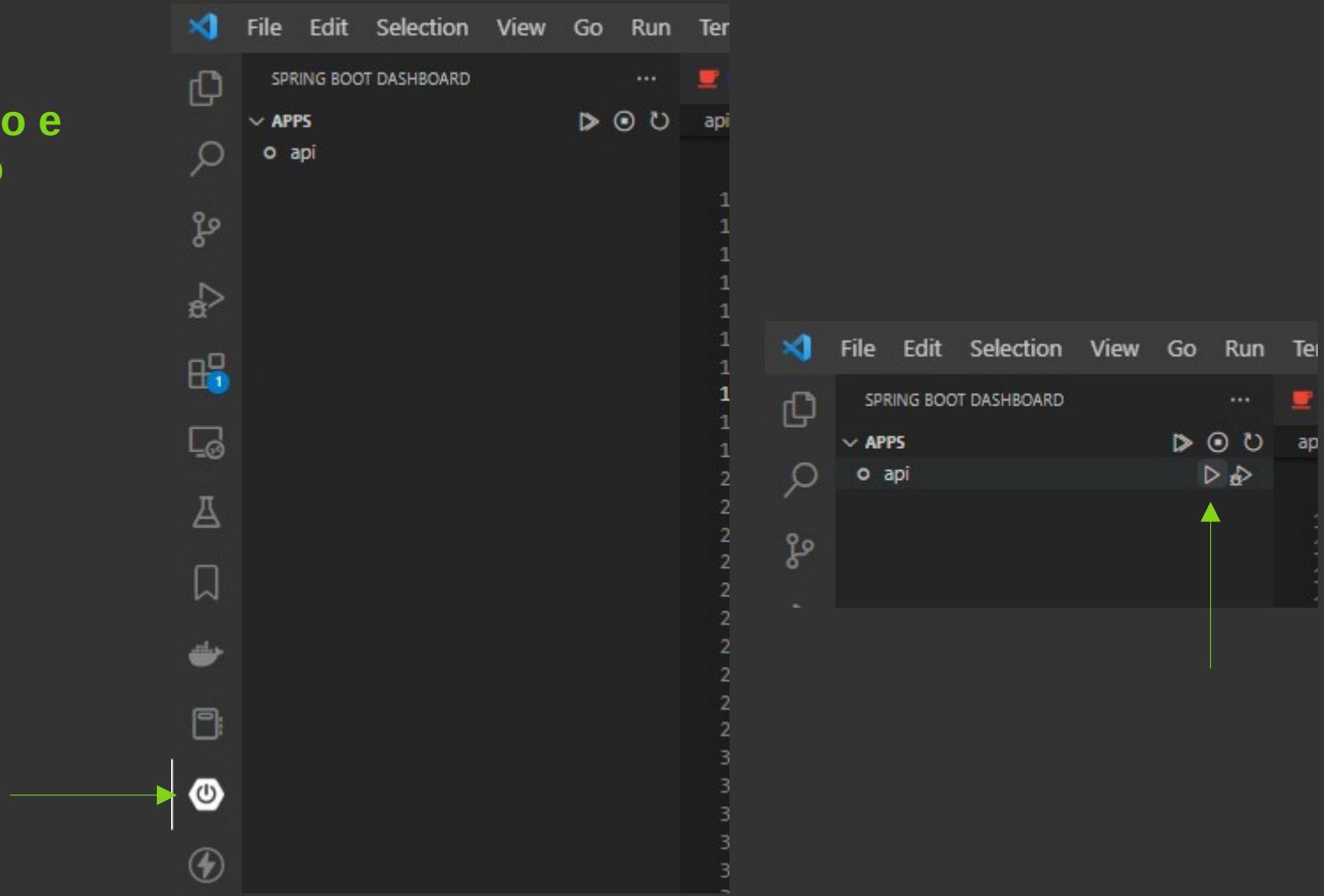
Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos os direitos reservados.  
Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

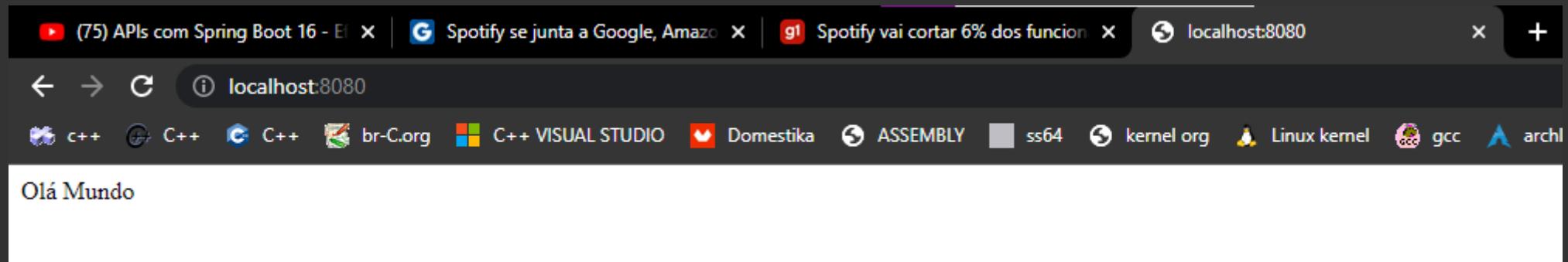
PS E:\APIS-SPRING-BOOT>

Ln 17, Col 30 Spaces: 4 UTF-8 CRLF Java Go Live

#16  
Efetuando cadastros com save

**Execute o projeto e  
depois abra o  
navegador**





O erro aparece porque criamos um objeto e ainda não utilizamos  
Vamos criar um metodo que sera nossa rota de pessoa

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
              - controle
    - resources
      - static
      - templates
      - application.properties
    - test
    - target
    - .gitignore
    - HELP.md
- Terminal:** Shows the command "Controle.java - APIS-SPRING-BOOT - Visual Studio Code".
- Code Editor:** Displays the "Controle.java" file with the following code:

```
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(){
20
21     }
22
23     @GetMapping("/")
24     public String mensagem(){
25         return "Hello World";
26     }
27
28     @GetMapping("/boasVindas")
29     public String boasVindas(){
30         return "Seja bem vindo(a) ";
31     }
32
33     @GetMapping("/boasVindas/{nome}")
34     public String boasVindas(@PathVariable String nome){
35         return "Seja bem vindo(a) " + nome;
36     }
}
```

A green arrow points from the text "Metodo criado" to the word "cadastrar" in the code editor.

Metodo criado

Vamos realizar um insert como temos que retornar uma pessoa  
Usamos o objeto criado acao e logo depois dele damos um ponto para acessar as  
propriedades  
E dentro das opções que aparecem escolhemos o save()

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
          - api
          - controle
  - modelo
  - repository
    - Repositorio.java
    - ApiApplication.java
  - resources
    - static
    - templates
    - application.properties
  - test
  - target
  - .gitignore
  - HELP.md

Repositorio.java Controle.java 2

```
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repository.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(){
20         return acao.save
21     }
22     @GetMapping("/")
23     public String mens
24         return "Hello"
25     }
26
27     @GetMapping("/boas")
28     public String boas
29         return "Seja b"
30     }
31
32     @GetMapping("/boasVindas/{nome}")
33     public String boasVindas(@PathVariable String nome){
34         return "Seja bem vindo(a) " + nome;
35     }
36
37 }
```

save(S entity) : S  
saveAndFlush(S entity) : S  
saveAll(Iterable<S> entities) : List<S>  
saveAllAndFlush(Iterable<S> entities) : List<S>  
count(Example<S> example) : long  
exists(Example<S> example) : boolean  
findBy(Example<S> example, Function<FetchableFlue...

findAll(Example<S> example) : List<S>  
findAll(Example<S> example, Pageable pageable) : ...  
findAll(Example<S> example, Sort sort) : List<S>  
findOne(Example<S> example) : Optional<S>  
findAll(Sort sort) : List<Pessoa>

CrudRepository.save(S entity) : S

Saves a given entity. Use the returned instance for further operations as the save operation might have changed the entity instance completely.

- Type Parameters:



- Parameters:

- entity must not be null.

- Returns:

- the saved entity; will never be null.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - API-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - API-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
      - br
        - com
          - projeto
          - api
          - controle
    - Controle.java
    - modelo
- Code Editor:** The Controle.java file is open, showing the following code:

```
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(){
20         return acao.save();
21     }
22
23     @GetMapping("/")
24     public String mensagem(){
```

Dentro passamos  
o objeto do tipo  
Pessoa

**Como eu pego essas informações do tipo Pessoa?**

# Adiconamos a anotation @RequestBody e seu import

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Editor:** Displays the file "Controle.java" with the following code:

```
api > src > main > java > br > com > projeto > api > controle > Controle.java
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(@RequestBody
20         return acao.save()
21     }
22
23     @GetMapping("/")
24     public String mensagem(){
25         return "Hello World";
26     }
27
28     @GetMapping("/boasVindas")
29     public String boasVindas(){
30         return "Seja bem vindo"
31     }
32 }
```

- IntelliJ IDEA-like Hover Documentation:** A tooltip appears over the `@RequestBody` annotation, providing the following information:
  - Annotation indicating a method parameter should be bound to the body of the web request. The body of the request is passed through an `HttpMessageConverter` to resolve the method argument depending on the content type of the request. Optionally, automatic validation can be applied by annotating the argument with `@Valid`.
  - Supported for annotated handler methods.

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code

REPOSITORY

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
        - api
          - controle
  - resources
  - static

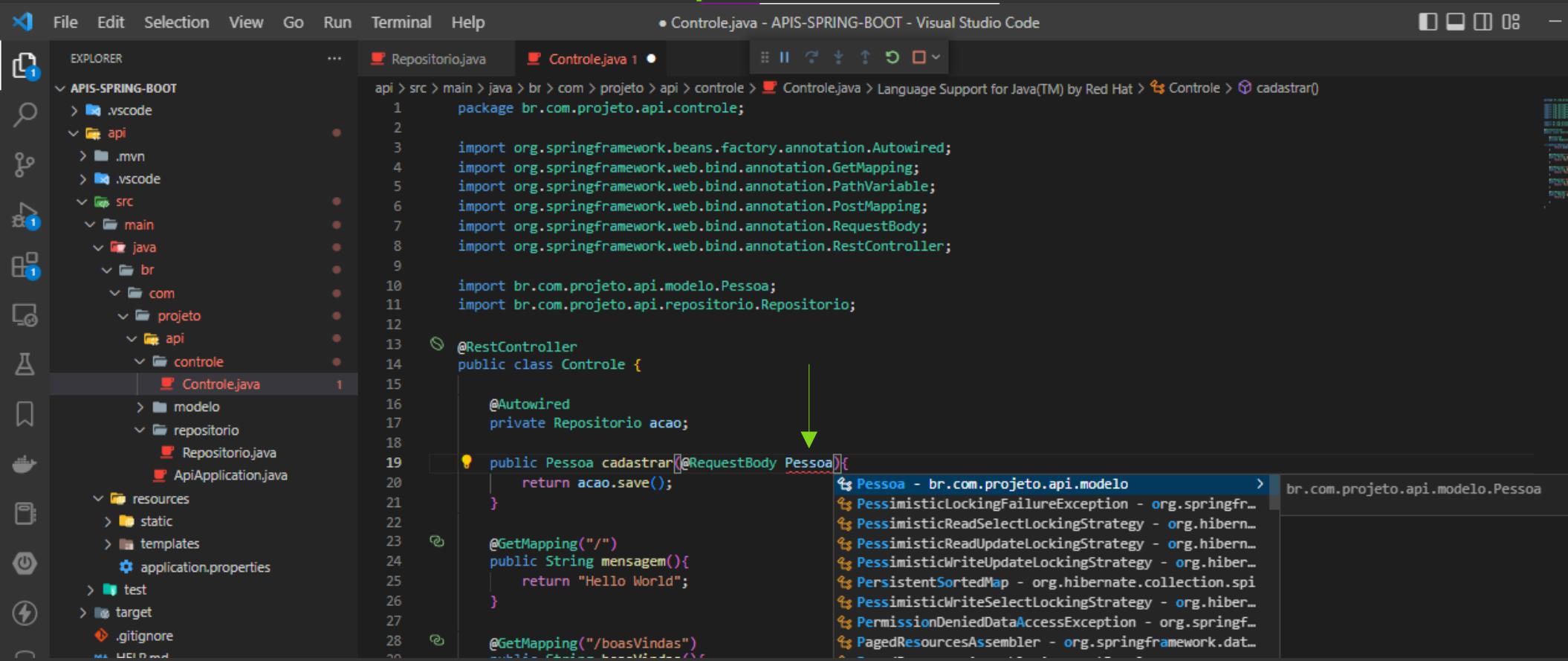
REPOSITORY

Controle.java 1 X

Language Support

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(@RequestBody){
20         return acao.save();
21     }
22 }
```

# Especificamos que estamos esperando uma requisição via body do tipo Pessoa



File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
          - api
          - controle
        - Controle.java
  - modelos
  - repositorio
    - Repositorio.java
    - ApiApplication.java
  - resources
    - static
    - templates
    - application.properties
  - test
  - target
  - .gitignore

REPOSITORY

CONTROLS

PROBLEMS

CONTROLE

CONTROLE.java 1

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > cadastrar()
```

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(@RequestBody Pessoa p) {
20         return acao.save();
21     }
22
23     @GetMapping("/")
24     public String mensagem(){
25         return "Hello World";
26     }
27
28     @GetMapping("/boasVindas")
29 }
```

br.com.projeto.api.modelo.Pessoa

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
          - projeto
        - api
          - controle
            - Controle.java
    - resources
    - static
- Editor View:** Controle.java 1 X (active tab).

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa; ←
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(@RequestBody Pessoa obj){↑
20         return acao.save();↑
21     }
22 }
```

A yellow lightbulb icon is shown above the line "public Pessoa cadastrar(@RequestBody Pessoa obj){". A green arrow points upwards from the line "return acao.save();".

E damos o nome de obj

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT". The "src" folder contains "main", "java", "br", "com", "projeto", "api", and "controle". Inside "controle", there is a file named "Controle.java".
- Editor (Center):** Displays the code for "Controle.java". The code is a Spring REST controller that interacts with a repository to save a Pessoa object.

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     public Pessoa cadastrar(@RequestBody Pessoa obj) {
20         return acao.save(obj);
21     }
22 }
```

- Bottom Status Bar:** Shows the status "Controle.java - APIS-SPRING-BOOT - visual studio".

O **save** efetua o cadastramento do objeto do tipo **Pessoa**

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- > api
  - .mvn
  - .vscode
- > src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - > modelo
  - > repositorio
    - Repositorio.java
  - > ApiApplication.java

resources

  - > static
  - > templates
  - > application.properties

test

Repositorio.java Controle.java 1

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > cadastrar(Pesso...
```

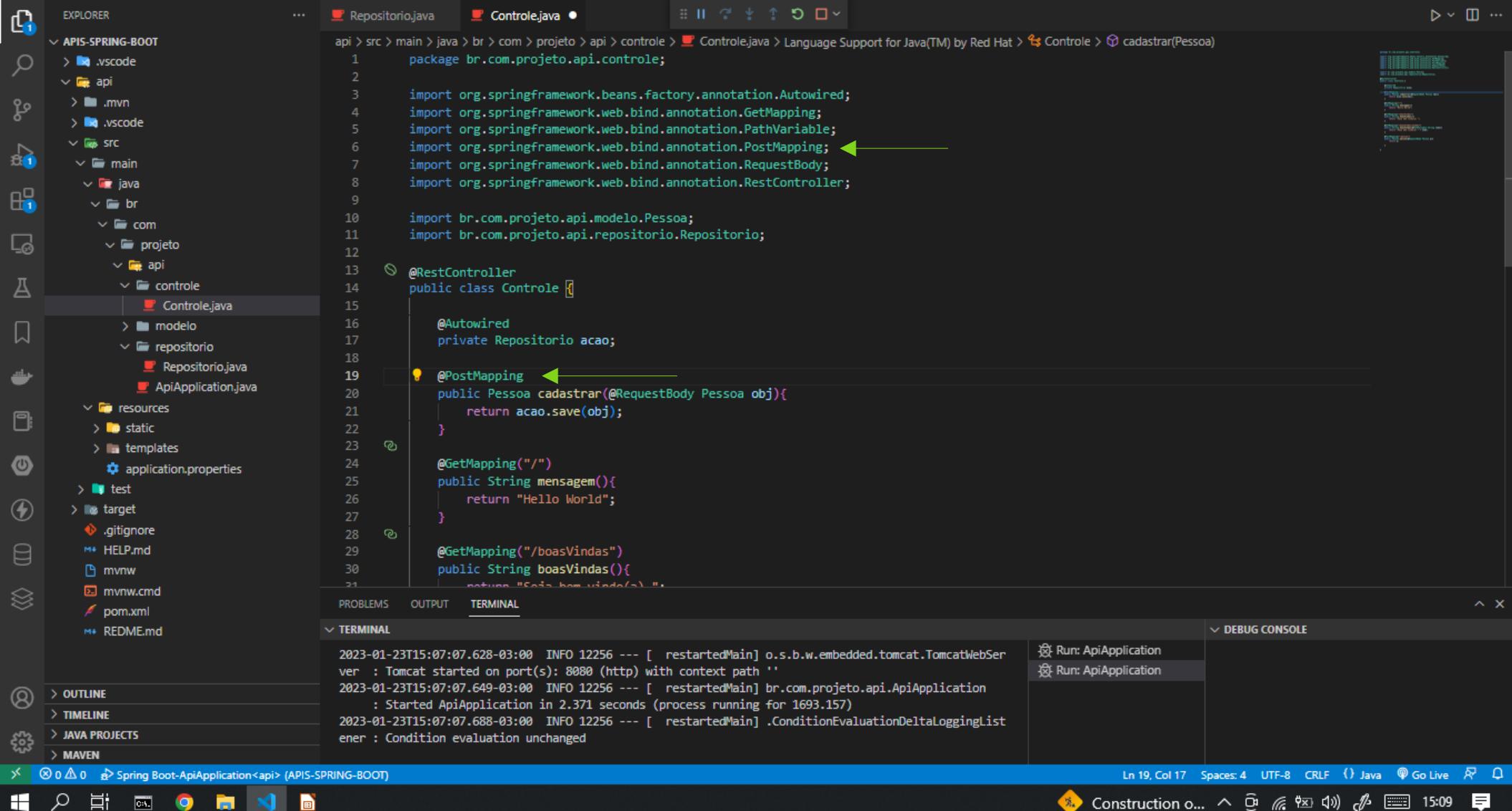
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     @PostMapping
20     public void PostMapping() {
21         PostConstruct();
22     }
23 }
24
25
26
```

Annotation for mapping HTTP POST requests onto specific handler methods.

Specifically, `@PostMapping` is a composed annotation that acts as a shortcut for `@RequestMapping(method = RequestMethod.POST)`.

  - Since:
    - 4.3
  - Author:
    - Sam Brannen
  - See Also:
    - GetMapping
    - PutMapping

Adidionamos a anotation `@PostMapping` e seu import para poder usar o `@RequestBody`



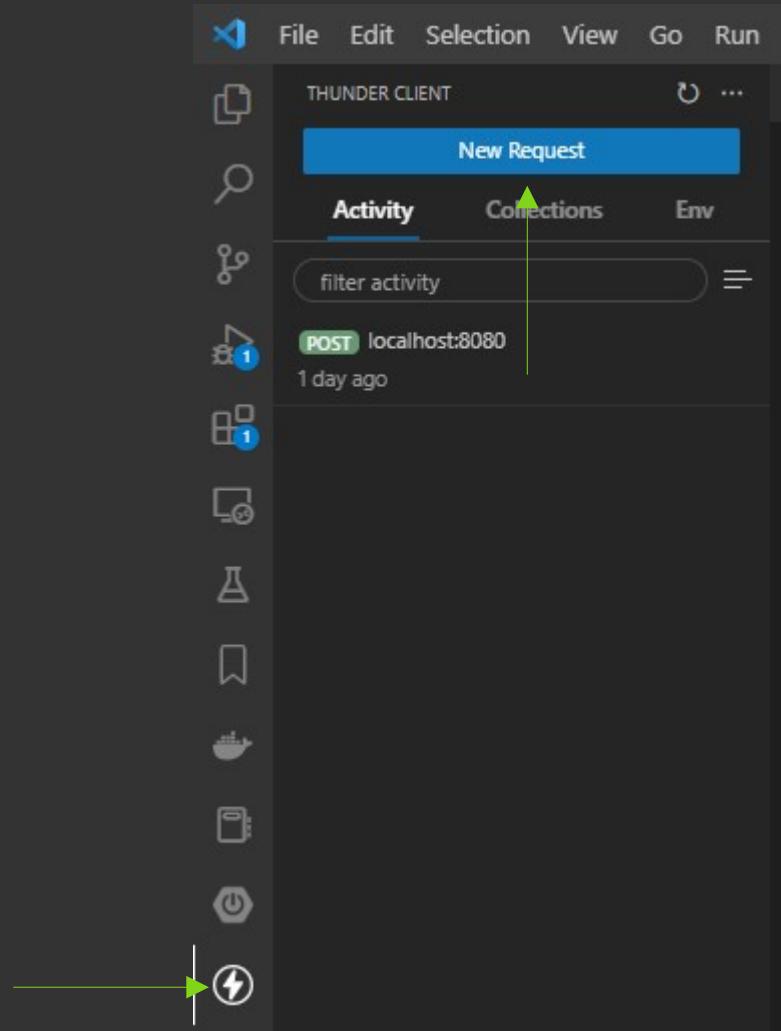
The screenshot shows the Visual Studio Code interface with the following details:

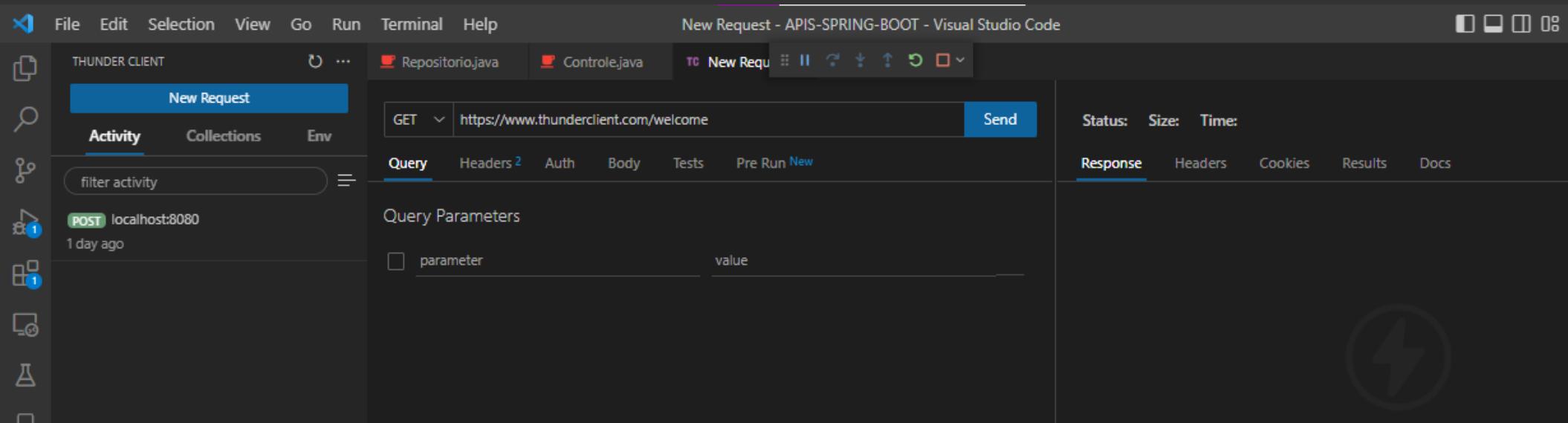
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
          - api
            - controle
    - resources
    - static
- Terminal View:** Shows the code for Controle.java:

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     @PostMapping("/api")
20     public Pessoa cadastrar(@RequestBody Pessoa obj){
21         return acao.save(obj);
22     }
23 }
```

Por fim colocamos a rota (“/api”)

**Salve abra o thunder client e faça uma nova requisição**





File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Repository.java Controle.java New Requ ...

New Request

Activity Collections Env

filter activity

POST localhost:8080 1 day ago

POST http://localhost:8080/api

Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Headers 2 Auth Body Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Format

Send Request Ctrl + Enter

Import Curl Ctrl + U

Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
2023-01-23T15:10:59.760-03:00 INFO 12256 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-01-23T15:10:59.842-03:00 INFO 12256 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApplication in 3.557 seconds (process running for 1925.349)
2023-01-23T15:10:59.866-03:00 INFO 12256 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
```

DEBUG CONSOLE

Run: ApiApplication

Run: ApiApplication

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Go Live

Chuva chegando

File Edit Selection View Go Run Terminal Help

New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080/api just now

POST localhost:8080 1 day ago

Repositorio.java Controle.java New Request

POST http://localhost:8080/api Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode GraphQL Binary

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "idade": 40  
4 }
```

Send Request Ctrl Enter

crie um objeto json

**Não estamos fazendo validação de dados  
O que pe validação de dados?**

**É quando uma pessoa digita algo incorreto e fazemos uma verificação pra saber se  
os dados estão ok**

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Repository.java Controle.java New Requ ...

New Request

Activity Collections Env

filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Response Headers Cookies Results Docs

Status: Size: Time:

Json Content Format

Json XML Text Form Form-encode GraphQL Binary

Json Content

```
1 {  
2   "nome": "Cristiano",  
3   "idade": 40  
4 }
```

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080/api 3 mins ago

POST localhost:8080 1 day ago

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content Format

```
1 {
2   "nome": "Cristiano",
3   "idade": 40
4 }
```

Status: 200 OK Size: 44 Bytes Time: 1.89 s

Response Headers 4 Cookies Results Docs

```
1 {
2   "codigo": 1,
3   "nome": "Cristiano",
4   "idade": "40"
5 }
```

Se isso apareceu significa que os dados estão corretos va no mysql pra ver os dados

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
2023-01-23T15:20:43.133-03:00 INFO 12256 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat-1].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-23T15:20:43.135-03:00 INFO 12256 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
: Initializing Servlet 'dispatcherServlet'
2023-01-23T15:20:43.141-03:00 INFO 12256 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
: Completed initialization in 5 ms
```

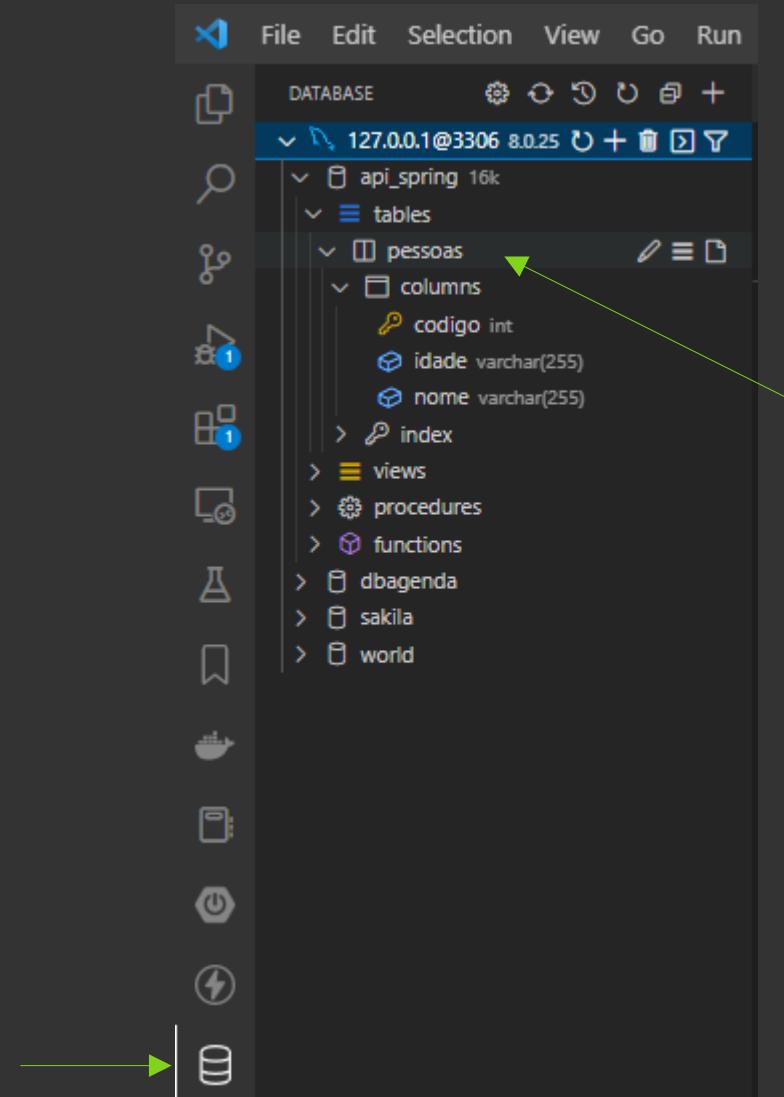
DEBUG CONSOLE

```
Run: ApiApplication
Run: ApiApplication
```

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Chuva chegando

Go Live



Click duplo em pessoas



File Edit Selection View Go Run Terminal Help

pessoas - APIS-SPRING-BOOT - Visual Studio Code



DATABASE



127.0.0.1@3306 8.0.25



api\_spring 16k



tables (1)



> pessoas



> views

> procedures

> functions

> dbagenda

> sakila

> world



Repository.java

Controle.java

New Requ

```
SELECT * FROM pessoas LIMIT 100;
```

Input to filter result

Free

1

+

-

↶

↑

↓

↷

Cost: 90ms

1 Total 1

|  | codigo | idade        | nome         |
|--|--------|--------------|--------------|
|  | int    | varchar(255) | varchar(255) |

|   |   |    |           |
|---|---|----|-----------|
| 1 | 1 | 40 | Cristiano |
|---|---|----|-----------|

**Fazendo mais um**

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25 Repository.java Controle.java New Req

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "codigo": 2,  
3   "nome": "Joyce",  
4   "idade": "33"  
5 }
```

Status: 200 OK Size: 40 Bytes Time: 115 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "codigo": 2,  
3   "nome": "Joyce",  
4   "idade": "33"  
5 }
```

File Edit Selection View Go Run Terminal Help

pessoas - APIS-SPRING-BOOT - Visual Studio Code

DATABASE

127.0.0.1@3306 8.0.25

api\_spring 16k

tables (1)

pessoas

columns

codigo int  
idade varchar(255)  
nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world

Repository.java

Controle.java

New Req

SELECT \* FROM pessoas LIMIT 100;

Input to filter result

Free 1

Locked

\* codigo idade nome

int varchar(255) varchar(255)

|   | codigo | idade | nome      |
|---|--------|-------|-----------|
| 1 | 1      | 40    | Cristiano |
| 2 | 2      | 33    | Joyce     |

Cost: 8ms < 1 > Total 2

#17

Listando os dados com o comando `findAll()`

**Antes de trabalhar com o controle temos que especificar pro nosso crudrepository que tipo de dados ele deve retornar**

**O que faz o comando findAll()?**  
**Lista todos os dados de uma tabela**

The screenshot shows a Visual Studio Code interface with a database sidebar on the left and a code editor on the right.

**Database Sidebar:**

- Connected to `127.0.0.1@3306` (8.0.25)
- Schema: `api`
- Tables (1): `pessoas`
  - Columns:
    - `codigo` int
    - `idade` varchar(255)
    - `nome` varchar(255)
  - Index
  - Views
  - Procedures
  - Functions
- `dbagenda`
- `sakila`
- `world`

**Code Editor (Repository.java):**

```
package br.com.projeto.api.repository;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;

//@Repository
@Transactional
public interface Repository extends JpaRepository<Pessoa, Integer> {
```

A code completion dropdown is open at the cursor position, showing various `List` implementations from different packages:

- `List - java.util` (selected)
- `List - org.hibernate.mapping`
- `ListAppender - ch.qos.logback.core.read`
- `ListAttribute - jakarta.persistence.metamodel`
- `ListBinder - org.hibernate.cfg.annotations`
- `ListCellRenderer - javax.swing`
- `ListCrudRepository - org.springframework.data.repository`
- `ListDataEvent - javax.swing.event`
- `ListDataHandler - javax.swing.plaf.basic.BasicComponentUI`
- `ListDataHandler - javax.swing.plaf.basic.BasicContainerUI`
- `ListDataHandler - javax.swing.plaf.basic.BasicListUI`
- `ListDataListener - javax.swing.event`

**Right Panel (Info about `List`):**

An ordered collection (also known as a sequence). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.

Unlike sets, lists typically allow duplicate elements. More formally, lists typically allow pairs of elements `e1` and `e2` such that `e1.equals(e2)`, and they typically allow multiple null elements if they allow null elements at all. It is not inconceivable that someone might wish to implement a list that prohibits duplicates, by throwing runtime exceptions when the user attempts to insert them, but we expect this usage to be rare.

The `List` interface places additional stipulations, beyond those specified in the `Collection` interface, on the contracts of the `iterator`, `add`, `remove`, `equals`, and `hashCode` methods. Declarations for other inherited methods are also included here for convenience.

The `List` interface provides four methods for positional (indexed) access to list elements. Lists (like Java arrays) are zero based. Note that these operations may execute in time proportional to the index value for some implementations (the

Vamos sobreescrivar um metodo e como vamos retornar uma lista devemos importar o List do pacote java.util

The screenshot shows the Visual Studio Code interface with a database sidebar on the left and a code editor on the right.

**Database Sidebar:**

- Connected to **127.0.0.1@3306** (version 8.0.25).
- Selected database: **api\_spring** (16k).
- Tables (1): **pessoas**
  - Columns:
    - codigo** int
    - idade** varchar(255)
    - nome** varchar(255)
  - index**
  - views**
  - procedures**
  - functions**
- dbagenda**
- sakila**
- world**

**Code Editor:**

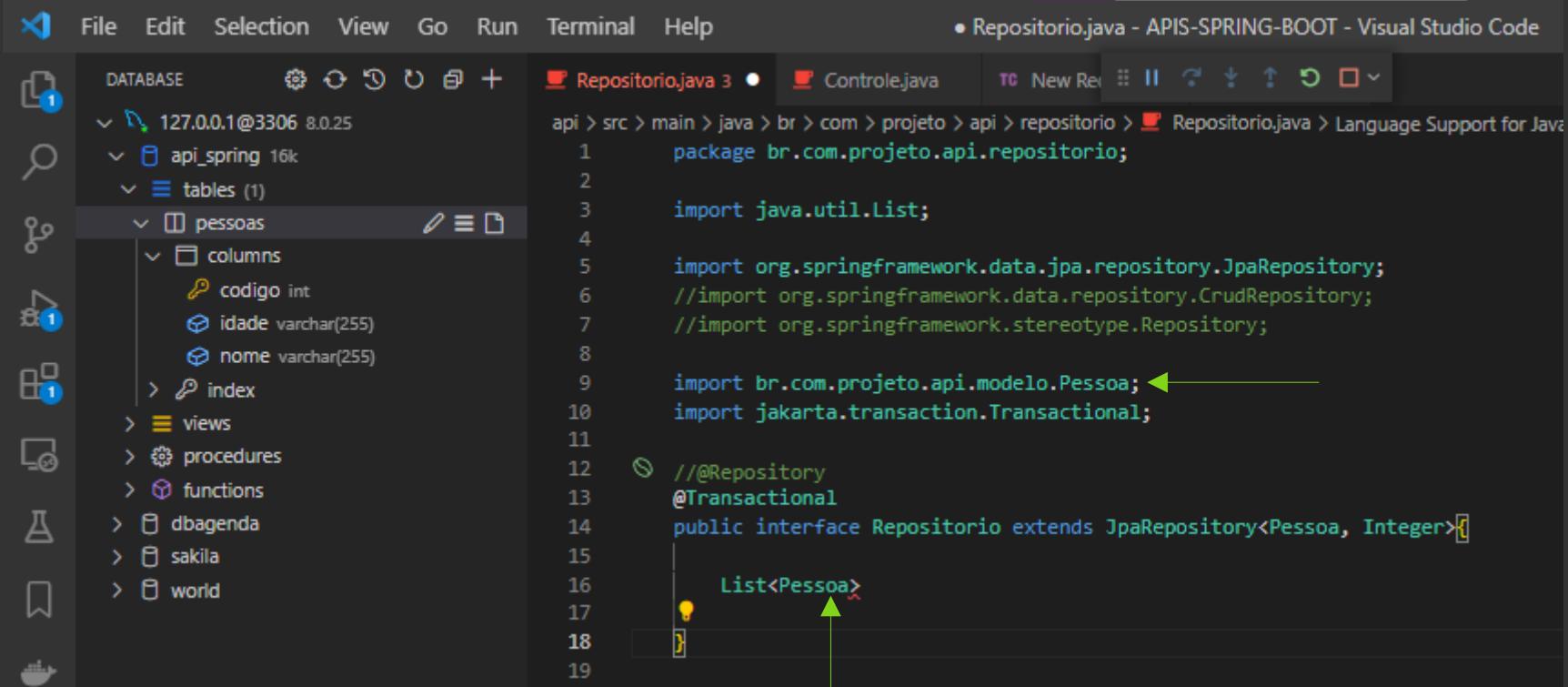
```
• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code
```

File Edit Selection View Go Run Terminal Help

```
Repositorio.java 3 • Controle.java New Re
```

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > •
```

```
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findByNome(String nome);
17 }
18
19 }
```



The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar titled "DATABASE" showing a connection to "127.0.0.1@3306 8.0.25" and a schema named "api\_spring". Inside "api\_spring", there is a table named "pessoas" with columns "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other items in the sidebar include "index", "views", "procedures", "functions", "dbagenda", "sakila", and "world". The main editor area has a title bar "Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code". The code itself is:

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa; ←
10 import jakarta.transaction.Transactional;
11
12 //Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> ↑
17 }
18
19 }
```

**Informamos o tipo de dados que queremos especificar  
Não esqueça do import**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Left Sidebar:** Shows a database connection to "127.0.0.1@3306" (8.0.25) with the schema "api\_spring" (16k). Under "api\_spring", there is a table named "pessoas" with columns "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other database objects like "index", "views", "procedures", and "functions" are also listed.
- Code Editor:** The file "Repositorio.java" is open, showing the following Java code:

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17 }
18
19 }
```

A yellow arrow points from the text "FindAll()" back up to the line "List<Pessoa> findAll();".

FindAll() é nativo do crud repository

**Salve e vamos pro controle criar uma rota pra ele**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Code Editor:** Repository.java (highlighted) and Controle.java 1 (background).
- Database Explorer:** Connected to 127.0.0.1@3306 8.0.25, showing tables (1): pessoas. The pessoas table has columns: codigo (int), idade (varchar(255)), nome (varchar(255)).
- Problems:** 1 error.
- Output:** 1 message.
- Terminal:** Shows logs from Tomcat and Spring Boot application startup.
- Search:** Shows the Java.util.List interface documentation.

**Java.util.List**

An ordered collection (also known as a sequence). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.

Unlike sets, lists typically allow duplicate elements. More formally, lists typically allow pairs of elements `e1` and `e2` such that `e1.equals(e2)`, and they typically allow multiple null elements if they allow null elements at all. It is not inconceivable that someone might wish to implement a list that prohibits duplicates, by throwing runtime exceptions when the user attempts to insert them, but we expect this usage to be rare.

The `List` interface places additional stipulations, beyond those specified in the `Collection` interface, on the contracts of the `iterator`, `add`, `remove`, `equals`, and `hashCode` methods. Declarations for other inherited methods are also included here for convenience.

The `List` interface provides four methods for positional (indexed) access to list elements. Lists (like Java arrays) are zero based. Note that these operations may execute in time proportional to the index value for some implementations (the `LinkedList` class, for example). Thus, iterating over the elements in a list is typically preferable to indexing through it if the caller does not know the implementation.

The `List` interface provides a special iterator, called a `ListIterator`, that allows element insertion and replacement, and bidirectional access in addition to the normal operations that the `Iterator` interface provides. A method is

**Não esqueça do import**

```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repository.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao;
18
19     @PostMapping("/api")
20     public Pessoa cadastrar(@RequestBody Pessoa obj){
21         return acao.save(obj);
22     }
23
24     public List<Pessoa> listar() {
25         return acao.listar();
26     }
27
28     @GetMapping("/api/{id}")
29     public Pessoa buscar(@PathVariable Long id) {
30         return acao.buscar(id);
31     }
32 }
```

PROBLEMS 1 OUTPUT TERMINAL DEBUG CO

2023-01-23T15:37:06.179-03:00 [Tomcat]: Tomcat started on port(s) :8080 [Tomcat]: Started Application in 3.823 seconds (process running for 3491704)



File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code



DATABASE



127.0.0.1@3306 8.0.25

api\_spring 16k

tables (1)

pessoas



columns

codigo int

idade varchar(255)

nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world

List porque  
vamos importar  
uma lista de  
pessoas

Repository.java

Controle.java 1

New Rec

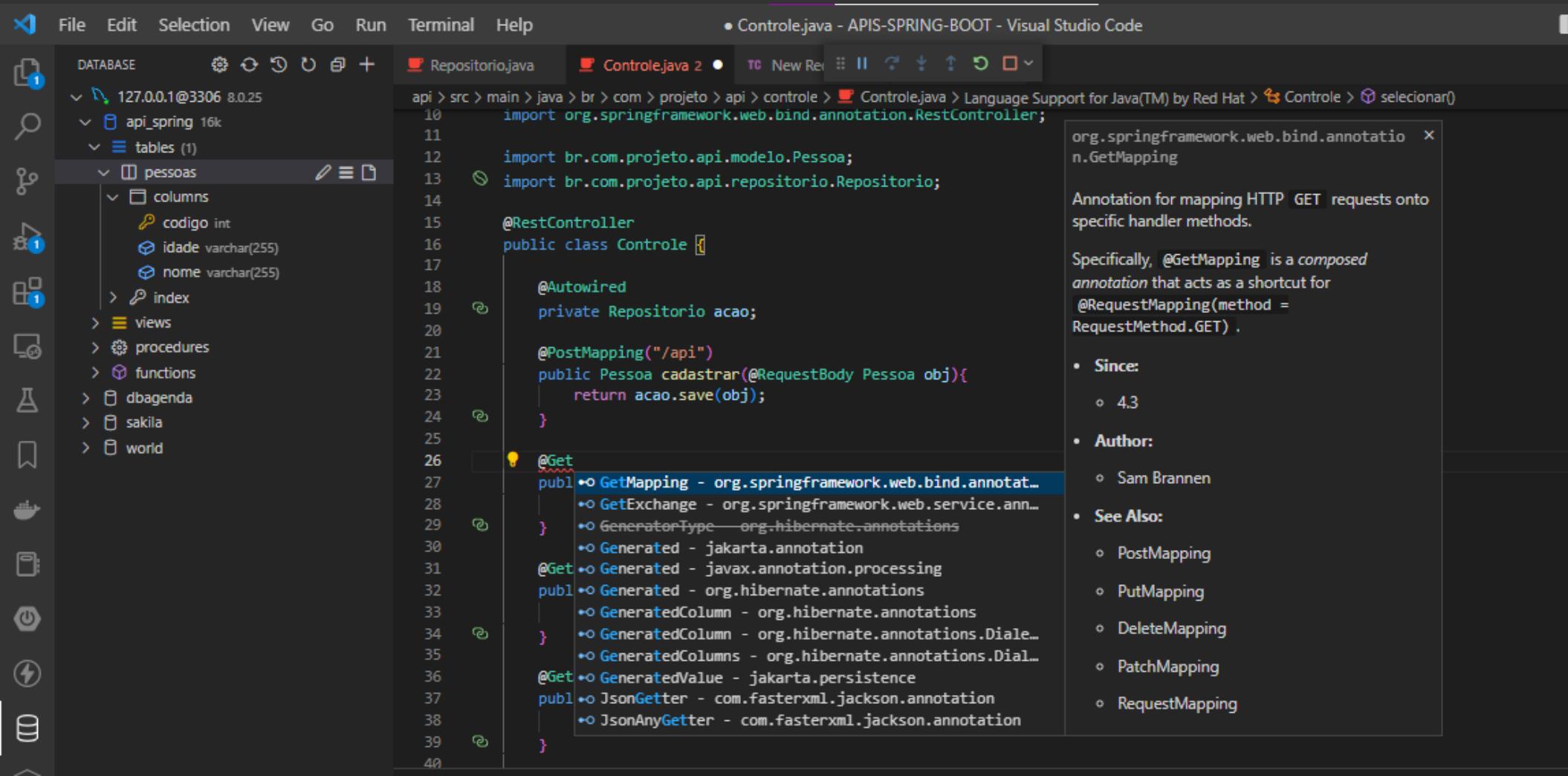


api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM)

```
1 package br.com.projeto.api.controle;  
2  
3 import java.util.List;  
4  
5 import org.springframework.beans.factory.annotation.Autowired;  
6 import org.springframework.web.bind.annotation.GetMapping;  
7 import org.springframework.web.bind.annotation.PathVariable;  
8 import org.springframework.web.bind.annotation.PostMapping;  
9 import org.springframework.web.bind.annotation.RequestBody;  
10 import org.springframework.web.bind.annotation.RestController;  
11  
12 import br.com.projeto.api.modelo.Pessoa;  
13 import br.com.projeto.api.repositorio.Repositorio;  
14  
15 @RestController  
16 public class Controle {  
17  
18     @Autowired  
19     private Repositorio acao;  
20  
21     @PostMapping("/api")  
22     public Pessoa cadastrar(@RequestBody Pessoa obj){  
23         return acao.save(obj);  
24     }  
25  
26     public List<Pessoa> selecionar(){  
27         return acao.findAll();  
28     }  
29 }
```

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar with various icons for file operations like opening, saving, and deleting files. The main area has a top navigation bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the navigation bar, there's a tab bar showing 'Controle.java - APIS-SPRING-BOOT - Visual Studio Code'. The main content area contains Java code for a controller class named 'Controle.java'.

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support  
10     import org.springframework.web.bind.annotation.RestController;  
11  
12     import br.com.projeto.api.modelo.Pessoa;  
13     import br.com.projeto.api.repositorio.Repositorio;  
14  
15     @RestController  
16     public class Controle {  
17  
18         @Autowired  
19         private Repositorio acao;  
20  
21         @PostMapping("/api")  
22         public Pessoa cadastrar(@RequestBody Pessoa obj){  
23             return acao.save(obj);  
24         }  
25  
26         public List<Pessoa> selecionar(){  
27             return acao.findAll();  
28         }  
29     }
```



The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar with various icons for file operations, a database connection, search, and other tools. The main area has a top navigation bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the navigation bar, the title bar indicates the current file is Controle.java - APIS-SPRING-BOOT - Visual Studio Code. The code editor displays Java code for a REST controller:

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java 1 • New Rec ⏪ ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ ⏻ ⏻
```

```
15     @RestController
16     public class Controle {
17
18         @Autowired
19         private Repositorio acao;
20
21         @PostMapping("/api")
22         public Pessoa cadastrar(@RequestBody Pessoa obj){
23             return acao.save(obj);
24         }
25
26         @GetMapping("/api") ←
27         public List<Pessoa> selecionar(){
28             return acao.findAll();
29         }
30     }
```

To the left of the code editor, there is a detailed view of the MySQL database structure. It shows a connection to 127.0.0.1@3306 (version 8.0.25). Under the 'api' schema, there is a table named 'pessoas'. The 'pessoas' table has three columns: 'codigo' (int), 'idade' (varchar(255)), and 'nome' (varchar(255)). There are also sections for 'index', 'views', 'procedures', 'functions', 'dbagenda', and 'sakila'.

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar with various icons for file operations like opening, saving, and running. The main area has a tab bar with 'Repositorio.java', 'Controle.java X', and 'New Requ'. Below the tabs is a status bar showing the path 'api > src > main > java > br > com > projeto > api > controle > Controle.java' and the message 'Language Support for Java(TM) by Red Hat'. The main content area displays the following Java code:

```
import br.com.projeto.api.repositorio.Repositorio;
import org.springframework.web.bind.annotation.*;

@RestController
public class Controle {

    @Autowired
    private Repositorio acao;

    @PostMapping("/api")
    public Pessoa cadastrar(@RequestBody Pessoa obj){
        return acao.save(obj);
    }

    @GetMapping("/api")
    public List<Pessoa> selecionar(){
        return acao.findAll();
    }
}
```



File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code



DATABASE



127.0.0.1@3306 8.0.25

api\_spring 16k

tables (1)

pessoas



columns

codigo int

idade varchar(255)

nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world



Só pra dar uma  
olhada de como  
esta até aqui

Repository.java

Controle.java X

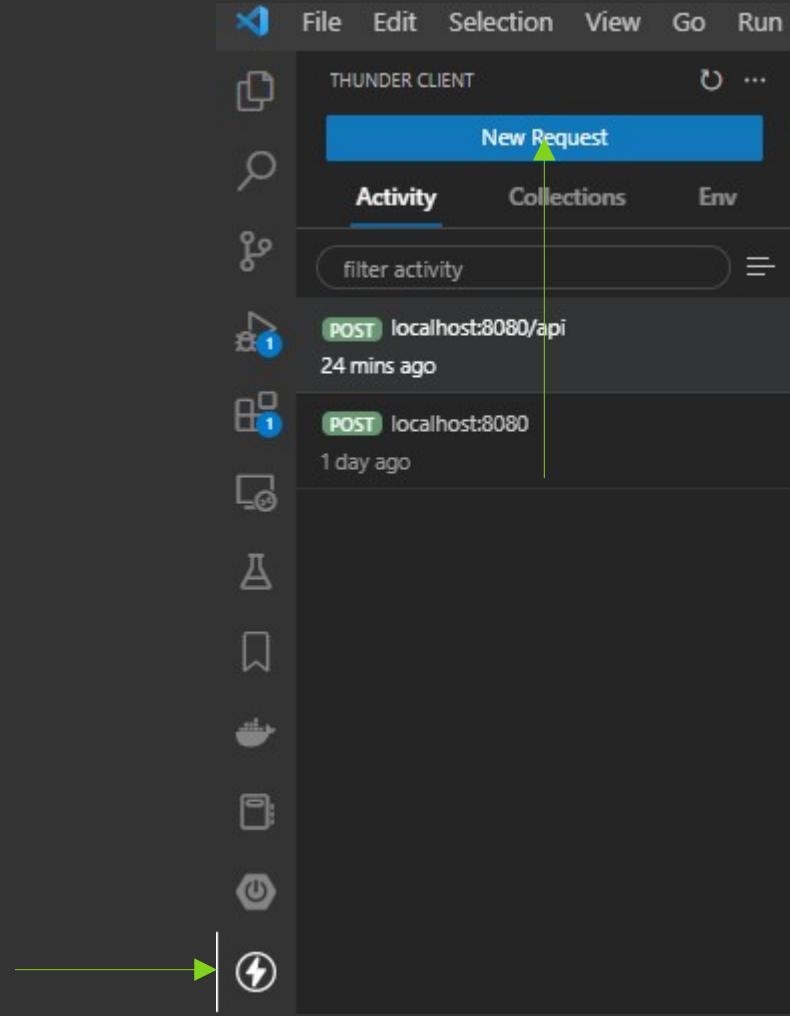
New Requ

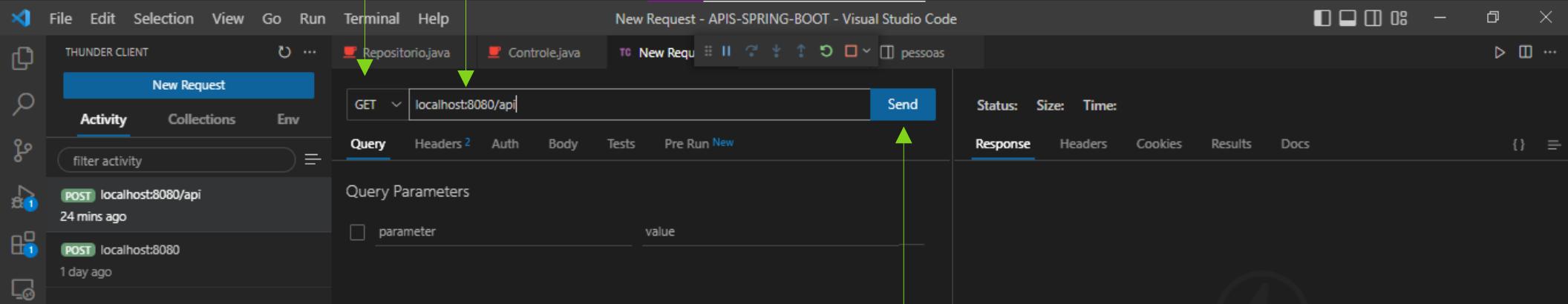


```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support
1     package br.com.projeto.api.controle;
2
3     import java.util.List;
4
5     import org.springframework.beans.factory.annotation.Autowired;
6     import org.springframework.web.bind.annotation.GetMapping;
7     import org.springframework.web.bind.annotation.PathVariable;
8     import org.springframework.web.bind.annotation.PostMapping;
9     import org.springframework.web.bind.annotation.RequestBody;
10    import org.springframework.web.bind.annotation.RestController;
11
12    import br.com.projeto.api.modelo.Pessoa;
13    import br.com.projeto.api.repositorio.Repositorio;
14
15    @RestController
16    public class Controle {
17
18        @Autowired
19        private Repositorio acao;
20
21        @PostMapping("/api")
22        public Pessoa cadastrar(@RequestBody Pessoa obj){
23            return acao.save(obj);
24        }
25
26        @GetMapping("/api")
27        public List<Pessoa> selecionar(){
28            return acao.findAll();
29        }
30
```

Agora quando realizarmos uma requisição get serão listados os dados da tabela de pessoas

Vamos no thunder client e realizamos uma nova requisição





THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/api just now

POST localhost:8080/api 24 mins ago

POST localhost:8080 1 day ago

Query Parameters

parameter value

Send

Status: 200 OK Size: 87 Bytes Time: 352 ms

Response Headers Cookies Results Docs

```
1 [  
2 {  
3   "codigo": 1,  
4   "nome": "Cristiano",  
5   "idade": "40"  
6 },  
7 {  
8   "codigo": 2,  
9   "nome": "Joyce",  
10  "idade": "33"  
11 }  
12 ]
```

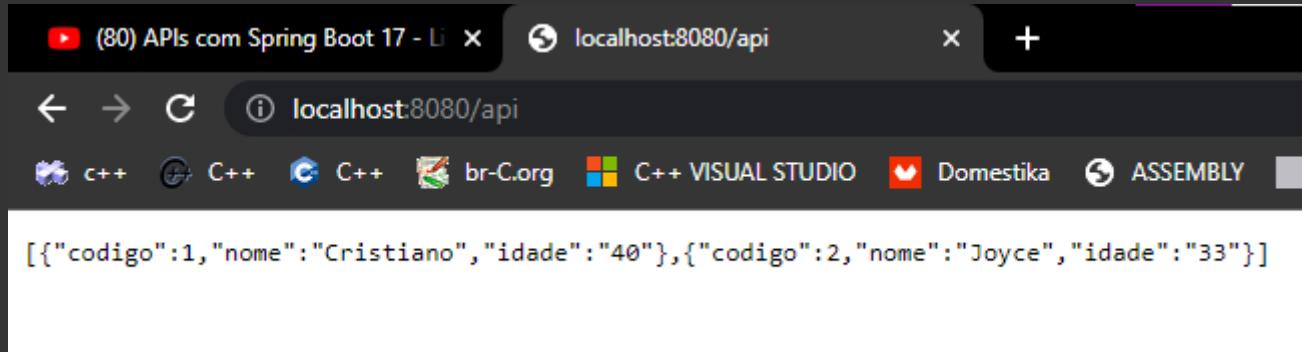
Dados passados

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, \*\*/node\_modules/\*\*)

**Da mesma forma os dados já são passados pelo navegador**



A screenshot of a browser window with a dark theme. The address bar shows the URL `localhost:8080/api`. The page content displays a JSON array:

```
[{"codigo":1,"nome":"Cristiano","idade":"40"}, {"codigo":2,"nome":"Joyce","idade":"33"}]
```

#18  
**Filtrando dado com findBy()**

O que faz o comando findBy()  
Posso procurar uma informação pela sua caracteristica

**Lembrando que podemos usar qualquer caracteristica que temos na classe**  
**Ex: codigo nome idade**

The screenshot shows the VS Code interface with the following details:

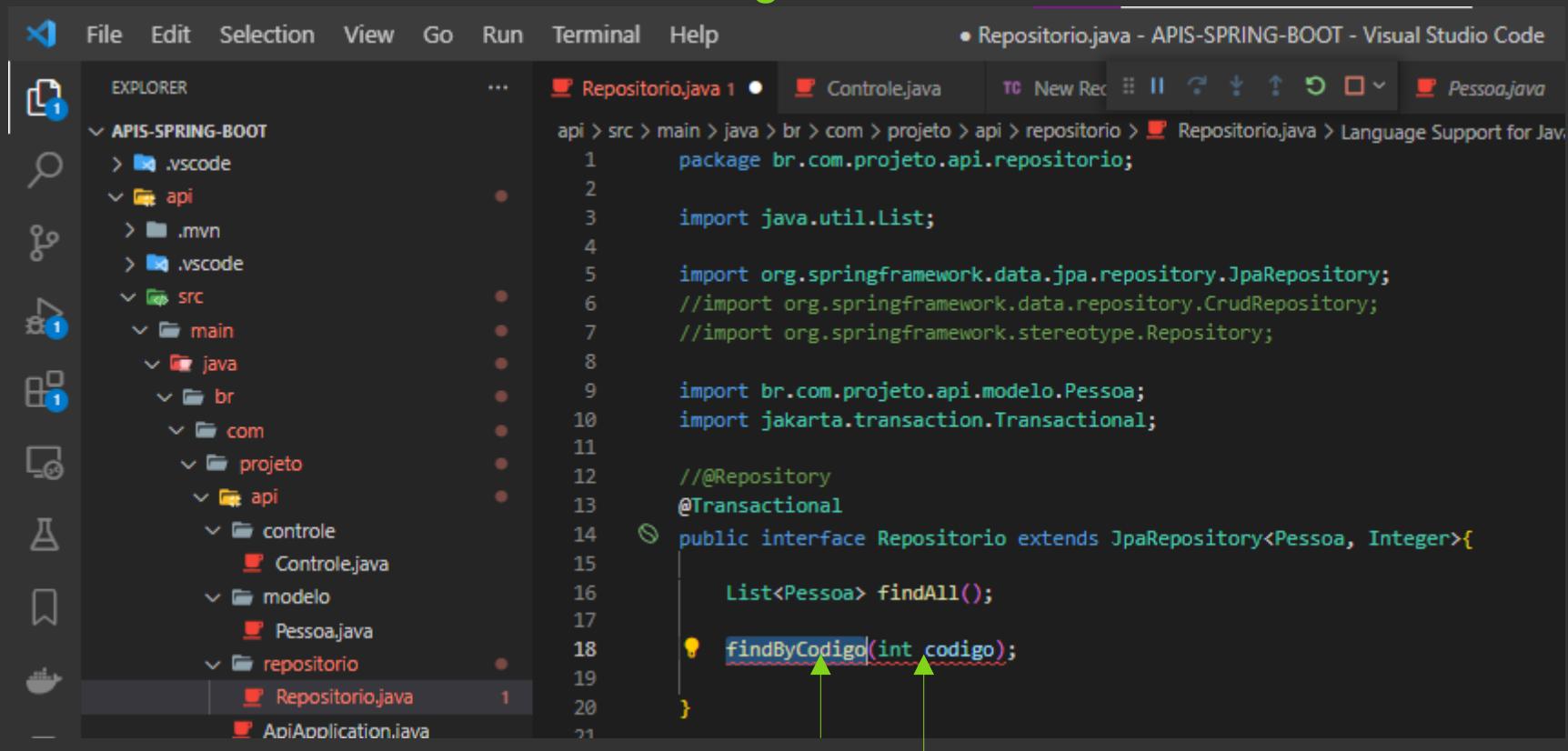
- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT".
  - Subfolders: ".vscode", "api", ".mvn", ".vscode", "src".
  - "src" folder contains:
    - "main" folder
      - "java" folder
        - "br" folder
          - "com" folder
            - "projeto" folder
              - "api" folder
                - "controle" folder
                  - "Controle.java"
                - "modelo" folder
                  - "Pessoajava.java"
            - "repository"
            - "ApiApplication.java"
  - Terminal (Top Right):** Displays the command "Pessoajava - APIS-SPRING-BOOT - Vis".
  - Code Editor (Right):** Shows the Java code for "Pessoajava.java".

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8
9 @Entity
10 @Table(name = "pessoas")
11 public class Pessoa {
12
13     // Atributos
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int codigo;
17     private String nome;
18     private String idade;
```

A cursor arrow points to the closing brace of the class definition at line 11.

Dai vem os dados ou eu posso trabalhar com  
findBy codigo  
findBy nome  
findBy idade  
Vou fazer um exemplo usando o codigo

## Passamos o nome do atributo e passamos um parametro int codigo



The screenshot shows the Visual Studio Code interface with the following details:

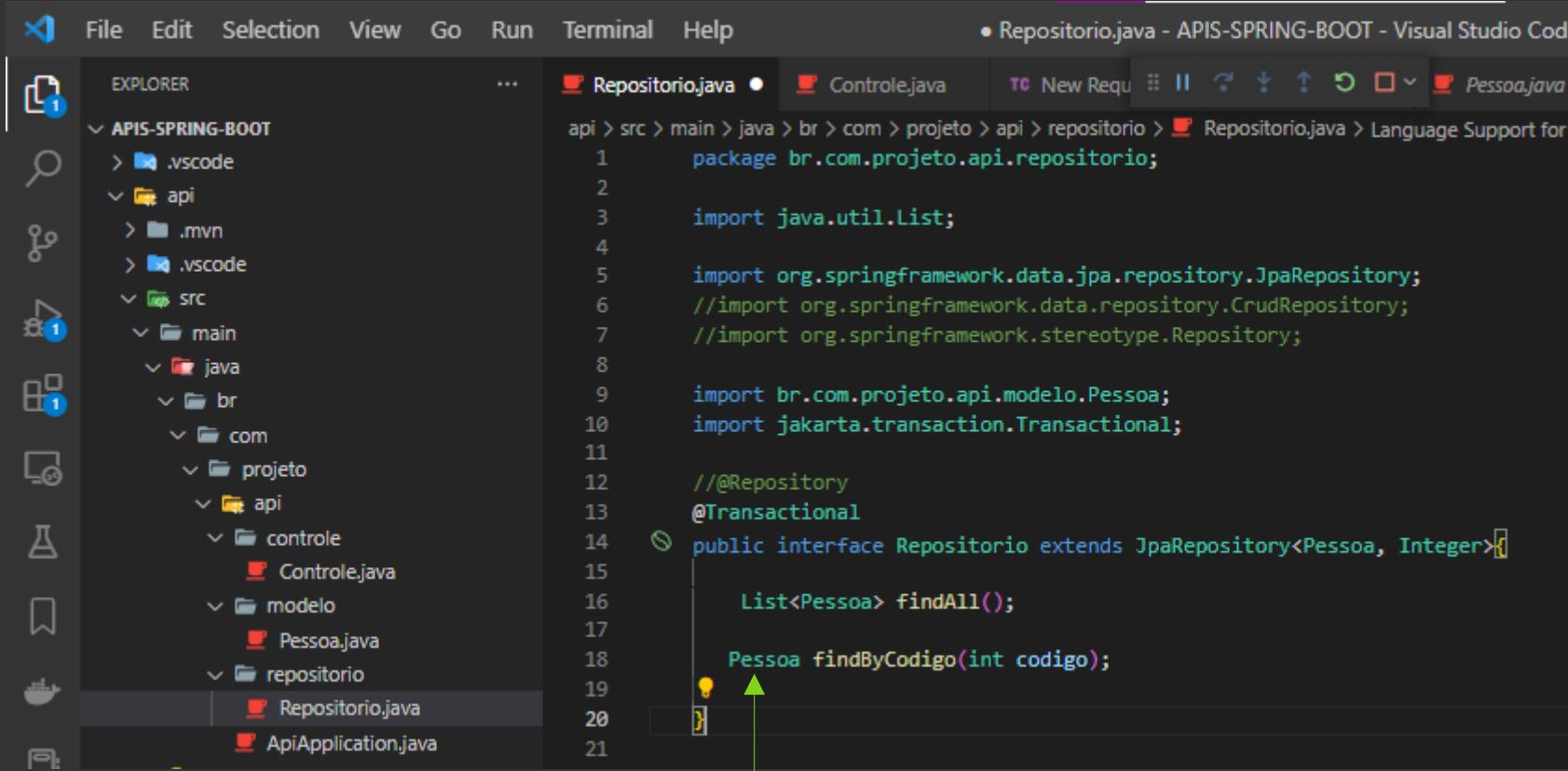
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle (Controle.java)
                - modelo (Pessoa.java)
  - Repository.java
  - ApiApplication.java
- Code Editor:** Displays the content of Repository.java:

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repository extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     findByCodigo(int codigo);
19
20 }
```

Annotations: `@Repository`, `@Transactional`, `findByCodigo`.

Visual elements: Two green arrows point upwards from the word "codigo" in the `findByCodigo` method signature to the parameter name in the code editor.

## Por fim o que o findByCodigo retorno uma Pessoa



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including .vscode, api, .mvn, .vscode, and src. The src folder contains main, java, br, com, projeto, api, controle, modelo, Pessoajava, and repositorio. The Repository.java file is selected in the repositorio folder.
- Code Editor:** Displays the code for Repository.java:

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repository extends JpaRepository<Pessoa, Integer> {
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20 }
```

A green arrow points from the text "Por fim o que o findByCodigo retorno uma Pessoa" at the top of the slide down to the "findByCodigo" method in the code editor.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT".
  - Root: .vscode, .mvn, .vscode
  - api: .vscode
  - src: .vscode, main
    - java: br
      - com: projeto, api
        - controle: Controle.java
        - modelo: Pessoa.java
      - repositorio: Repositorio.java
    - ApiApplication.java
- Code Editor (Right):** Displays the content of `Repositorio.java`.

```
package br.com.projeto.api.repositorio;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;
//@Repository
@Transactional
public interface Repositorio extends JpaRepository<Pessoa, Integer>{
    List<Pessoa> findAll();
    List<Pessoa> findByCodigo(int codigo);
}
```

A yellow arrow points upwards from the bottom of the code editor towards the closing brace of the interface definition.
- Top Bar:** Shows the title "Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code" and various icons for file operations.

Faça assim se o dado for retornar uma lista de pessoas ou seja se formos retornar mais de um registro

**Salve e va em controle para criarmos a rota**

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- ✓ api
  - > .mvn
  - > .vscode
- > src
  - ✓ main
    - > java
      - ✓ br
        - > com
          - > projeto
          - > api
          - > controle
        - > controle
          - Controle.java 2
  - > resources
  - > test
  - > target
  - ↳ .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

REPOSITORY.java Controle.java 2 New Re

api > src > main > java > br > com > projeto > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > selecionarPeloCodigo()

```
12 import br.com.projeto.api.modelo.Pessoa;
13 import br.com.projeto.api.repositorio.Repositorio;
14
15 @RestController
16 public class Controle {
17
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
31     public Pessoa selecionarPeloCodigo(){
32         return acao.
33             findByCodigo(int codigo) : Pessoa
34
35     @GetMapping("/")
36     public String me @getReferenceById(Integer id) : Pessoa
37         return "Hello";
38
39     @GetMapping("/bo")
40     public String bo @equals(Object arg0) : boolean
41
42     return "Seja"
```

PROBLEMS 2 OUTPUT TERMINAL

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

The screenshot shows a Java application named 'APIS-SPRING-BOOT' in Visual Studio Code. The 'src/main/java' directory contains a package 'br.com.projeto.api.control'. Inside this package, there is a 'controle' folder containing a 'Controle.java' file. This file contains code for a REST controller with methods to save a new person and to find all persons. A tooltip is displayed over the 'findByCodigo' method call, showing the parameter type 'int codigo' and the return type 'Pessoa'. A dropdown menu lists the available methods for the 'Repositorio' interface, including 'findById', 'getOne', 'getReferenceById', 'save', 'saveAndFlush', 'count', and 'findAll'. The status bar at the bottom indicates the current file is 'Controle.java' located at 'api/src/main/java(br/com/projeto/api/control/Controle.java)'.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
    - mvnw
  - Terminal:** api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java
  - Code Editor:** Controle.java (Line 1). The code is as follows:

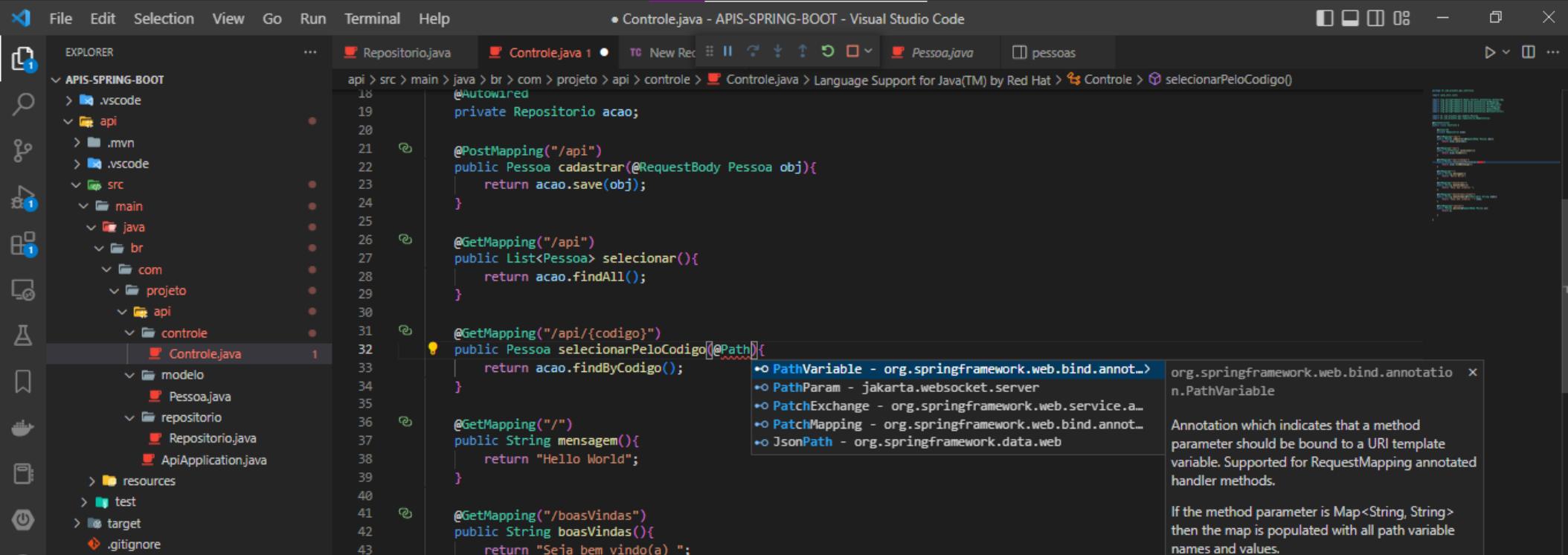
```
12     import br.com.projeto.api.modelo.Pessoa;
13     import br.com.projeto.api.repositorio.Repositorio;
14
15     @RestController
16     public class Controle {
17
18         @Autowired
19         private Repositorio acao;
20
21         @PostMapping("/api")
22         public Pessoa cadastrar(@RequestBody Pessoa obj){
23             return acao.save(obj);
24         }
25
26         @GetMapping("/api")
27         public List<Pessoa> selecionar(){
28             return acao.findAll();
29         }
30
31         public Pessoa selecionarPeloCodigo(){
32             return acao.findByCodigo();
33         }
34
35         @GetMapping("/")
36         public String mensagem(){
37             return "Hello World";
38         }
39
```

A red arrow points to the line "return acao.findByCodigo();".

Como eu pego esse código

Isso é um parametro

```
File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Co  
EXPLORER ... Repository.java Controle.java 1 • New Rec... Pess...  
APIS-SPRING-BOOT .vscode api .mvn .vscode src main java br com projeto api controle Controle.java 1  
18 @Autowired  
19 private Repositorio acao;  
20  
21 @PostMapping("/api")  
22 public Pessoa cadastrar(@RequestBody Pessoa obj){  
23     return acao.save(obj);  
24 }  
25  
26 @GetMapping("/api")  
27 public List<Pessoa> selecionar(){  
28     return acao.findAll();  
29 }  
30  
31 @GetMapping("/api/{codigo}")  
32 public Pessoa selecionarPeloCodigo(){  
33     return acao.findByCodigo();  
34 }  
35  
36 @GetMapping("/")  
37 public String mensagem(){  
38     return "Hello World";  
39 }
```



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
    - modelo
      - Pessoa.java
    - repository
- Terminal View:** Shows the current file path: api > src > main > java > br > com > projeto > api > controle > Controle.java.
- Code Editor:** The "Controle.java" file is open, showing the following Java code:

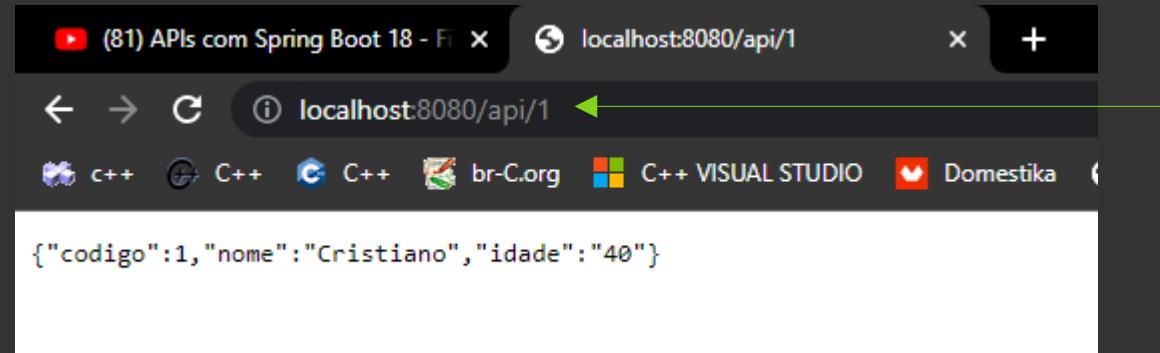
```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
31     @GetMapping("/api/{codigo}")
32     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
33         return acao.findByCodigo();
34     }
35 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT:
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                - Pessoa.java
  - Terminal View:** Shows the Java code for the Controle.java file, which contains three methods: cadastrar, selecionar, and selecionarPeloCodigo.

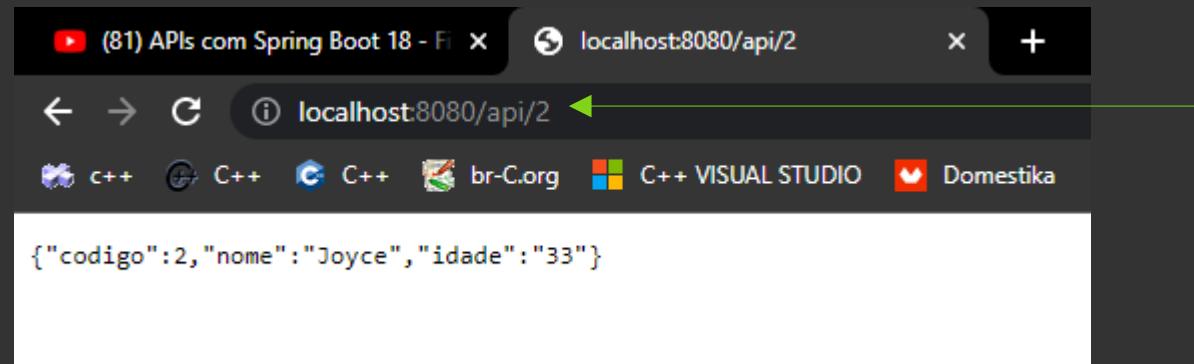
```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
31     @GetMapping("/api/{codigo}")
32     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
33         return acao.findByCodigo(codigo);
34     }
35 }
```

Salve e va na url testar



A screenshot of a web browser window. The title bar says "(81) APIs com Spring Boot 18 - Fi" and the address bar shows "localhost:8080/api/1". The main content area displays the following JSON object:

```
{"codigo":1,"nome":"Cristiano","idade":"40"}
```



A screenshot of a web browser window. The title bar says "(81) APIs com Spring Boot 18 - Fi" and the address bar shows "localhost:8080/api/2". The main content area displays the following JSON object:

```
{"codigo":2,"nome":"Joyce","idade":"33"}
```

#19

**Alterando dados com save equivalente ao update do sql**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
              - controle
                - Controle.java
      - resources
      - test
      - target
      - .gitignore
      - HELP.md
- Code Editor:** The current file is Controle.java, which contains the following code:

```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
31     @GetMapping("/api/{codigo}")
32     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
33         return acao.findByCodigo(codigo);
34     }
35
36     public Pessoa editar(){}
37
38
39
40     @GetMapping("/")
41     public String mensagem(){
42         return "Hello World";
43     }
44
```

A cursor is positioned at the start of the `editar()` method definition.

## Criamos o metodo

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure for "APIS-SPRING-BOOT".
- Code Editor:** The file "Controle.java" is open, showing Java code for a Spring Boot API. A tooltip is displayed over the line containing the annotation `@PutMapping`.
- Tooltip Content:**
  - Annotation for mapping HTTP `PUT` requests onto specific handler methods.
  - Specifically, `@PutMapping` is a composed annotation that acts as a shortcut for `@RequestMapping(method = RequestMethod.PUT)`.
  - Since:
    - 4.3
  - Author:
    - Sam Brannen
  - See Also:
    - GetMapping
    - PostMapping
    - DeleteMapping

```
• Controle.java - APIS-SPRING-BOOT - Visual Studio Code
  • Controle.java 1 •
  org.springframework.web.bind.annotation.PutMapping
Annotation for mapping HTTP PUT requests onto specific handler methods.

Specifically, @PutMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.PUT).

• Since:
  ◦ 4.3
• Author:
  ◦ Sam Brannen
• See Also:
  ◦ GetMapping
  ◦ PostMapping
  ◦ DeleteMapping
```

```
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

```
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

```
    @Autowired
    private Repositorio acao;

    @PostMapping("/api")
    public Pessoa cadastrar(@RequestBody Pessoa obj){
        return acao.save(obj);
    }

    @GetMapping("/api")
    public List<Pessoa> selecionar(){
        return acao.findAll();
    }

    @GetMapping("/api/{codigo}")
    public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
        return acao.findByCodigo(codigo);
    }

    @PutMapping("/")
    public void PutMapping(@PutMapping - org.springframework.web.bind.annotation.PutMapping - jakarta.persistence.StoredProcedureParameter - jakarta.persistence.StoredProcedureParameter){}

    @GetMapping("/")
    public String mensagem(){}
```

Adicionamos a anotation `@PutMapping`

The screenshot shows the Visual Studio Code interface with the following details:

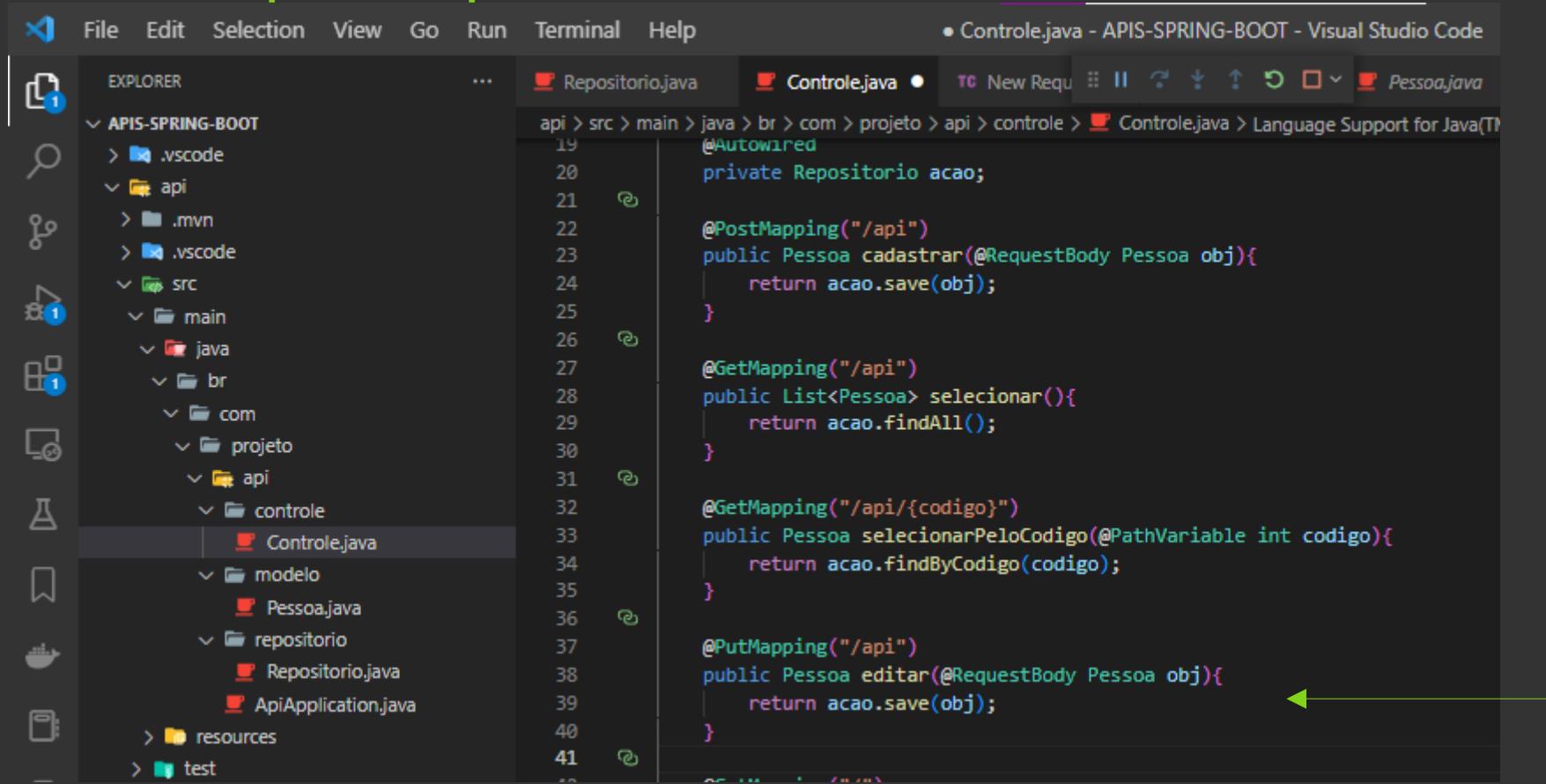
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Terminal:** Shows the command "Controle.java - APIS-SPRING-BOOT - Visual Studio Code".
- Code Editor:** Displays Java code for "Controle.java".

```
19     @Autowired
20     private Repositorio acao;
21
22     @PostMapping("/api")
23     public Pessoa cadastrar(@RequestBody Pessoa obj){
24         return acao.save(obj);
25     }
26
27     @GetMapping("/api")
28     public List<Pessoa> selecionar(){
29         return acao.findAll();
30     }
31
32     @GetMapping("/api/{codigo}")
33     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
34         return acao.findByCodigo(codigo);
35     }
36
37     @PutMapping("/api")
38     public Pessoa editar(){ ←
39
40     }
41
42     @GetMapping("/")
43     public String mensagem(){ ←
44
45 }
```

A green arrow points from the text "Sempre que formos alterar usando o put temos de passa um objeto completo" to the line "public Pessoa editar()".

**Sempre que formos alterar usando o put temos de passa um objeto completo**

# Com put temos de passar o objeto completo não podemos passar só um atributo



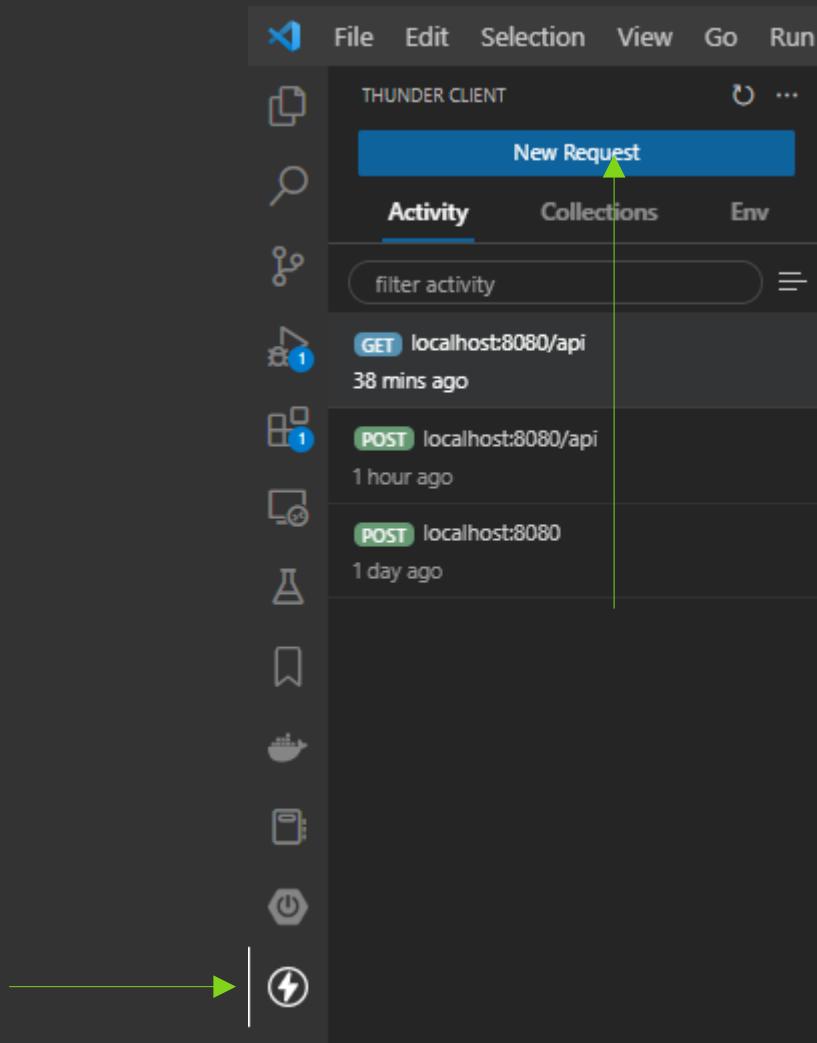
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
      - resources
      - test
- Editor:** Displays the Java code for the Controle.java file.

```
19  @Autowired
20  private Repositorio acao;
21
22  @PostMapping("/api")
23  public Pessoa cadastrar(@RequestBody Pessoa obj){
24      return acao.save(obj);
25  }
26
27  @GetMapping("/api")
28  public List<Pessoa> selecionar(){
29      return acao.findAll();
30  }
31
32  @GetMapping("/api/{codigo}")
33  public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
34      return acao.findByCodigo(codigo);
35  }
36
37  @PutMapping("/api")
38  public Pessoa editar(@RequestBody Pessoa obj){
39      return acao.save(obj);
40  }
41
```

A green arrow points from the bottom right towards the editor area.

Salve e va no thunder client



[←](#) [→](#) [C](#)[localhost:8080/api](#)

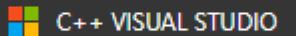
c++



C++



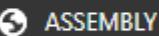
br-C.org



C++ VISUAL STUDIO



Domestika



ASSEMBLY



ss64

```
[{"codigo":1,"nome":"Cristiano","idade":"40"}, {"codigo":2,"nome":"Joyce","idade":"33"}]
```

Colamos e mudamos os dados de cristiano  
para akilles

File Edit Selection View Go Run Terminal Help

New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/api  
38 mins ago

POST localhost:8080/api  
1 hour ago

POST localhost:8080  
1 day ago

PUT localhost:8080/api

New Req New Request Pessoas

Status: Size: Time:

Response Headers Cookies Results Docs

PUT localhost:8080/api

Send

Query Headers 2 Auth Body Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Format

Json Content

```
1 {  
2   "codigo":1,  
3   "nome":"Akilles",  
4   "idade":11  
5  
6 }
```

The screenshot shows a Visual Studio Code interface with a 'THUNDER CLIENT' extension open. On the left, there's a sidebar with activity logs for various API requests. The main area is a 'New Request' window for a 'PUT' operation to 'localhost:8080/api'. The 'Body' tab is selected, showing a JSON content pane with the following code:

```
1 {  
2   "codigo":1,  
3   "nome":"Akilles",  
4   "idade":11  
5  
6 }
```

A green arrow points from the JSON content in the VS Code interface up towards the browser window above, indicating the data being sent.

The screenshot shows the Thunder Client extension in Visual Studio Code. The interface includes a navigation bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the bar, there are tabs for Repository.java, Controle.java, New Request, Pessoa.java, and pessoas. The main area is titled "New Request - APIS-SPRING-BOOT - Visual Studio Code". On the left, a sidebar shows activity logs with entries for PUT localhost:8080/api, GET localhost:8080/api, POST localhost:8080/api, and POST localhost:8080. The central workspace shows a "New Request" configuration for a PUT request to localhost:8080/api. The "Body" tab is selected, containing a JSON content block:

```
1 {  
2   "codigo": 1,  
3   "nome": "Akilles",  
4   "idade": 11  
5 }
```

The response panel on the right shows a successful 200 OK status with a size of 42 Bytes and a time of 659 ms. The response body is identical to the JSON sent in the request.

Como cristiano tem um code que já existe ele sofre uma atualização por akilles é substituido

**Temos os dados alterados se ele não existisse seria criado**



A screenshot of a web browser window. The address bar shows the URL `localhost:8080/api`. The main content area displays a JSON array:

```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":2,"nome":"Joyce","idade":"33"}]
```

#20

**Removendo os dados com delete equivalente ao sql delete**

File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code

REPOSITORY

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md

REPOSITORY

Repository.java

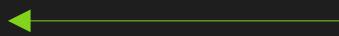
Controle.java

New Request

New Request

Pessoa.java

```
19 @Autowired
20     private Repositorio acao;
21
22     @PostMapping("/api")
23     public Pessoa cadastrar(@RequestBody Pessoa obj){
24         return acao.save(obj);
25     }
26
27     @GetMapping("/api")
28     public List<Pessoa> selecionar(){
29         return acao.findAll();
30     }
31
32     @GetMapping("/api/{codigo}")
33     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
34         return acao.findByCodigo(codigo);
35     }
36
37     @PutMapping("/api")
38     public Pessoa editar(@RequestBody Pessoa obj){
39         return acao.save(obj);
40     }
41
42     public void remover(){
43     }
44 }
```

Criamos um metodo remover(){}

The screenshot shows a Java code editor with the following code snippet:

```
    }
    @Delete
    public void DeleteExchange() {
    }
}

@GetMa
public void DefaultValue() {
}

```

A tooltip is displayed for the `@Delete` annotation, providing the following information:

org.springframework.web.service.annotation.DeleteExchange  
Shortcut for @HttpExchange for HTTP DELETE requests.

- Since:
  - 6.0
- Author:
  - Rossen Stoyanchev

Below the code editor, there is a status bar with the text "Controle.java api\src\main\java\br\com\projeto\api\controle" and a warning message: "Delete cannot be resolved to a type Java(16777218) [Ln 42, Col 5]".

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer:** Shows the project structure under APIS-SPRING-BOOT:
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
      - modelo
        - Pessoa.java
      - repository
        - Repositorio.java
        - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
  - Code Editor:** The file Controle.java is open, showing the following code:

```
private Repositorio acao;

@PostMapping("/api")
public Pessoa cadastrar(@RequestBody Pessoa obj){
    return acao.save(obj);
}

@GetMapping("/api")
public List<Pessoa> selecionar(){
    return acao.findAll();
}

@GetMapping("/api/{codigo}")
public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
    return acao.findByCodigo(codigo);
}

@PutMapping("/api")
public Pessoa editar(@RequestBody Pessoa obj){
    return acao.save(obj);
}

@DeleteMapping("/api")
public void remover(){}
```
  - Status Bar:** Shows tabs for Repository.java, Controle.java (active), New Request, and Pessoa.java.

Adicionamos a anotation e criamos a rota

**Utilizamos esse parametro pois temos de saber quem será removido**

```
        return dados.save(obj);
    }

    @DeleteMapping("/api/{codigo}")
    public void remover(){
    }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT:
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
      - modelo
        - Pessoa.java
      - repositorio
        - Repositorio.java
        - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
- Terminal View:** Shows the current file path: api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java.
- Code Editor View:** Displays the Controle.java code with syntax highlighting and line numbers. A red arrow points to the closing brace of the remove method.

```
22
23     @PostMapping("/api")
24     public Pessoa cadastrar(@RequestBody Pessoa obj){
25         return acao.save(obj);
26     }
27
28     @GetMapping("/api")
29     public List<Pessoa> selecionar(){
30         return acao.findAll();
31     }
32
33     @GetMapping("/api/{codigo}")
34     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
35         return acao.findByCodigo(codigo);
36     }
37
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){[red arrow]
45
46     }
```

Para remover uma pessoa temos que ter todos os dados nome  
idade e codigo

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer:** Shows the project structure for "APIS-SPRING-BOOT". The "src" folder contains "main", which has "java", "br", "com", "projeto", "api", and "controle". "controle" contains "Controle.java". Other files in "main/java" include "Repositorio.java", "Pessoa.java", and "ApiApplication.java".
- Code Editor:** The "Controle.java" file is open. The code defines a controller for a "Pessoa" entity using Spring Boot annotations. It includes methods for creating, reading, updating, and deleting people from a database.

```
22
23     @PostMapping("/api")
24     public Pessoa cadastrar(@RequestBody Pessoa obj){
25         return acao.save(obj);
26     }
27
28     @GetMapping("/api")
29     public List<Pessoa> selecionar(){
30         return acao.findAll();
31     }
32
33     @GetMapping("/api/{codigo}")
34     public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
35         return acao.findByCodigo(codigo);
36     }
37
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46     }
47
```

Criamos o objeto

Esse metodo  
retorna um  
objeto  
pessoa com  
os dados

```
40     |         return acao.save(obj);
41     |     }
42     @
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46
47     @
48     |     acao.de
49     |
50     @GetMapping("/")
51     public String m
```

PROBLEMS 3 OUTPUT TERMINAL

✓ Controle.java api\src\main\java\bri Syntax error, insert "VariableDeletion" for type information

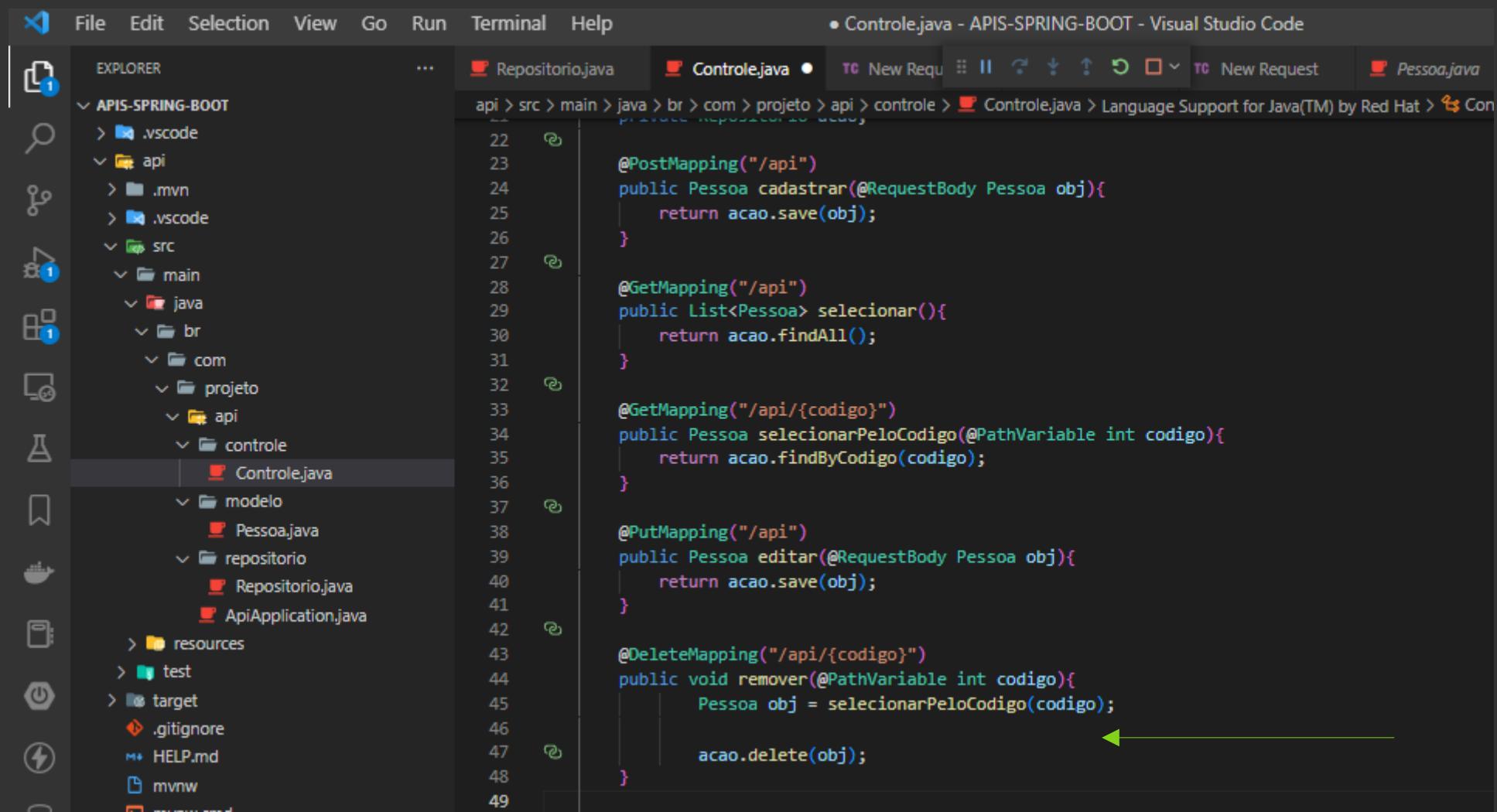
Deletes a given entity.

- **Parameters:**

- **entity** must not be null.

- **Throws:**

- **IllegalArgumentException** - in case the given entity is null.
- **OptimisticLockingFailureException** - when the entity uses optimistic locking and has a version attribute with a different value from that found in the persistence store. Also thrown if the entity is assumed to be present but does not exist in the database.



**Salve e va no thunder client e faça uma new request**

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

**PUT** localhost:8080/api  
20 mins ago

**GET** localhost:8080/api  
1 hour ago

**POST** localhost:8080/api  
1 hour ago

**POST** localhost:8080  
1 day ago

Repository.java Controle.java New Requ New Request New Request Pessoajava pessoas

GET https://www.thunderclient.com/welcome Send

Status: Size: Time:

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

parameter value

Send Request Ctrl + Enter

Import Curl Ctrl + U

Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, \*\*/node\_modules/\*\*)

The screenshot shows the Thunder Client extension interface within Visual Studio Code. The top bar displays the title "New Request - APIS-SPRING-BOOT - Visual Studio Code". The main area is divided into two sections: "Activity" on the left and "New Request" on the right.

**Activity:**

- PUT localhost:8080/api (20 mins ago)
- GET localhost:8080/api (1 hour ago)
- POST localhost:8080/api (1 hour ago)
- POST localhost:8080 (1 day ago)

**New Request:**

**Request Type:** DELETE

**URL:** localhost:8080/api/2

**Buttons:** Send, Headers, Auth, Body, Tests, Pre Run, New

**Query Parameters:**

| parameter | value |
|-----------|-------|
|           |       |

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

DEL localhost:8080/api/2 just now

PUT localhost:8080/api 20 mins ago

GET localhost:8080/api 1 hour ago

POST localhost:8080/api 1 hour ago

POST localhost:8080 1 day ago

Repository.java Controle.java New Request New Request Pessoajava pessoas

DELETE localhost:8080/api/2 Send

Status: 200 OK Size: 0 Bytes Time: 170 ms

Response Headers Cookies Results Docs

1

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

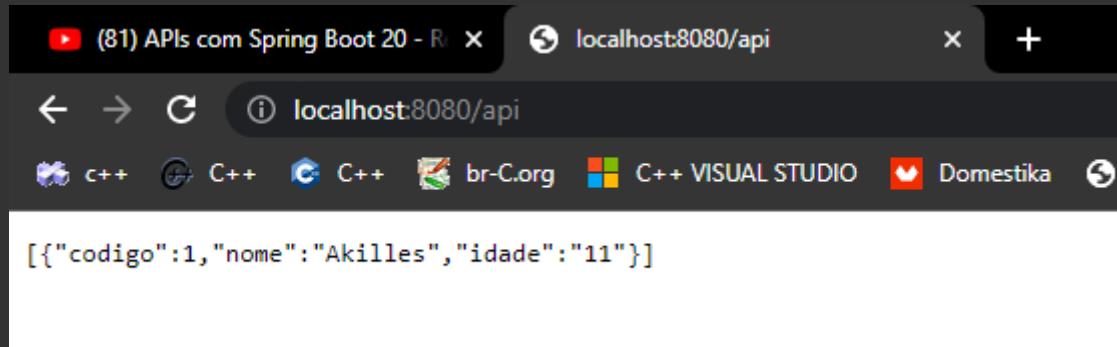
parameter value

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, \*\*/node\_modules/\*\*)

**Atualize o navegador e veja que um registro foi  
deletado**



File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Repository.java Controle.java New Request New Request Pessoajava pessoas

New Request

Activity Collections Env filter activity

DELETE localhost:8080/api/2 Send

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters parameter value

Status: 500 Internal Server Error Size: 9.81 KB Time: 181 ms

Response Headers 4 Cookies Results Docs

```
1 {
2   "timestamp": "2023-01-23T20:00:48.044+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "trace": "org.springframework.dao.InvalidDataAccessApiUsageException:  
Entity must not be null\r\n\tat org.springframework.orm.jpa.  
.EntityManagerFactoryUtils.convertJpaAccessExceptionIfPossible  
(EntityManagerFactoryUtils.java:371)\r\n\tat org.springframework.orm.  
.jpa.vendor.HibernateJpaDialect.translateExceptionIfPossible  
(HibernateJpaDialect.java:235)\r\n\tat org.springframework.orm.jpa.  
.AbstractEntityManagerFactoryBean.translateExceptionIfPossible  
(AbstractEntityManagerFactoryBean.java:550)\r\n\tat org  
.springframework.dao.support.ChainedPersistenceExceptionTranslator  
.translateExceptionIfPossible(ChainedPersistenceExceptionTranslator  
.java:61)\r\n\tat org.springframework.dao.support.DataAccessUtils  
.translateIfNecessary(DataAccessUtils.java:242)\r\n\tat org  
.springframework.dao.support.  
.PersistenceExceptionTranslationInterceptor.invoke  
(PersistenceExceptionTranslationInterceptor.java:152)\r\n\tat org  
.springframework.aop.framework.ReflectiveMethodInvocation.proceed  
(ReflectiveMethodInvocation.java:184)\r\n\tat org.springframework.aop.  
.data.jpa.repository.support  
.CrudMethodMetadataPostProcessor$CrudMethodMetadataPopulatingMethodIn  
terceptor.invoke(CrudMethodMetadataPostProcessor.java:163)\r\n\tat org  
.springframework.aop.framework.ReflectiveMethodInvocation.proceed  
(ReflectiveMethodInvocation.java:184)\r\n\tat org.springframework.aop.  
.interceptor.ExposeInvocationInterceptor.invoke
```

Caso os dados não existam ele da um erro mais resolveremos com a validação

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, \*\*/node\_modules/\*\*)

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) 127.0.0.1 api\_Spring Go Live

Chuva por parar 17:00

#21

**Colocando mais pessoas no banco de dados  
CONTAR REGISTROS COM O COMANDO COUNT()**

File Edit Selection View Go Run Terminal Help localhost:8080/pessoa - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Repository.java Controle.java localhost:8080 New Request

New Request

Activity Collections Env filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode GraphQL Binary

Json Content Format

```
1 {  
2   "codigo": 4,  
3   "nome": "Joyce",  
4   "idade": "33"  
5 }
```

Status: 200 OK Size: 40 Bytes Time: 104 ms

Response Headers 4 Cookies Results Docs

NO THUNDER  
Escreva seu objeto json e send  
A imagem mostra como  
Lembre-se de adicionar a rota certa

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
2023-01-24T12:56:14.869-03:00  WARN 10052 --- [nio-8080-exec-2] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]  
2023-01-24T13:01:40.743-03:00  WARN 10052 --- [nio-8080-exec-6] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
```

DEBUG CONSOLE

Run: ApiApplication Run: ApiApplication

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) 127.0.0.1 api\_Spring Go Live 27°C Nublado 13:05

File Edit Selection View Go Run Terminal Help localhost:8080/pessoa - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Repository.java Controle.java localhost:8080 New Request

New Request

Activity Collections Env filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content

```
1 {  
2   "nome": "Cristiano",  
3   "idade": 33  
4 }
```

Format

Status: 200 OK Size: 44 Bytes Time: 120 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "codigo": 5,  
3   "nome": "Cristiano",  
4   "idade": "33"  
5 }
```

POST localhost:8080/pessoa just now

DEL localhost:8080/api/2 20 hours ago

PUT localhost:8080/api 21 hours ago

GET localhost:8080/api 21 hours ago

POST localhost:8080/api 22 hours ago

POST localhost:8080 2 days ago

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
2023-01-24T12:56:14.869-03:00  WARN 10052 --- [nio-8080-exec-2] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]  
2023-01-24T13:01:40.743-03:00  WARN 10052 --- [nio-8080-exec-6] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
```

DEBUG CONSOLE

Run: ApiApplication Run: ApiApplication

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) 127.0.0.1 apiSpring Go Live

27°C Nublado

localhost:8080/api

Vamos adicionar mais um dessa vez start wars

```
[{"codigo":1,"nome":"Akilles","idade":11}, {"codigo":3,"nome":"Carlos","idade":65}, {"codigo":4,"nome":"Joyce","idade":33}, {"codigo":5,"nome":"Cristiano","idade":33}]
```

localhost:8080/pessoa - APIS-SPRING-BOOT - Visual Studio Code

New Request

Activity Collections Env

filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content

```
1 {  
2   "nome": "Luke",  
3   "idade": 34  
4 }
```

Status: 200 OK Size: 39 Bytes Time: 166 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "codigo": 6,  
3   "nome": "Luke",  
4   "idade": 34  
5 }
```

localhost:8080/api

localhost:8080/api

idade do luke - Pesquisa Google

```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":3,"nome":"Carlos","idade":"65"}, {"codigo":4,"nome":"Joyce","idade":"33"}, {"codigo":5,"nome":"Cristiano","idade":"33"}, {"codigo":6,"nome":"Luke","idade":"34"}]
```

localhost:8080/pessoa - APIS-SPRING-BOOT - Visual Studio Code

New Request

Activity Collections Env

filter activity

POST http://localhost:8080/api

Send

Status: 200 OK Size: 46 Bytes Time: 154 ms

Response Headers Cookies Results Docs

```
1 {  
2   "codigo": 7,  
3   "nome": "Darth Vader",  
4   "idade": "41"  
5 }
```

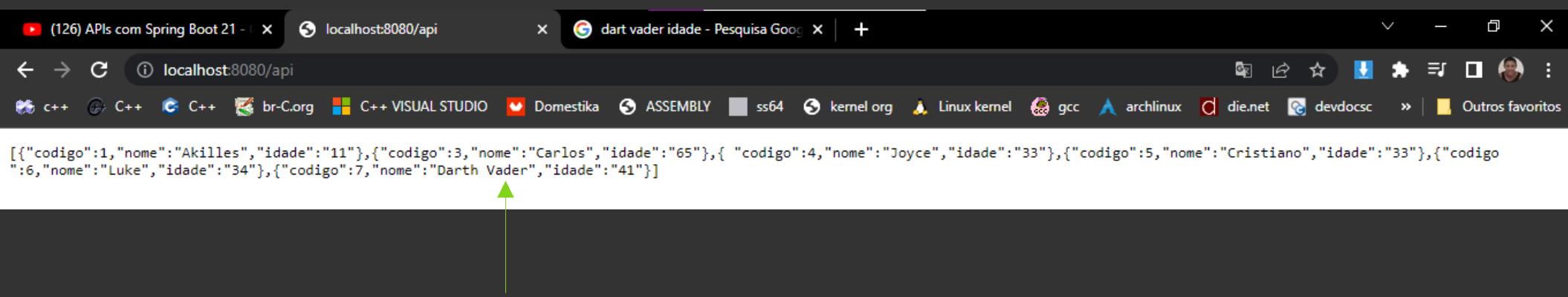
localhost:8080/pessoa just now

localhost:8080/api/2 20 hours ago

localhost:8080/api 21 hours ago

localhost:8080/api 21 hours ago

localhost:8080/api 21 hours ago

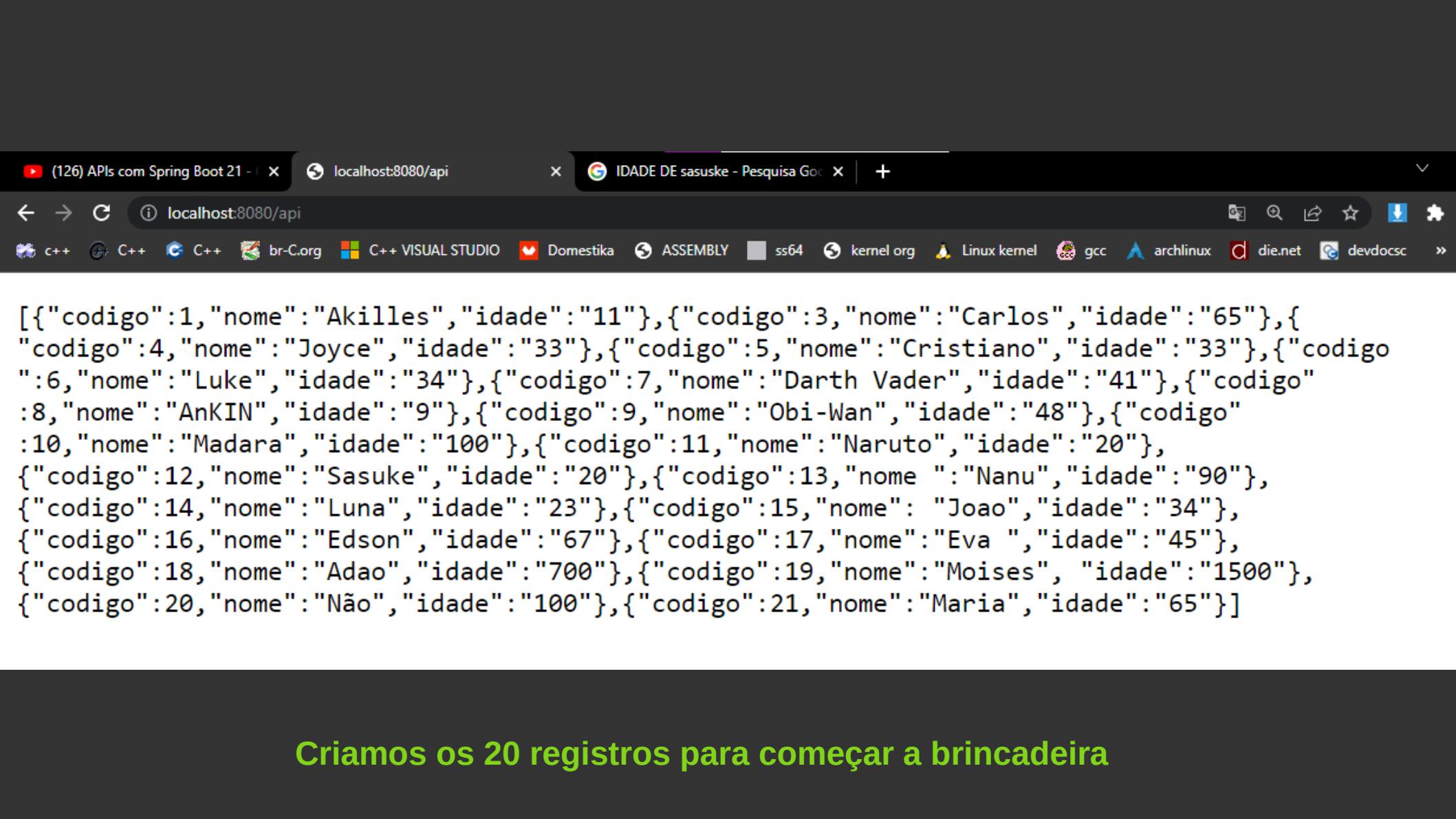


A screenshot of a browser window with a dark theme. The address bar shows 'localhost:8080/api'. The main content area displays a JSON array of objects:

```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":3,"nome":"Carlos","idade":"65"}, {"codigo":4,"nome":"Joyce","idade":"33"}, {"codigo":5,"nome":"Cristiano","idade":"33"}, {"codigo":6,"nome":"Luke","idade":"34"}, {"codigo":7,"nome":"Darth Vader","idade":"41"}]
```

A green arrow points upwards from the bottom of the slide towards the JSON data.

Seguimos no processo criamos o objeto json e enviamos e depois abrimos o navegador pra conferir fazemos isso para 20 nomes mais pode ser infinitos nomes e depois conferimos no banco de dados

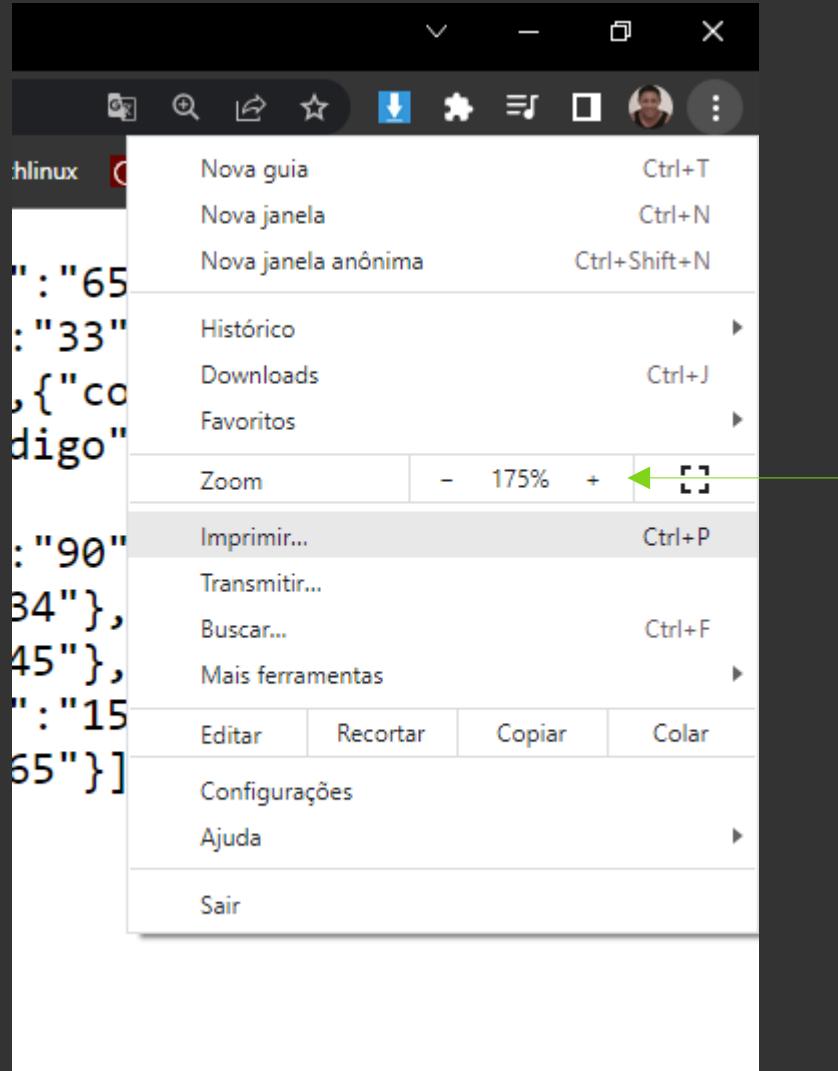


A screenshot of a browser window with a dark theme. The address bar shows "localhost:8080/api". The main content area displays a JSON array of 20 objects, each containing "codigo", "nome", and "idade" fields. The JSON is formatted as follows:

```
[{"codigo":1,"nome":"Akilles","idade":11}, {"codigo":3,"nome":"Carlos","idade":65}, {"codigo":4,"nome":"Joyce","idade":33}, {"codigo":5,"nome":"Cristiano","idade":33}, {"codigo":6,"nome":"Luke","idade":34}, {"codigo":7,"nome":"Darth Vader","idade":41}, {"codigo":8,"nome":"AnKIN","idade":9}, {"codigo":9,"nome":"Obi-Wan","idade":48}, {"codigo":10,"nome":"Madara","idade":100}, {"codigo":11,"nome":"Naruto","idade":20}, {"codigo":12,"nome":"Sasuke","idade":20}, {"codigo":13,"nome":"Nanu","idade":90}, {"codigo":14,"nome":"Luna","idade":23}, {"codigo":15,"nome":"Joao","idade":34}, {"codigo":16,"nome":"Edson","idade":67}, {"codigo":17,"nome":"Eva","idade":45}, {"codigo":18,"nome":"Adao","idade":700}, {"codigo":19,"nome":"Moises","idade":1500}, {"codigo":20,"nome":"Não","idade":100}, {"codigo":21,"nome":"Maria","idade":65}]
```

**Criamos os 20 registros para começar a brincadeira**

Pra dar zoom no navegador



File Edit Selection View Go Run Terminal Help pessoas - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25 Repository.java Controle.java localhost:8080 New Request

SELECT \* FROM pessoas LIMIT 100;

Input to filter result Free 1

Free 1

columns

|    | * codigo | int | idade       | nome |
|----|----------|-----|-------------|------|
| 1  | 1        | 11  | Akilles     |      |
| 2  | 3        | 65  | Carlos      |      |
| 3  | 4        | 33  | Joyce       |      |
| 4  | 5        | 33  | Cristiano   |      |
| 5  | 6        | 34  | Luke        |      |
| 6  | 7        | 41  | Darth Vader |      |
| 7  | 8        | 9   | AnKIN       |      |
| 8  | 9        | 48  | Obi-Wan     |      |
| 9  | 10       | 100 | Madara      |      |
| 10 | 11       | 20  | Naruto      |      |
| 11 | 12       | 20  | Sasuke      |      |
| 12 | 13       | 90  | Nanu        |      |
| 13 | 14       | 23  | Luna        |      |
| 14 | 15       | 24  | Iago        |      |

PROBLEMS OUTPUT TERMINAL

TERMINAL

```
2023-01-24T12:56:14.869-03:00  WARN 10052 --- [nio-8080-exec-2] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
2023-01-24T13:01:40.743-03:00  WARN 10052 --- [nio-8080-exec-6] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
```

DEBUG CONSOLE

Run: ApiApplication  
Run: ApiApplication

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) 127.0.0.1 apiSpring

Go Live

28°C Nublado

13:29

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Left Sidebar (DATABASE):** Shows a connection to "127.0.0.1@3306 8.0.25". Under "api\_spring 16k", there is a "tables (1)" section with "pessoas". The "pessoas" table has columns: "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other sections shown include "index", "views", "procedures", "functions", "dbagenda", "sakila", and "world".
- Right Pane:** The "Controle.java" file is open. The code is as follows:

```
30     return acao.findAll();
31 }
32
33 @GetMapping("/api/{codigo}")
34 public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
35     return acao.findByCodigo(codigo);
36 }
37
38 @PutMapping("/api")
39 public Pessoa editar(@RequestBody Pessoa obj){
40     return acao.save(obj);
41 }
42
43 @DeleteMapping("/api/{codigo}")
44 public void remover(@PathVariable int codigo){
45     Pessoa obj = selecionarPeloCodigo(codigo);
46     acao.delete(obj);
47 }
48
49 public Long
50
51 @GetMapping("/")
52 public String mensagem(){
53     return "Hello World";
54 }
```

A cursor is positioned at the start of the "public Long" line on line 49. A lightbulb icon is visible on the far left of the line, indicating a potential code completion or suggestion.

Começamos criando um metodo publico que retorna um Long

**Porque Long?**

**Porque é o tipo de dado que o metodo count vai retornar**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active Editor:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Terminal:** Localhos.
- Code Content:** A Java Spring Boot controller named Controle.java. The code includes methods for selecting, editing, removing, and counting entities from a database. A green arrow points from the text "Nome do metodo contador" to the method definition at line 49.

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java
```

```
30     return acao.findAll();  
31 }  
32  
33 @GetMapping("/api/{codigo}")  
34 public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
35     return acao.findByCodigo(codigo);  
36 }  
37  
38 @PutMapping("/api")  
39 public Pessoa editar(@RequestBody Pessoa obj){  
40     return acao.save(obj);  
41 }  
42  
43 @DeleteMapping("/api/{codigo}")  
44 public void remover(@PathVariable int codigo){  
45     Pessoa obj = selecionarPeloCodigo(codigo);  
46     acao.delete(obj);  
47 }  
48  
49 public Long contador(){  
50 }  
51  
52  
53 @GetMapping("/")  
54 public String mensagem(){  
55     return "Hello World";  
56 }  
57
```

- Database Explorer:** Shows a connection to 127.0.0.1@3306 (8.0.25) and a schema named api\_spring (16k). It lists tables (1), specifically pessoas, which has columns: codigo (int), idade (varchar(255)), and nome (varchar(255)). Other objects like index, views, procedures, and functions are also listed.

Nome do metodo  
contador



File Edit Selection View Go Run Terminal Help

• Controle.java - APIS-SPRING-BOOT - Visual Studio Code



DATABASE

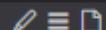


127.0.0.1@3306 8.0.25

api\_spring 16k

tables (1)

pessoas



columns

codigo int

idade varchar(255)

nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world

30                   return acao.findAll();  
31       }  
32  
33       @GetMapping("/api/{codigo}")  
34       public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
35            return acao.findByCodigo(codigo);  
36       }  
37  
38       @PutMapping("/api")  
39       public Pessoa editar(@RequestBody Pessoa obj){  
40            return acao.save(obj);  
41       }  
42  
43       @DeleteMapping("/api/{codigo}")  
44       public void remover(@PathVariable int codigo){  
45            Pessoa obj = selecionarPeloCodigo(codigo);  
46            acao.delete(obj);  
47       }  
48  
49       public Long contador(){  
50            return acao.count();  
51       }  
52  
53       @GetMapping("/")  
54       public String mensagem(){  
55            return "Hello World";  
56       }

The screenshot shows a Visual Studio Code interface with a dark theme. On the left, there's a sidebar with various icons for file operations like opening, saving, and running. The main area has a top navigation bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the navigation is a title bar for 'Controle.java - APIS-SPRING-BOOT - Visual Studio Code'. The central part of the screen contains two panes: a database sidebar on the left and a code editor on the right.

**Database Sidebar:**

- CONNECTED TO DATABASE: 127.0.0.1@3306 8.0.25
- SCHEMA: api\_spring (16k)
- TABLES (1):
  - pessoas
    - COLUMNS:
      - codigo int
      - idade varchar(255)
      - nome varchar(255)
- INDEXES (1): index
- VIEWS (1): views
- PROCEDURES (1): procedures
- FUNCTIONS (1): functions
- DBAGENDA (1): dbagenda
- SAKILA (1): sakila
- WORLD (1): world

**Code Editor:**

```
  • Controle.java - APIS-SPRING-BOOT - Visual Studio Code
  Repository.java  Controle.java  localhost:3001  New Request
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat
    return acao.findAll();
}
@GetMapping("/api/{codigo}")
public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
    return acao.findByCodigo(codigo);
}

@PostMapping("/api")
public Pessoa editar(@RequestBody Pessoa obj){
    return acao.save(obj);
}

@DeleteMapping("/api/{codigo}")
public void remover(@PathVariable int codigo){
    Pessoa obj = selecionarPeloCodigo(codigo);
    acao.delete(obj);
}

@GetMapping("/api/contador")
public Long contador(){
    return acao.count(); ←
}

@GetMapping("/")
public String mensagem(){
    return "Hello World";
}
```

**Salve va no navegador  
E coloque a rota *api/contador*  
Ela deve mostrar quantos registros temos**

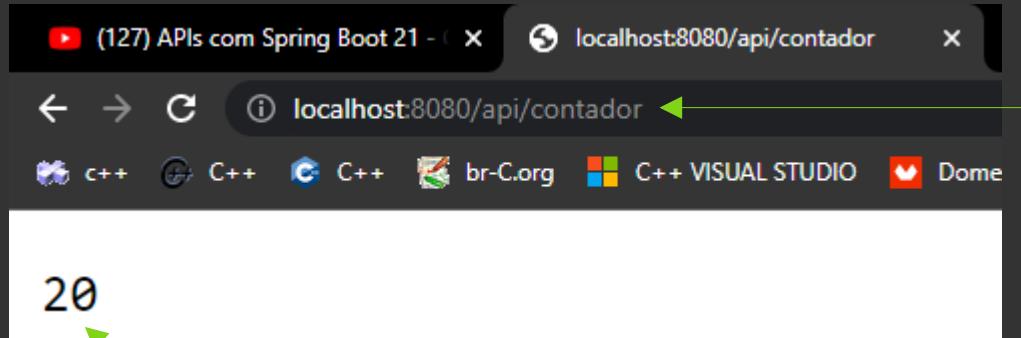
The screenshot shows a MySQL Workbench interface with the following details:

- File Edit Selection View Go Run Terminal Help**
- pessoas - APIS-SPRING-BOOT**
- DATABASE** node expanded, showing:
  - 127.0.0.1@3306 8.0.25
  - api\_spring 16k
  - tables (1)
    - pessoas
  - views
  - procedures
  - functions
  - dbagenda
  - sakila
  - world
- Terminal** tab open with the command:

```
SELECT * FROM pessoas LIMIT 100;
```
- Results Grid** showing the output of the query:

|    | * codigo | idade        | nome         |
|----|----------|--------------|--------------|
|    | int      | varchar(255) | varchar(255) |
| 1  | 8        | 9            | Obi-Wan      |
| 2  | 9        | 10           | Madara       |
| 3  | 10       | 11           | Naruto       |
| 4  | 11       | 12           | Sasuke       |
| 5  | 12       | 13           | Nanu         |
| 6  | 13       | 14           | Luna         |
| 7  | 14       | 15           | Joao         |
| 8  | 15       | 16           | Edson        |
| 9  | 16       | 17           | Eva          |
| 10 | 17       | 18           | Adao         |
| 11 | 18       | 19           | Moises       |
| 12 | 19       | 20           | Noe          |
| 13 | 20       | 21           | Maria        |

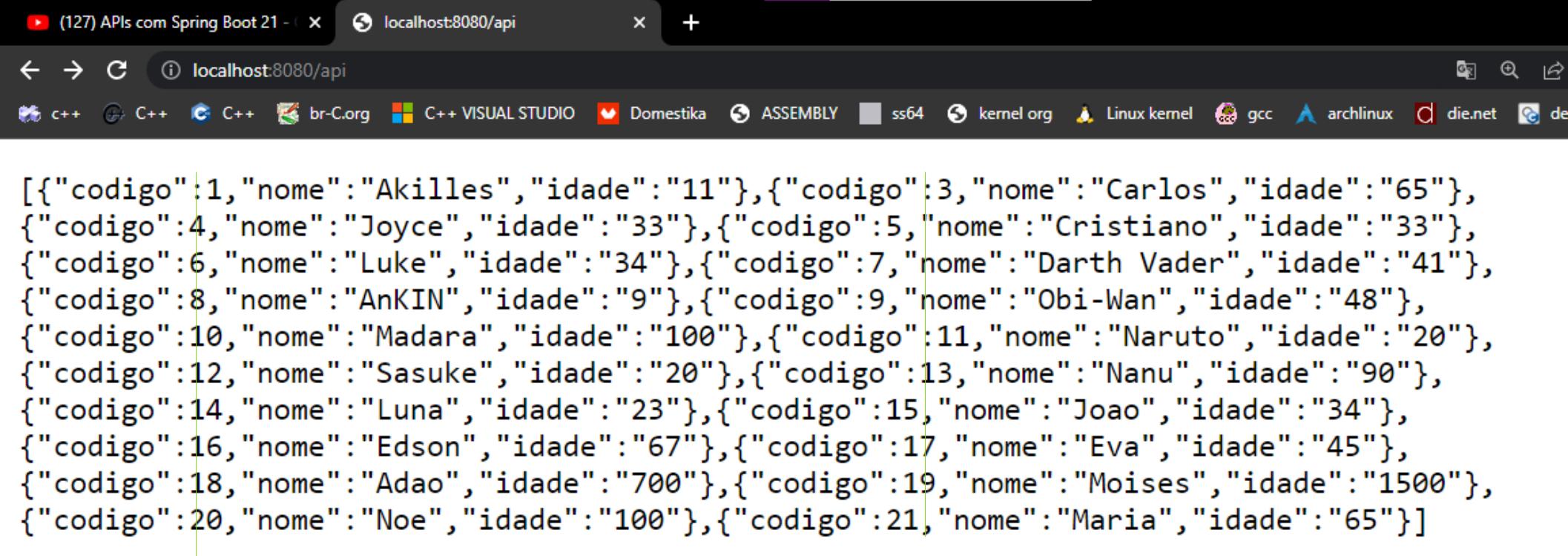
**Nosso banco de dados mostra 21 registros**



Pois deletamos um

#22

Ordenar registros  
Ordenar informações de uma tabela



A screenshot of a web browser window titled "localhost:8080/api". The address bar also shows "localhost:8080/api". The page content displays a JSON array of 21 objects, each containing "codigo", "nome", and "idade" fields. The objects are ordered by "codigo" from 1 to 20, and then one additional object with "codigo": 21.

```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":3,"nome":"Carlos","idade":"65"}, {"codigo":4,"nome":"Joyce","idade":"33"}, {"codigo":5,"nome":"Cristiano","idade":"33"}, {"codigo":6,"nome":"Luke","idade":"34"}, {"codigo":7,"nome":"Darth Vader","idade":"41"}, {"codigo":8,"nome":"AnKIN","idade":"9"}, {"codigo":9,"nome":"Obi-Wan","idade":"48"}, {"codigo":10,"nome":"Madara","idade":"100"}, {"codigo":11,"nome":"Naruto","idade":"20"}, {"codigo":12,"nome":"Sasuke","idade":"20"}, {"codigo":13,"nome":"Nanu","idade":"90"}, {"codigo":14,"nome":"Luna","idade":"23"}, {"codigo":15,"nome":"Joao","idade":"34"}, {"codigo":16,"nome":"Edson","idade":"67"}, {"codigo":17,"nome":"Eva","idade":"45"}, {"codigo":18,"nome":"Adao","idade":"700"}, {"codigo":19,"nome":"Moises","idade":"1500"}, {"codigo":20,"nome":"Noe","idade":"100"}, {"codigo":21,"nome":"Maria","idade":"65"}]
```

**Os registros estão ordenados porém e se quisermos ordenar por nome**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Left Sidebar:** Shows a database connection to "127.0.0.1@3306 8.0.25" and tables like "pessoas", "views", "procedures", "functions", "dbagenda", "sakila", and "world".
- Central Editor:** A Java file named "Repositorio.java" is open. The code defines a public interface "Repositorio" extending "JpaRepository<Pessoa, Integer>". It includes methods "findAll()" and "findByCodigo(int codigo)".
- Completion Panel:** A dropdown menu is open at the bottom of the editor, showing suggestions for the word "List". The top suggestion is "List - java.util". Other suggestions include "ListAppender - ch.qos.logback.core.read", "ListAttribute - jakarta.persistence.metamodel", "ListBinder - org.hibernate.cfg.annotations", "ListCellRenderer - javax.swing", "ListCrudRepository - org.springframework.data.rep...", "ListDataEvent - javax.swing.event", "ListDataHandler - javax.swing.plaf.basic.BasicCom...", "ListDataHandler - javax.swing.plaf.basic.BasicCom...", "ListDataHandler - javax.swing.plaf.basic.BasicList...", "ListDataListener - javax.swing.event", and "ListELResolver - jakarta.el".
- Right Sidebar:** A tooltip for "java.util.List" is displayed, providing documentation:
  - An ordered collection (also known as a *sequence*). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.
  - Unlike sets, lists typically allow duplicate elements. More formally, lists typically allow pairs of elements *e*<sub>1</sub> and *e*<sub>2</sub> such that *e*<sub>1</sub>.*equals*(*e*<sub>2</sub>) , and they typically allow multiple null elements if they allow null elements at all. It is not inconceivable that someone might wish to implement a list that prohibits duplicates, by throwing runtime exceptions when the user attempts to insert them, but we expect this usage to be rare.
  - The *List* interface places additional stipulations, beyond those specified in the *Collection* interface, on the contracts of the *iterator*, *add* , *remove* , *equals* , and *hashCode* methods. Declarations for other inherited methods are also included here for convenience.
  - The *List* interface provides four methods for positional (indexed) access to list elements. Lists (like Java arrays) are zero based. Note that these operations may execute in time proportional to the index value for some implementations (the *LinkedList* class, for example). Thus, iterating over the elements in a list is typically preferable to indexing through it if the caller does not know the implementation.
  - The *List* interface provides a special iterator, called a *ListIterator* , that allows element
- Bottom Status Bar:** Shows "PROBLEMS 3", "OUTPUT", and "TERMINAL". The terminal output shows: "2023-01-24T13:52:04.584-03:00 INFO 10052 --- [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[localhost] : Initializing Spring DispatcherServlet 'dispatcherServlet'".
- Bottom Right:** Buttons for "Run: ApiApplication" and "Run: ApiApplication".

File Edit Selection View Go Run Terminal Help

• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25  
api api\_spring 16k  
tables (1)  
pessoas  
views  
procedures  
functions  
dbagenda  
sakila  
world

Repositorio.java 3 ● Controle.java localhos New Request

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > * Repository.java
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
→ Controle | ← 3 beans
14 public interface Repositorio extends JpaRepository<Pessoa, Integer> {
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findAll();
21
22 }
```

Criamos o metodo e ele  
retorna uma lista de  
pessoas

File Edit Selection View Go Run Terminal Help

• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 1 127.0.0.1@3306 8.0.25  
api api\_spring 16k  
tables (1)  
pessoas  
views  
procedures  
functions  
dbagenda  
sakila  
world

Repositorio.java 1 ● Controlejava localhos New Request

api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > Repositorio.java

```
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 // @Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrder
21 }
22
23 }
```

findByOrderList : List<br.com.projeto.api.modelo...  
findByOrderPessoas : List<br.com.projeto.api.mode...

Pra ele encontrar usamos o metodo `findBy` pra ele Ordenar usamos `OrderBy`  
E por ultimo especificamos uma caracteristica da nossa entidade que pode ser codigo nome  
idade nesse caso queremos o nome

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code
- Left Sidebar:** Shows a database connection to "127.0.0.1@3306 8.0.25" and a schema named "api". Inside "api", there are tables (1), views, procedures, functions, dbagenda, sakila, and world. The "pessoas" table is currently selected.
- Terminal:** localhost:3306
- Code Editor:** The file "Repositorio.java" is open, containing the following code:

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNome(); ←
21
22 }
```
- Status Bar:** Shows Language Support for Java(TM) by Red Hat, a progress bar, and a "New Request" button.

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Database Explorer:** Shows a connection to "127.0.0.1@3306 8.0.25" with a schema "api" containing tables: pessoas, views, procedures, functions, dbagenda, sakila, and world.
- Terminal:** Localhos
- Code Editor:** File: Controle.java 1, Path: api > src > main > java > br > com > projeto > api > controle > Controle.java. The code implements methods for Pessoa service: findAll(), selecionarPeloCodigo(), editar(), remover(), and contador().
- Completion Panel:** A dropdown menu is open at the bottom of the editor, showing suggestions for the word "List". The top suggestion is "List - java.util". Other suggestions include ListAppender, ListAttribute, ListBinder, ListCellRenderer, ListCrudRepository, ListDataEvent, ListDataHandler, and ListDataHandler.
- Right Panel:** A tooltip for "java.util.List" is displayed, providing documentation: "An ordered collection (also known as a sequence). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list. Unlike sets, lists typically allow duplicate elements. More formally, lists typically allow pairs of elements e<sub>1</sub> and e<sub>2</sub> such that e<sub>1</sub>.equals(e<sub>2</sub>) , and they typically allow multiple null elements if they allow null elements at all. It is not inconceivable that someone might wish to implement a list that prohibits duplicates, by throwing runtime exceptions when the user attempts to insert them, but we expect this usage to be rare." It also notes that the List interface places additional stipulations beyond those in Collection, including iterator, add, remove, equals, and hashCode methods.

**Agora vamos adicionar o metodo que criamos na Controle.java  
Onde criamos as rotas**

File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25  
api api\_spring 16k  
tables (1)  
pessoas  
views  
procedures  
functions  
dbagenda  
sakila  
world

Repository.java Controle.java 2 localhos New Request

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > ordenarNomes()
```

```
30     return acao.findAll();  
31 }  
32  
33 @GetMapping("/api/{codigo}")  
34 public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
35     return acao.findByCodigo(codigo);  
36 }  
37  
38 @PutMapping("/api")  
39 public Pessoa editar(@RequestBody Pessoa obj){  
40     return acao.save(obj);  
41 }  
42  
43 @DeleteMapping("/api/{codigo}")  
44 public void remover(@PathVariable int codigo){  
45     Pessoa obj = selecionarPeloCodigo(codigo);  
46     acao.delete(obj);  
47 }  
48  
49 @GetMapping("/api/contador")  
50 public Long contador(){  
51     return acao.count();  
52 }  
53  
54 public List<Pessoa> ordenarNomes(){  
55     return acao.  
56 }  
57  
58 @GetMapping("/")  
59 public String me  
60
```

PROBLEMS 2 OUTPUT TERMINAL

TERMINAL

```
2023-01-24T14:04:27.305-03:00 I tWebServer : Tomcat started on  
2023-01-24T14:04:27.319-03:00 I tion : Started ApiApplica  
2023-01-24T14:04:27.350-03:00 I ngListener : Condition evaluation unchanged
```

Repositorio.findByOrderByName() : List<Pessoa>

findByOrderByName() : List<Pessoa>

Run: ApiApplication

Run: ApiApplication

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "Controle.java" file is selected and highlighted in blue.
- Terminal View:** Shows the path: api > src > main > java > br > com > projeto > api > controle > Controle.java.
- Code Editor:** Displays the Java code for the Controle.java class. The code includes annotations for PUT, DELETE, and GET mappings, as well as a method to get a counter value.

```
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNome();
57     }
58
59     @GetMapping("/")
60     public String mensagem(){
61         return "Hello World";
62     }
63
```

Salve va no navegador e coloque a rota ela deve ordenar



```
13     @Transactional  
14     → Controle | - 3 beans  
15     public interface Repositorio extends JpaRepository<Pessoa, Integer>{  
16         List<Pessoa> findAll();  
17         Pessoa findByCodigo(int codigo);  
18         List<Pessoa> br.com.projeto.api.repositorio.Repositorio.findByOrderByNome()  
19         List<Pessoa> findByOrderByNome();  
20     }  
21     |  
22     ↑
```

Podemos adicionar 2 palavras após a palavra de ordenar nome  
podemos usar Asc ordena letras de a – z  
Se for numeros do menor para o maior  
Asc é o padrão default

```
17  
18     Pessoa findByCodigo(int codigo);  
19  
20     List<Pessoa> findByOrderByNomeAsc();  
21  
22 }  
23
```

E temos o Desc que faz o contrario a ordem decrescente

File Edit Selection View Go Run Terminal Help Repository.java - APIS-SPRING-BOOT - Visual Studio Code

DATABASE 127.0.0.1@3306 8.0.25 api api\_spring 16k tables (1) pessoas columns codigo int idade varchar(255) nome varchar(255) index views procedures functions dbagenda sakila world

```
Repository.java X Controle.java 1 localhos New Request
```

```
api > src > main > java > br > com > projeto > api > repositorio > Repository.java > ...
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
→ Controle | -- 3 beans
14 public interface Repository extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNameDesc();
21
22 }
23
```

# Adicionamos o Desc no final e mudamos nos dois arquivos Repositorio.java e Controle.java

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a database connection to "127.0.0.1@3306" (version 8.0.25). Under the "api" schema, there is a "pessoas" table with columns: "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other tables listed are "dbagenda", "sakila", and "world".
- Code Editor (Right):** The file "Controle.java" is open, showing Java code for a REST API. The code includes annotations like @PutMapping, @DeleteMapping, and @GetMapping, and methods for saving, removing, and counting entities.
- Terminal (Top):** Shows "localhost:3001" and "Language Support for Java(TM) by Red Hat".
- Status Bar:** Shows "Controle.java - APIS-SPRING-BOOT - Visual Studio Code".

```
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNomeDesc();
57     }
58 }
```

File Edit Selection View Go Run Terminal Help Repository.java - APIS-SPRING-BOOT - Visual Studio Code

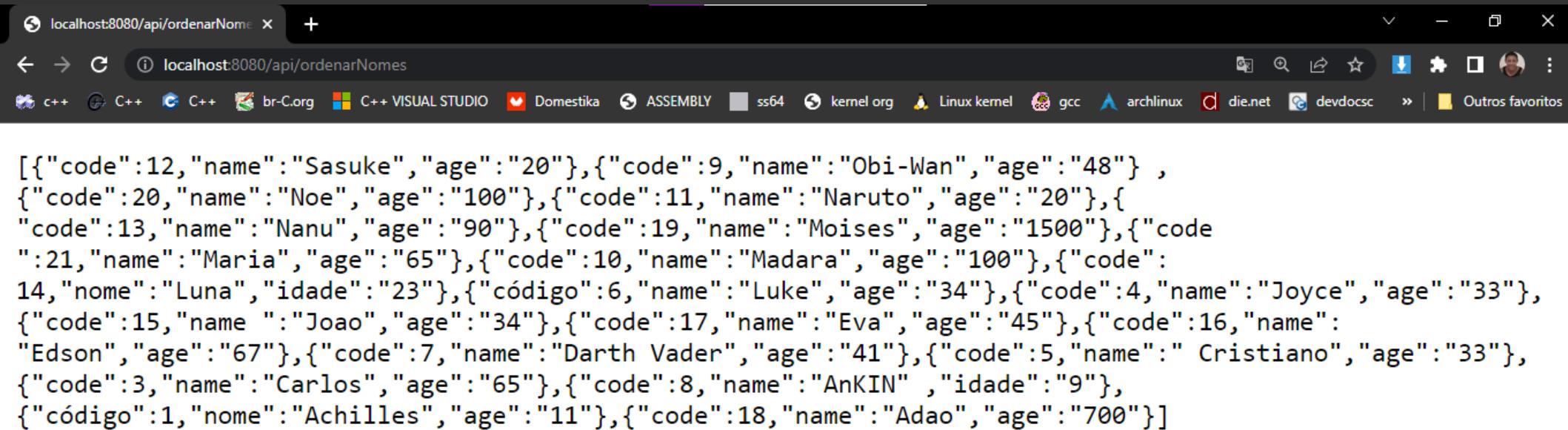
DATABASE 127.0.0.1@3306 8.0.25  
api\_spring 16k  
tables (1)  
pessoas  
columns  
codigo int  
idade varchar(255)  
nome varchar(255)  
index  
views  
procedures  
functions  
dbagenda  
sakila  
world

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repository extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNomeDesc();
21
22 }
23 }
```

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar with various icons for file operations like copy, paste, search, and refresh. The main area has a tab bar at the top with "Repositorio.java", "Controle.java X", and "localhost:3001". The "Controle.java" tab is active, displaying Java code for a Spring Boot application. The code defines several methods with annotations like @PostMapping, @DeleteMapping, and @GetMapping, interacting with a DAO named "acao". To the left of the code editor is a "DATABASE" sidebar showing a connection to "127.0.0.1@3306" with a schema "api\_spring" containing a table "pessoas" with columns "codigo", "idade", and "nome". Other schemas listed are "dbagenda", "sakila", and "world".

```
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNameDesc();
57     }
58 }
```

## Salvando e indo no navegador temos a ordem decrescente dos nomes



A screenshot of a web browser window. The address bar shows 'localhost:8080/api/ordenarNomes'. The page content displays a JSON array of objects, each containing 'code', 'name', and 'age' fields. The objects are sorted by name in descending order. The browser interface includes a toolbar with various icons and a bookmarks bar at the bottom.

```
[{"code":12,"name":"Sasuke","age":"20"}, {"code":9,"name":"Obi-Wan","age":"48"} ,  
 {"code":20,"name":"Noe","age":"100"}, {"code":11,"name":"Naruto","age":"20"}, {  
 "code":13,"name":"Nanu","age":"90"}, {"code":19,"name":"Moises","age":"1500"}, {"code":  
 :21,"name":"Maria","age":"65"}, {"code":10,"name":"Madara","age":"100"}, {"code":  
 14,"nome":"Luna","idade":"23"}, {"código":6,"name":"Luke","age":"34"}, {"code":4,"name":"Joyce","age":"33"},  
 {"code":15,"name ":"Joao","age":"34"}, {"code":17,"name":"Eva","age":"45"}, {"code":16,"name":  
 "Edson","age":"67"}, {"code":7,"name":"Darth Vader","age":"41"}, {"code":5,"name":" Cristiano","age":"33"},  
 {"code":3,"name":"Carlos","age":"65"}, {"code":8,"name":"AnKIN","idade":"9"},  
 {"código":1,"nome":"Achilles","age":"11"}, {"code":18,"name":"Adao","age":"700"}]
```

#23  
Ordenar registros parte 2

**Fazendo uma simulação restringindo as pessoas que eu quero  
Nesse caso vamos filtrar pessoas específicas e ordenar pela idade**

File Edit Selection View Go Run Terminal Help

• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code

The screenshot shows the Visual Studio Code interface. On the left, the Database Explorer sidebar is open, connected to '127.0.0.1@3306' (8.0.25). It lists the schema 'api\_spring' with one table 'pessoas'. The 'pessoas' table has three columns: 'codigo' (int), 'idade' (varchar(255)), and 'nome' (varchar(255)). Other database objects like 'index', 'views', 'procedures', 'functions', 'dbagenda', 'sakila', and 'world' are also listed. The main editor area shows the Java code for the 'Repositorio' interface.

```
package br.com.projeto.api.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;
//@Repository
@Transactional
public interface Repositorio extends JpaRepository<Pessoa, Integer>{
    List<Pessoa> findAll();
    Pessoa findByCodigo(int codigo);
    List<Pessoa> findByOrderByNomeDesc();
}
```

Repetimos os passos vamos retornar uma lista do tipo  
pessoa

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Active File:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code
- Left Sidebar:** Shows a database connection to "127.0.0.1@3306 8.0.25" and a table named "pessoas" with columns "codigo", "idade", and "nome". Other tables like "dbagenda" and "sakila" are also listed.
- Code Editor:** Displays the following Java code for a repository interface:

```
package br.com.projeto.api.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;
//@Repository
@Transactional
public interface Repository extends JpaRepository<Pessoa, Integer>{
    List<Pessoa> findAll();
    Pessoa findByCodigo(int codigo);
    List<Pessoa> findByOrderByNomeDesc();
    List<Pessoa> findByNome
}
```

A tooltip is visible at the bottom right of the code editor, listing two methods: `findByNList` and `findByNPessoas`.

E passamos a caracteristica que queremos filtrar nesse caso  
Nome

The screenshot shows a Visual Studio Code interface with a database sidebar on the left and a code editor on the right.

**Database Sidebar:**

- Connected to `127.0.0.1@3306` (version 8.0.25).
- Selected database: `api_spring` (16k).
- Tables (1): `pessoas`.
  - Columns:
    - `codigo` int
    - `idade` varchar(255)
    - `nome` varchar(255)
  - Index
  - Views
  - Procedures
  - Functions
  - dbagenda
  - sakila
  - world

**Code Editor:**

File API - Repository.java - APIS-SPRING-BOOT - Visual Studio Code

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repository extends JpaRepository<Pessoa, Integer> {
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNameDesc();
21
22     List<Pessoa> findByNomeOrderByIdade
23 }
24
25 }
```

A tooltip is visible at the bottom of the code editor, showing two method suggestions:

- [e] `findByNomeOrderByIdade`: `List<br.com.projeto.a...`
- [e] `findByNomeOrderByIdadeList`: `List<br.com.projeto.a...`

Em seguida escolhemos a segunda caracteristica Idade e podemos depois dela usar o Desc ou Asc

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                - Pessoa.java
              - modelo
              - repositorio
                - Repositorio.java
    - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
  - Terminal:** localhost:3001
  - Code Editor:** The file "Repositorio.java" is open, showing Java code for a repository interface. The code includes imports for List, JpaRepository, CrudRepository, Repository, Pessoa, and Transactional. It defines a public interface named "Repositorio" that extends JpaRepository<Pessoa, Integer>. The interface contains methods for findAll(), findByCodigo(int codigo), findByOrderByNameDesc(), and findByNomeOrderByIdadeDesc(). A tooltip for "findByNomeOrderByIdadeDesc()" indicates it is a warning.

Temos que pegar esse nome que é um parametro string



EXPLORER

...

Repositorio.java X

Controle.java

localhost:

II

▼

▲

C

□

▼

TC

New Request



APIS-SPRING-BOOT



.vscode



api



.mvn



.vscode



src



main



java



br



com



projeto



api



controle



Controle.java



modelo



Pessoa.java



repositorio



Repositorio.java

ApiApplication.java

resources

test

target

.gitignore

.gitlab-ci.yml

```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 → Controle | ← 3 beans
15
16 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
17
18     List<Pessoa> findAll();
19
20     Pessoa findByCodigo(int codigo);
21
22     List<Pessoa> findByOrderByNameDesc();
23
24     List<Pessoa> findByNameOrderByidadeDesc(String nome); ←
25 }
```

Adicionei mais um akilles pra poder ordenar pelo mesmo nome com o diferencial idade

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** localhost:8080/pessoa - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** API-SPRING-BOOT project structure is visible, including .vscode, api, src, main, java, br, com, projeto, api, controle, Controle.java, and modelo.
- Request Panel:** A POST request is being made to <http://localhost:8080/api>. The Body tab is selected, showing the following JSON content:

```
1 {  
2   "codigo": 22,  
3   "nome": "Akilles",  
4   "idade": 65  
5 }
```
- Response Panel:** Status: 200 OK, Size: 43 Bytes, Time: 121 ms. The Response tab shows the same JSON object.

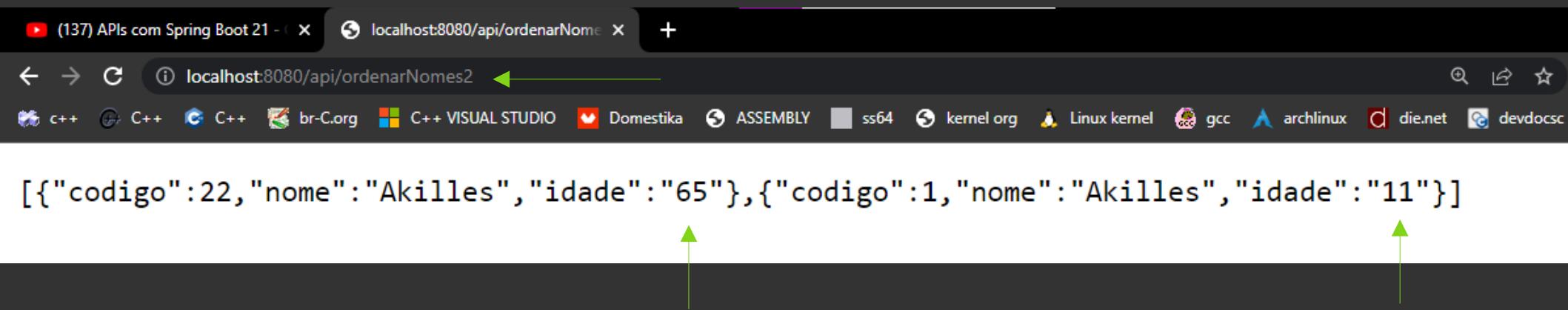
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Terminal:** Shows "localhost" and "New Request".
- Code Editor:** Displays the "Controle.java" file content.

```
• Controle.java - APIS-SPRING-BOOT - Visual Studio Code
REPOSITORY
Controle.java
localhost:3001
New Request
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNomeDesc();
57     }
58
59     @GetMapping("/api/ordenarNomes2")
60     public List<Pessoa> ordenarNomes2(){
61         return acao.findByNomeOrderByIdadeDesc(nome: "Akilles");
62     }
63
```

adicionamos a rota e o metodo depois é só ir no navegador

**Ordenamos pelo nome akilles e pela maior idade para a menor idade**



A screenshot of a web browser window. The address bar shows the URL `localhost:8080/api/ordenarNomes2`. The page content displays a JSON array: `[{"codigo":22,"nome":"Akilles","idade":65}, {"codigo":1,"nome":"Akilles","idade":11}]`. Two green arrows point upwards from the bottom of the slide towards the 'idade' field in the JSON data.

```
[{"codigo":22,"nome":"Akilles","idade":65}, {"codigo":1,"nome":"Akilles","idade":11}]
```

**Vamos mudar isso colocar a menor idade no caso do mais jovem para o mais velho**

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- api
  - > .mvn
  - > .vscode
- > src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - > resources
  - > test
  - > target
  - > .gitignore

Repositorio.java Controle.java X

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > ordenarNomes2()

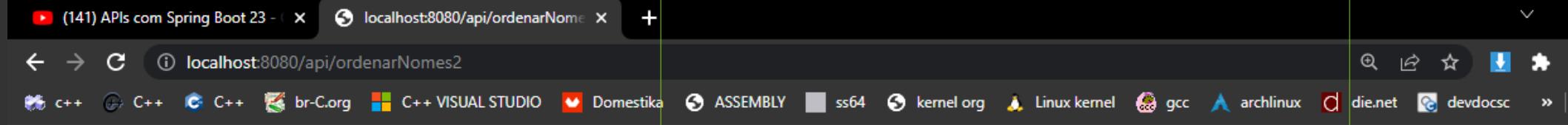
```
38     @PostMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNameDesc();
57     }
58
59     @GetMapping("/api/ordenarNomes2")
60     public List<Pessoa> ordenarNomes2(){
61         return acao.findByNomeOrderByidadeAsc(nome: "Akilles");
62     }
63 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
              - modelo
              - repositorio
                - Repositorio.java
  - resources
  - test
  - target
  - .gitignore
- Code Editor:** Displays the content of `Repositorio.java`.

```
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNomeDesc();
21
22     List<Pessoa> findByNomeOrderByIdadeAsc(String nome); ←
23
24 }
25 }
```

Lembre-se temos de mudar nos dois arquivos



Quem tem a menor idade agora é exibido primeiro  
Vamos fazer um ultimo test retirando o Asc

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View (Left):** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
                - repository
                  - Repository.java
    - resources
    - test
    - target
    - .aitianore
  - Editor View (Right):** Displays the Java code for the Repository interface.

```
package br.com.projeto.api.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;
//@Repository
@Transactional
public interface Repository extends JpaRepository<Pessoa, Integer>{
    List<Pessoa> findAll();
    Pessoa findByCodigo(int codigo);
    List<Pessoa> findByOrderByNameDesc();
    List<Pessoa> findByNameOrderByIdade(String nome);
}
```

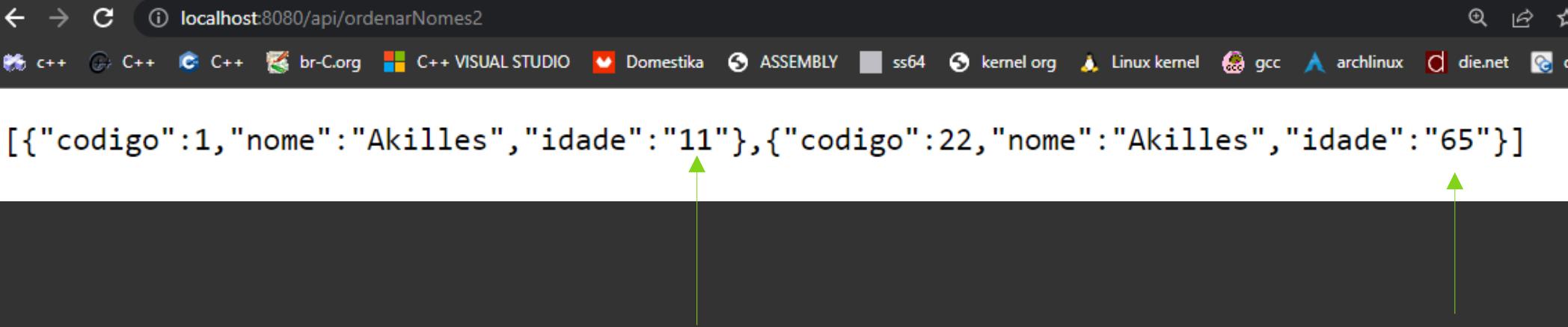
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
      - resources
      - test
      - target
      - .gitignore
- Code Editor:** The "Controle.java" file is open.

```
38     @PutMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
47     }
48
49     @GetMapping("/api/contador")
50     public Long contador(){
51         return acao.count();
52     }
53
54     @GetMapping("/api/ordenarNomes")
55     public List<Pessoa> ordenarNomes(){
56         return acao.findByOrderByNomeDesc();
57     }
58
59     @GetMapping("/api/ordenarNomes2")
60     public List<Pessoa> ordenarNomes2(){
61         return acao.findByNomeOrderByIdade(nome: "Akilles");
62     }
63 }
```

Salvamos e vamos no navegador

```
localhost:8080/api/ordenarNomes2
```



```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":22,"nome":"Akilles","idade":"65"}]
```

O de maior idade continua sendo exibido porque é o default do programa é o Asc

#24

Filtrar dados atravez do comando  
containing

**Exemplo quero listar todos os nomes que contenham a letra R  
Posso usar isso usando essa funcionalidade**

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code
- Terminal:** Shows the path: api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java
- Help:** Language Support for Java(TM) by Red Hat
- Explorer:** Shows the project structure:
  - APIS-SPRING-BOOT
    - .vscode
    - api
      - .mvn
      - .vscode
    - src
      - main
        - java
          - br
            - com
              - projeto
                - api
                  - controle
                  - Controle.java
                - modelo
                - Pessoa.java
              - repositorio
                - Repositorio.java
  - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
- Code Editor:** Displays the content of the Repositorio.java file:

```
1 package br.com.projeto.api.repositorio;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
6 //import org.springframework.data.repository.CrudRepository;
7 //import org.springframework.stereotype.Repository;
8
9 import br.com.projeto.api.modelo.Pessoa;
10 import jakarta.transaction.Transactional;
11
12 //@Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNomeDesc();
21
22     List<Pessoa> findByNomeOrderByIdade(String nome);
23
24     List<Pessoa> findByNomeContaining(String termo); ←
```

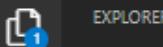
Começamos criamos uma função que retorna uma lista de pessoas

`FindByNomeContaining(String termo);`

O que eu quero procurar nesse  
caso o nome

Pois vamos  
procurar por  
uma letra

Função que  
procura



EXPLORER

&gt; .vscode

&gt; API-SPRING-BOOT

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

Controle.java

&gt; modelo

Pessoa.java

&gt; repository

Repositorio.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

Repositorio.java

Controle.java 2



api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; controle &gt; Controle.java &gt; Language Support for Java(TM) by Red Hat &gt; Controle &gt; nomeContem()

```
    46         acao.delete(obj));
    47     }
    48
    49     @GetMapping("/api/contador")
    50     public Long contador(){
    51         return acao.count();
    52     }
    53
    54     @GetMapping("/api/ordenarNomes")
    55     public List<Pessoa> ordenarNomes(){
    56         return acao.findByOrderByNomeDesc();
    57     }
    58
    59     @GetMapping("/api/ordenarNomes2")
    60     public List<Pessoa> ordenarNomes2(){
    61         return acao.findByNomeOrderByIdade(nome: "Akilles");
    62     }
    63
    64     public List<Pessoa> nomeContem(){
    65         return acao.
    66     }
```

Repositorio.findAll() : List&lt;Pessoa&gt;

Returns all instances of the type.

Repositorio.findAll() : List&lt;Pessoa&gt;

Returns all instances of the type.

## • Returns:

- all entities

PROBLEMS

OUTPUT

TERMINAL

TERMINAL

DEBUG

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
            - api
            - controle
          - modelo
          - Pessoa.java
          - repositorio
          - Repository.java
          - ApiApplication.java
      - resources
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Code Editor:** Controle.java (selected tab). The code is as follows:

```
46     acao.delete(id);
47 }
48
49 @GetMapping("/api/contador")
50 public Long contador(){
51     return acao.count();
52 }
53
54 @GetMapping("/api/ordenarNomes")
55 public List<Pessoa> ordenarNomes(){
56     return acao.findByOrderByNomeDesc();
57 }
58
59 @GetMapping("/api/ordenarNomes2")
60 public List<Pessoa> ordenarNomes2(){
61     return acao.findByNomeOrderByIdade(nome: "Akilles");
62 }
63
64 @GetMapping("/api/nomeContem")
65 public List<Pessoa> nomeContem(){
66     return acao.findByNomeContaining(termo: "1");
67 }
68 }
```

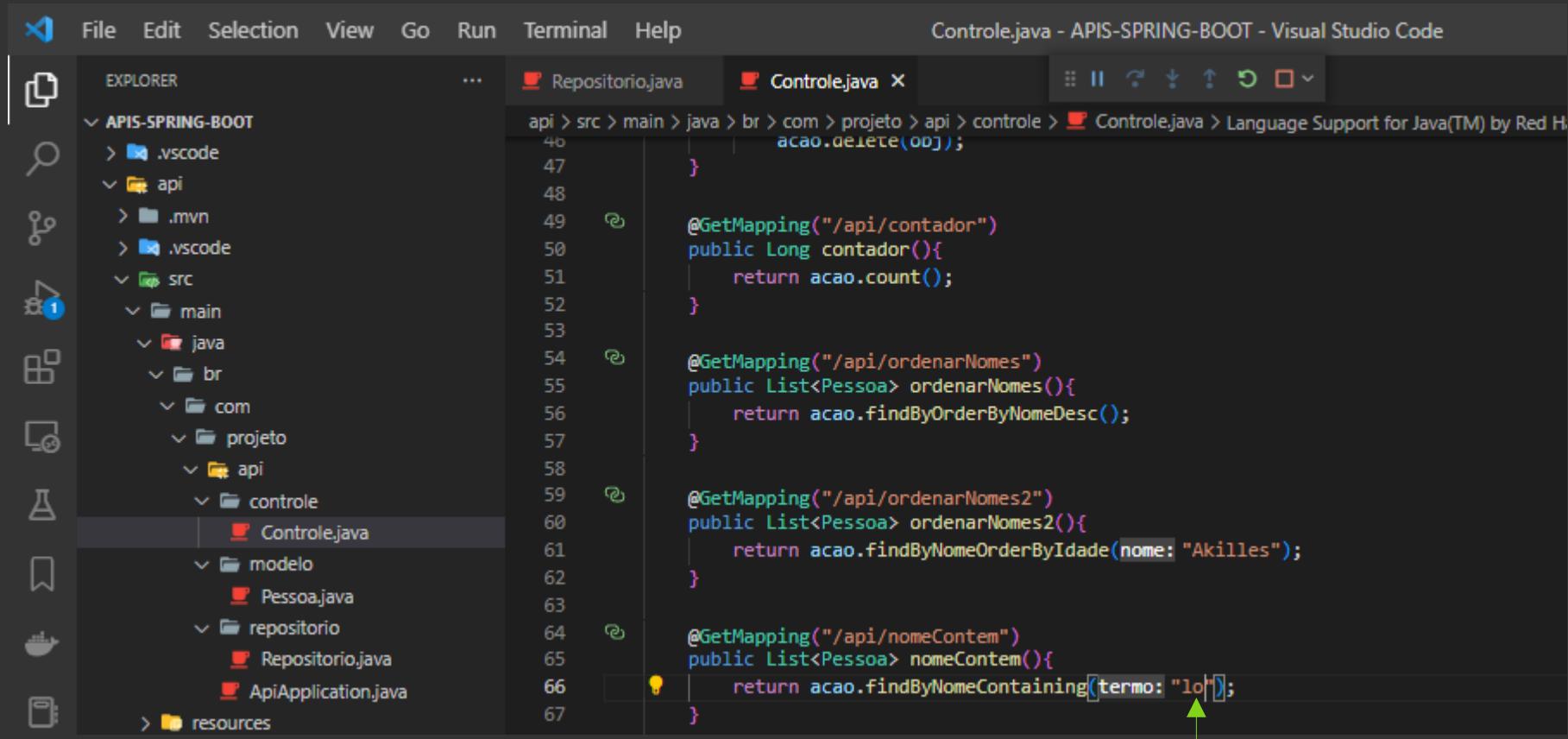
Em tabelas não importa a letra ser maiuscula ou minuscula nesse caso colocamos I mais poderiamos usar L tanto faz  
Salve e coloque a rota no navegador

Ele traz pra mim todos os nomes que possuem L em algum lugar

The screenshot shows a browser window with the address bar set to `localhost:8080/api/nomeContem`. The page content displays a JSON array of names and ages, where each name contains the letter 'L'. The names listed are Akilles, Carlos, Luke, and Luna.

```
[{"code":1,"name":"Akilles","age":"11"}, {"code":3,"name":"Carlos","age":"65"}, {"code":6,"name":"Luke","age":"34"}, {"code":14,"name":"Luna","age":"23"}, {"code":22,"name":"Akilles","age":"65"}]
```

# Adicionando um “o” logo depois do L ELE PROCURA POR PESSOAS QUE CONTENHAM LO EM ALGUM LUGAR



The screenshot shows the Visual Studio Code interface with the following details:

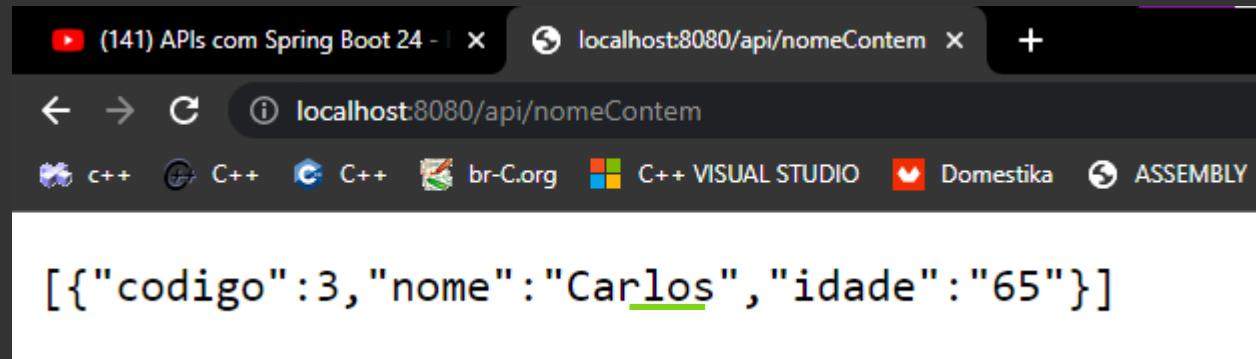
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
  - src folder contains .mvn, .vscode, and java.
  - java folder contains br, com, projeto, and api.
  - api folder contains controle, modelo, repository, and ApiApplication.java.
  - controle folder contains Controle.java.
  - repository folder contains Repositorio.java.
  - resources folder.
- Code Editor:** Displays the Controle.java file with the following code:

```
46     acao.delete(o);
47 }
48
49 @GetMapping("/api/contador")
50 public Long contador(){
51     return acao.count();
52 }
53
54 @GetMapping("/api/ordenarNomes")
55 public List<Pessoa> ordenarNomes(){
56     return acao.findByOrderByNomeDesc();
57 }
58
59 @GetMapping("/api/ordenarNomes2")
60 public List<Pessoa> ordenarNomes2(){
61     return acao.findByNomeOrderByIdade(nome: "Akilles");
62 }
63
64 @GetMapping("/api/nomeContem")
65 public List<Pessoa> nomeContem(){
66     return acao.findByNomeContaining(termo: "lo");
67 }
```

A yellow warning icon is present at line 66, character 11, indicating a potential issue with the code.

NESSE CASO SÓ TENHO O CARLOS “LO”

**Salvando e indo no navegador temos**

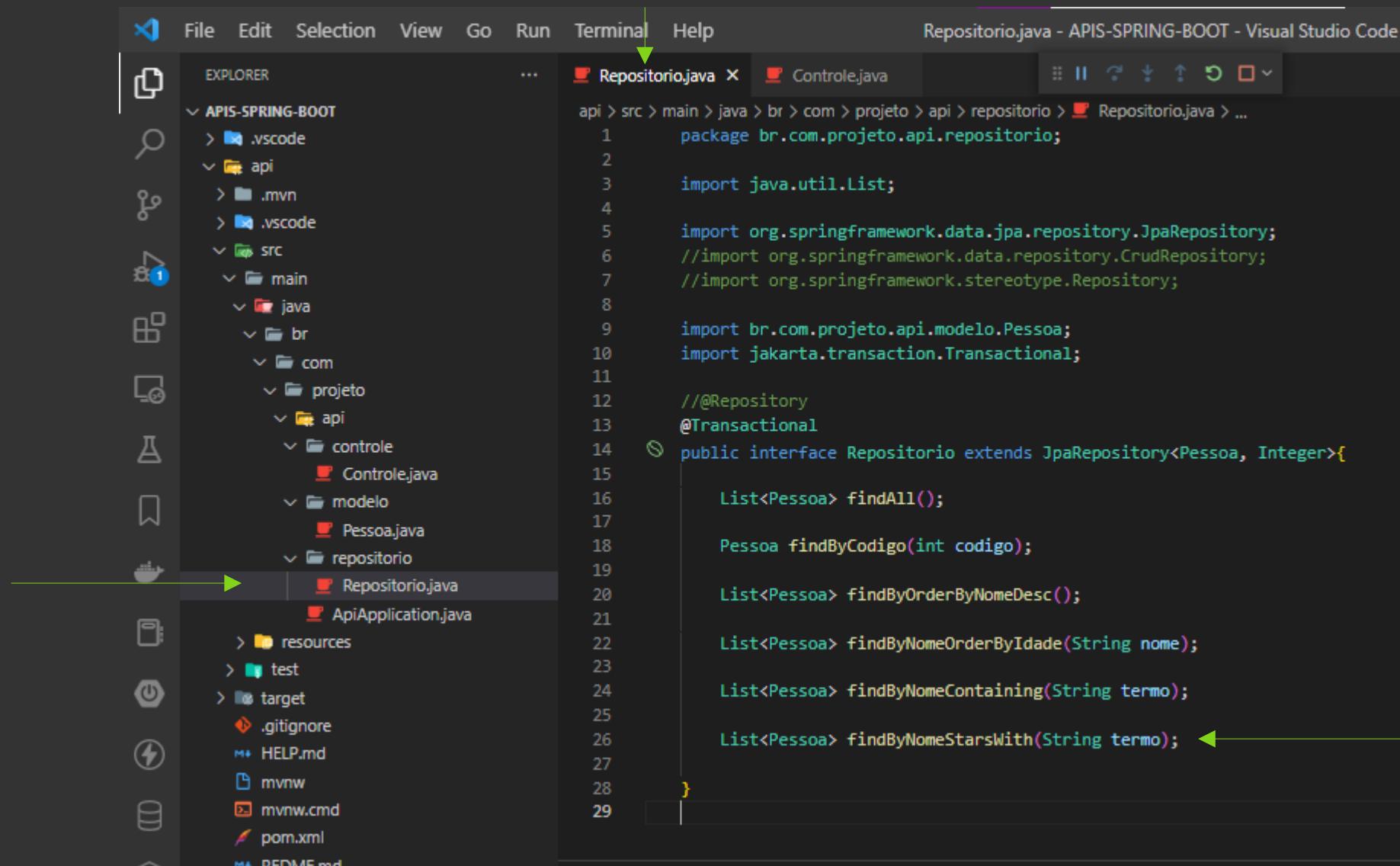


A screenshot of a dark-themed browser window. The address bar shows the URL `localhost:8080/api/nomeContem`. The main content area displays a JSON array:

```
[{"codigo":3,"nome":"Carlos","idade":"65"}]
```

#25  
**StarsWith e EndsWith**  
São funções

**Com essas funções podemos filtrar por coisas que começam ou finalizam com alguma coisa que queremos**



**Ou seja se eu digitar o termo “a” todos os nomes que iniciam com a letra a serão listados**



EXPLORER

## APIS-SPRING-BOOT

&gt; .vscode

api

&gt; .mvn

&gt; .vscode

src

main

java

br

com

projeto

api

controle

Controle.java

modelo

Pessoa.java

repositorio

Repositorio.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

README.md

...

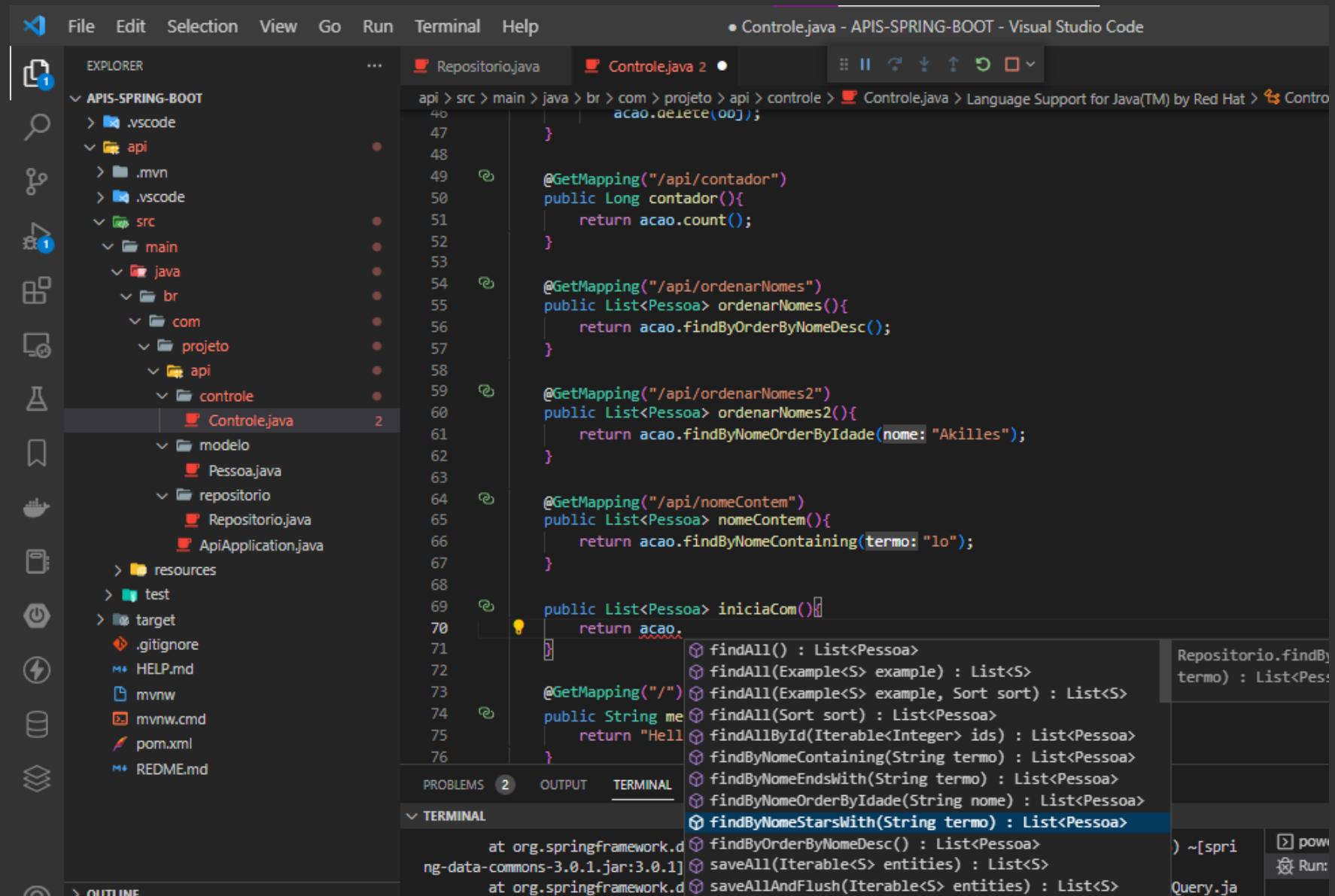
Repositorio.java X

Controle.java



```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > Repo
      1 package br.com.projeto.api.repositorio;
      2
      3 import java.util.List;
      4
      5 import org.springframework.data.jpa.repository.JpaRepository;
      6 //import org.springframework.data.repository.CrudRepository;
      7 //import org.springframework.stereotype.Repository;
      8
      9 import br.com.projeto.api.modelo.Pessoa;
     10 import jakarta.transaction.Transactional;
     11
     12 //@Repository
     13 @Transactional
     14 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
     15
     16     List<Pessoa> findAll();
     17
     18     Pessoa findByCodigo(int codigo);
     19
     20     List<Pessoa> findByOrderByNameDesc();
     21
     22     List<Pessoa> findByNameOrderByIdade(String nome);
     23
     24     List<Pessoa> findByNameContaining(String termo);
     25
     26     List<Pessoa> findByNameStarsWith(String termo);
     27
     28     List<Pessoa> findByNameEndsWith(String termo); ←
     29
     30 }
```

**De igual modo o end se eu digitar o termo “a” todos os nomes que terminam com o termo a serão listados**



File Edit Selection View Go Run Terminal Help • Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
              - modelo
                - Pessoa.java
              - repositorio
                - Repositorio.java
  - resources
  - test
  - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md

```
Repositorio.java  Controle.java
```

```
    46          acao.delete(obj);  
    47      }  
    48  
    49      @GetMapping("/api/contador")  
    50      public Long contador(){  
    51          return acao.count();  
    52      }  
    53  
    54      @GetMapping("/api/ordenarNomes")  
    55      public List<Pessoa> ordenarNomes(){  
    56          return acao.findByOrderByNomeDesc();  
    57      }  
    58  
    59      @GetMapping("/api/ordenarNomes2")  
    60      public List<Pessoa> ordenarNomes2(){  
    61          return acao.findByNomeOrderByIdade(nome: "Akilles");  
    62      }  
    63  
    64      @GetMapping("/api/nomeContem")  
    65      public List<Pessoa> nomeContem(){  
    66          return acao.findByNomeContaining(termo: "lo");  
    67      }  
    68  
    69      @GetMapping("/api/iniciaCom")  
    70      public List<Pessoa> iniciaCom(){  
    71          return acao.findByNomeStarsWith(termo: "a");  
    72      }  
    73  
    74      @GetMapping("/")  
    75      public String mensagem(){  
    76          return "Hello World";
```

Criamos a rota  
pra listar todas as  
pessoas que  
iniciam com a  
letra “a”  
independente de  
ser maiuscula ou  
minuscula



# De igual modo as que terminam com “a”

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
            - controle
              - Controle.java
  - resources
  - test
  - target
  - .gitignore

Repositorio.java Controle.java

```
58
59     @GetMapping("/api/ordenarNomes2")
60     public List<Pessoa> ordenarNomes2(){
61         return acao.findByNomeOrderByNome("Akilles");
62     }
63
64     @GetMapping("/api/nomeContem")
65     public List<Pessoa> nomeContem(){
66         return acao.findByNomeContaining(termo: "lo");
67     }
68
69     @GetMapping("/api/iniciaCom")
70     public List<Pessoa> iniciaCom(){
71         return acao.findByNomeStarsWith(termo: "a");
72     }
73
74     @GetMapping("/api/terminaCom")
75     public List<Pessoa> terminaCom(){
76         return acao.findByNomeStarsWith(termo: "a");
77     }
78
79     @GetMapping("/")
80     public String mensagem(){
81         return "Hello World";
82     }
```

Salvamos e vamos no navegador

**Deu um erro vou resolver**

(143) APIs com Spring Boot 25 - x localhost:8080/api x +

← → C i localhost:8080/api



c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel.org Linux kernel gcc archlinux die.net devdocsc » | Outros favoritos

```
[{"codigo":1,"nome":"Akilles",  
 {"codigo":4,"nome":"Joyce","ida  
 {"codigo":6,"nome":"Luke","ida  
 {"codigo":8,"nome":"AnKIN","ida  
 {"codigo":10,"nome":"Madara",  
 {"codigo":12,"nome":"Sasuke",  
 {"codigo":14,"nome":"Luna","ida  
 {"codigo":16,"nome":"Edson","ida  
 {"codigo":18,"nome":"Adao","ida  
 {"codigo":20,"nome":"Noe","ida  
 {"codigo":22,"nome":"Akilles",
```

localhost:8080/api/nomeContem x +

← → C i localhost:8080/api/nomeContem



c++ C++ C++ br-C.org C++ VISUAL STUDIO » | Outros favoritos

```
[{"codigo":3,"nome":"Carlos","idade":"65"}]
```

"},"},

Eu tinha digitado errado



EXPLORER



...

ApiApplication.java

Repositorio.java

Con

||

^

v

↑

↓

↻

□

▼

APIS-SPRING-BOOT

&gt; .vscode

&lt; API SPRING BOOT

&gt; .mvn

&gt; .vscode

&lt; API SPRING BOOT

```
57 }  
58  
59 http://127.0.0.1:8080/api/ordenarNomes2  
60 @GetMapping("/api/ordenarNomes2")  
61 public List<Pessoa> ordenarNomes2(){  
62     return acao.findByNomeOrderByidade(nome: "Akilles");  
63 }  
  
64 http://127.0.0.1:8080/api/nomeContem  
65 @GetMapping("/api/nomeContem")  
66 public List<Pessoa> nomeContem(){  
67     return acao.findByNomeContaining(termo: "lo");  
68 }  
  
69 // @GetMapping("/api/iniciaCom")  
70 // public List<Pessoa> iniciaCom(){  
71 //     return acao.findByNomeStarsWith("a");  
72 // }  
  
73 // @GetMapping("/api/terminaCom")  
74 // public List<Pessoa> terminaCom(){  
75 //     return acao.findByNomeStarsWith("a");  
76 // }  
  
77 http://127.0.0.1:8080/  
78 @GetMapping("/")  
79 public String mensagem(){  
80     return "Hello World";  
81 }  
  
82 http://127.0.0.1:8080/boasVindas  
83 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
                - repositorio
                  - Repositorio.java
    - resources
    - test
    - target
    - .gitignore
  - Code Editor:** The "Repositorio.java" file is open.

```
import jakarta.transaction.Transactional;
import org.springframework.stereotype.Repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.QueryHints;
import org.springframework.transaction.annotation.Transactional;

public interface Repositorio extends JpaRepository<Pessoa, Integer> {

    List<Pessoa> findAll();

    Pessoa findByCodigo(int codigo);

    List<Pessoa> findByOrderByNomeDesc();

    List<Pessoa> findByNameOrderByIdade(String nome);

    List<Pessoa> findByNameContaining(String termo);

    List<Pessoa> findByNameStartsWith(String termo);

    List<Pessoa> findByNameEndsWith(String termo);

}
```

```
3     List<Pessoa> findByNomeContaining(String termo);  
4  
5     List<Pessoa> findByNomeStartsWith(String termo);   
6  
7     List<Pessoa> findByNomeEndsWith(String termo);  
8  
9 }  
0
```

Tinha digitado errado **findByNomeStartsWith**



EXPLORER

...

ApiApplication.java

Repositorio.java

Con

II

?

▼

▲

↻

□

▼

## APIS-SPRING-BOOT

&gt; .vscode

&gt; api

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

Controle.java

&gt; modelo

Pessoa.java

&gt; repositorio

Repositorio.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

```
    57     }
    58
    59     http://127.0.0.1:8080/api/ordenarNomes2
    60     @GetMapping("/api/ordenarNomes2")
    61     public List<Pessoa> ordenarNomes2(){
    62         return acao.findByNomeOrderByidade(nome: "Akilles");
    63
    64     http://127.0.0.1:8080/api/nomeContem
    65     @GetMapping("/api/nomeContem")
    66     public List<Pessoa> nomeContem(){
    67         return acao.findByNomeContaining(termo: "lo");
    68
    69     http://127.0.0.1:8080/api/iniciaCom
    70     @GetMapping("/api/iniciaCom")
    71     public List<Pessoa> iniciaCom(){
    72         return acao.findByNomeStartsWith(termo: "a");
    73
    74     http://127.0.0.1:8080/api/terminaCom
    75     @GetMapping("/api/terminaCom")
    76     public List<Pessoa> terminaCom(){
    77         return acao.findByNomeEndsWith(termo: "a");
    78
    79     http://127.0.0.1:8080/
    80     @GetMapping("/")
    81     public String mensagem(){
    82         return "Hello World";
    83
```

## Depois de escrever certo salvamos e colocamos a rota no navegador

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - API-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
                - repositorio
                  - Repositorio.java
    - resources
    - test
- Code Editor:** The "Repositorio.java" file is open, showing the following Java code:

```
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;
import jakarta.transaction.Transactional;
import java.util.List;
import org.springframework.stereotype.Repository;

@Repository
@Transactional
public interface Repositorio extends JpaRepository<Pessoa, Integer>{

    List<Pessoa> findAll();

    Pessoa findByCodigo(int codigo);

    List<Pessoa> findByOrderByNomeDesc();

    List<Pessoa> findByNomeOrderByIdade(String nome);

    List<Pessoa> findByNomeContaining(String termo);

    List<Pessoa> findByNomeStartsWith(String termo);

    List<Pessoa> findByNomeEndsWith(String termo);
}
```

```
localhost:8080/api/iniciaCom x +  
localhost:8080/api/iniciaCom  
[{"code":1,"name":"Akilles","age":"11"}, {"code":8,"name":"AnKIN","age":"9"}, {"code":18,"name":"Adao","age":"700"}, {"code":22,"name":"Akilles","age":"65"}]
```

```
"nome":"Carlos","idade":"65"},  
me":"Cristiano","idade":"33"},  
e":"Darth Vader","idade":"41"},  
e":"Obi-Wan","idade":"48"},  
, "nome":"Naruto","idade":"20"},  
"nome":"Nanu","idade":"90"},
```

```
localhost:8080/api/terminaCom x +  
localhost:8080/api/terminaCom  
[{"codigo":10,"nome":"Madara","idade":"100"}, {"codigo":14,"nome":"Luna","idade":"23"}, {"codigo":17,"nome":"Eva","idade":"45"}, {"codigo":21,"nome":"Maria","idade":"65"}]
```

Pegamos todos os nome que inicial com “a” e terminam com “a”

O erro foi porque digitei errado os `findByNameStartsWith`

#26  
**Utilizando annotation @Query**

# Criamos um metodo que retorna um numero inteiro E se chama somaldades

File Edit Selection View Go Run Terminal Help

REPOSITORY.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
          - controle
            - Controle.java
          - modelo
            - Pessoa.java
          - repositorio
            - Repositorio.java
            - ApiApplication.java
  - resources
  - test
  - target

api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat

```
10 import jakarta.transaction.Transactional;
11
12 // @Repository
13 @Transactional
14 public interface Repositorio extends JpaRepository<Pessoa, Integer> {
15
16     List<Pessoa> findAll();
17
18     Pessoa findByCodigo(int codigo);
19
20     List<Pessoa> findByOrderByNameDesc();
21
22     List<Pessoa> findByNameOrderByidade(String nome);
23
24     List<Pessoa> findByNameContaining(String termo);
25
26     List<Pessoa> findByNameStartsWith(String termo);
27
28     List<Pessoa> findByNameEndsWith(String termo);
29
30     int somaIdades();
31 }
32 }
```

Queremos somar todas as idades

File Edit Selection View Go Run Terminal Help • Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < API-SPRING-BOOT
- > api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- < controle
- Controle.java
- < modelo
- Pessoa.java
- < repositorio
- Repositorio.java

ApiApplication.java

Repositorio.java 1

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > * Repositorio > somadades  
import jakarta.transaction.Transactional;  
//@Repository  
@Transactional  
→ Controle | ← 3 beans  
public interface Repositorio extends JpaRepository<Pessoa, Integer>{  
    List<Pessoa> findAll();  
    Pessoa findByCodigo(int codigo);  
    List<Pessoa> findByOrderByNomeDesc();  
    List<Pessoa> findByNomeOrderByIdade(String nome);  
    List<Pessoa> findByNomeContaining(String termo);  
    List<Pessoa> findByNomeStartsWith(String termo);  
    List<Pessoa> findByNomeEndsWith(String termo);  
}  
int @Query - org.springframework.data.jpa.repository  
• Query - org.springframework.data.jpa.repository  
• QueryHints - org.springframework.data.jpa.repository  
• QueryAnnotation - org.springframework.data.annotation  
• QueryHint - jakarta.persistence  
• QuerydslPredicate - org.springframework.data.queryDSL  
Annotation to declare finder queries directly on repository methods.  
• Author:  
• Oliver Gierke  
• Thomas Darimont
```

Adicionamos a anotation e seu import



EXPLORER

◀ APIS-SPRING-BOOT

>  .vscode

api

> mvn

>  .vscode

src

main

java

br

v com

 projeto

×  ani

controle

Control

 modelo

Pessoal Jav

 repositorio

Depositorio.java

— Ap

585

test

Be target

citizen

• 53

1

ANSWER

normal

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support
  1 import org.springframework.data.jpa.repository.JpaRepository;
  2 //import org.springframework.data.repository.CrudRepository;
  3 //import org.springframework.stereotype.Repository;
  4 import org.springframework.data.jpa.repository.Query; ◀
  5
  6
  7
  8
  9
 10 import br.com.projeto.api.modelo.Pessoa;
 11 import jakarta.transaction.Transactional;
 12
 13 // @Repository
 14 @Transactional
 15 → Controler | ← 3 beans
 16 public interface Repositorio extends JpaRepository<Pessoa, Integer>[

 17     List<Pessoa> findAll();
 18
 19     Pessoa findByCodigo(int codigo);
 20
 21     List<Pessoa> findByOrderByNomeDesc();
 22
 23     List<Pessoa> findByNomeOrderByIdade(String nome);
 24
 25     List<Pessoa> findByNomeContaining(String termo);
 26
 27     List<Pessoa> findByNomeStartsWith(String termo);
 28
 29     List<Pessoa> findByNomeEndsWith(String termo);
 30
 31     @Query| ◀
 32     int somaIdades();
 33
 34 ]
```



File Edit Selection View Go Run Terminal Help

• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code



EXPLORER

APIS-SPRING-BOOT

> .vscode

✓ api

> .mvn

> .vscode

✓ src

> main

> java

> br

> com

> projeto

> api

> controle

Controle.java

> modelo

Pessoa.java

> repositorio

Repositorio.java

ApiApplication.java

> resources

> test

> target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml



ApiApplication.java



Repositorio.java



Con



II



Q



▼



↑



○



□



▼

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > ...
      import org.springframework.data.repository.CrudRepository;
      //import org.springframework.stereotype.Repository;
      import org.springframework.data.jpa.repository.Query;
      import br.com.projeto.api.modelo.Pessoa;
      import jakarta.transaction.Transactional;

      //@Repository
      @Transactional
      → Controle | ← 3 beans
public interface Repositorio extends JpaRepository<Pessoa, Integer>{

    List<Pessoa> findAll();

    Pessoa findByCodigo(int codigo);

    List<Pessoa> findByOrderByNameDesc();

    List<Pessoa> findByNameOrderByIdade(String nome);

    List<Pessoa> findByNameContaining(String termo);

    List<Pessoa> findByNameStartsWith(String termo);

    List<Pessoa> findByNameEndsWith(String termo);

    @Query(value = "SELECT SUM(idade) FROM pessoas")
    int somaIdades();
}
```

**Note que o value da Query é praticamente um código sql  
Qualquer comando sql funciona**

File Edit Selection View Go Run Terminal Help • Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle
- Controle.java
- modelo
- Pessoa.java
- repositorio
- Repositorio.java
- ApiApplication.java

ApiApplication.java

Repositorio.java 2

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > Repositorio > somaldades()  
import org.springframework.data.repository.JpaRepository;  
import org.springframework.data.repository.CrudRepository;  
import org.springframework.stereotype.Repository;  
import org.springframework.data.jpa.repository.Query;  
  
import br.com.projeto.api.modelo.Pessoa;  
import jakarta.transaction.Transactional;  
  
//@Repository  
@Transactional  
→ Controle | ← 3 beans  
public interface Repositorio extends JpaRepository<Pessoa, Integer>{  
  
    List<Pessoa> findAll();  
  
    Pessoa findByCodigo(int codigo);  
  
    List<Pessoa> findByOrderByNameDesc();  
  
    List<Pessoa> findByNameOrderByidade(String nome);  
  
    List<Pessoa> findByNameContaining(String termo);  
  
    List<Pessoa> findByNameStartsWith(String termo);  
  
    List<Pessoa> findByNameEndsWith(String termo);  
  
    @Query(value = "SELECT SUM(idade) FROM pessoas", nativeQuery : boolean)  
    int somaIdad  
    Configures whether the given query is a native one.  
    Defaults to false.
```

Ativamos a query com um valor booleano

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
              - Controle.java
            - modelo
            - Pessoa.java
          - repositorio
            - Repositorio.java
  - resources
  - test
  - target
    - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

```
ApiApplication.java Repository.java Con ... II ? ^ C □ v
api > src > main > java > br > com > projeto > api > repositorio > Repository.java > ...
5   import org.springframework.data.repository.CrudRepository;
6   //import org.springframework.stereotype.Repository;
7   //import org.springframework.data.jpa.repository.Query;
8
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import jakarta.transaction.Transactional;
12
13 //@Repository
14 @Transactional
→ Controle | ← 3 beans
15 public interface Repository extends JpaRepository<Pessoa, Integer>{
16
17     List<Pessoa> findAll();
18
19     Pessoa findByCodigo(int codigo);
20
21     List<Pessoa> findByOrderByNameDesc();
22
23     List<Pessoa> findByNameOrderByIdade(String nome);
24
25     List<Pessoa> findByNameContaining(String termo);
26
27     List<Pessoa> findByNameStartsWith(String termo);
28
29     List<Pessoa> findByNameEndsWith(String termo);
30
31     @Query(value = "SELECT SUM(idade) FROM pessoas", nativeQuery = true)
32     int somaIdades();
33
34 }
```



EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

&gt; api

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

Controle.java



ApiApplication.java

Repositorio.java

Con

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java
    67
    68
    69 @GetMapping("/api/iniciaCom")
    70     public List<Pessoa> iniciaCom(){
    71         return acao.findByNomeStartsWith(termo: "a");
    72     }
    73
    74 @GetMapping("/api/terminaCom")
    75     public List<Pessoa> terminaCom(){
    76         return acao.findByNomeEndsWith(termo: "a");
    77     }
    78
    79     public int somaIdades(){
    80         return acao.
    81     }
    82
    83     http://127.0.0.1:8080/
    84 @GetMapping("/")
    85     public String me()
    86         return "Hello";
    87
    88     http://127.0.0.1:8080/b
    89 @GetMapping("/bo")
    90     public String bo()
    91         return "Seja";
    92
```

Repositorio.som

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure: APIS-SPRING-BOOT, .vscode, api, .mvn, .vscode, src, main, java, br, com, projeto, api, controle, Controle.java (selected), modelo, Pessoa.java, repository, Repositorio.java, ApiApplication.java, resources, test, target.
- Code Editor:** Displays Java code for the Controle.java file. The code includes methods for handling API requests and a method for summing ages.

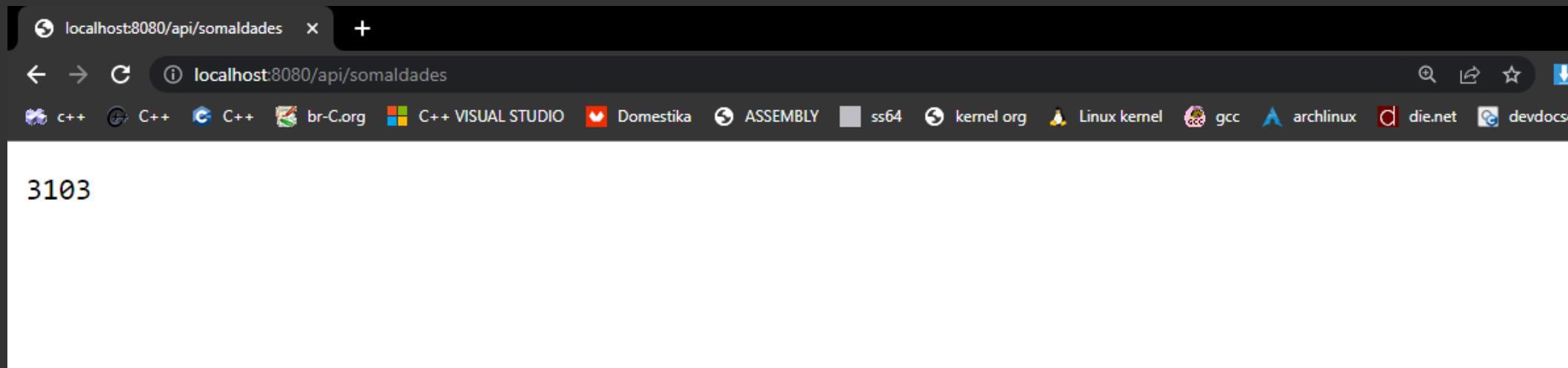
```
67    }
68
69    @GetMapping("/api/iniciaCom")
70    public List<Pessoa> iniciaCom(){
71        return acao.findByNomeStartsWith(termo: "a");
72    }
73
74    @GetMapping("/api/terminaCom")
75    public List<Pessoa> terminaCom(){
76        return acao.findByNomeEndsWith(termo: "a");
77    }
78
79    public int somaIdades(){
80        return acao.somaIdades();
81    }
82
83    http://127.0.0.1:8080/
84    @GetMapping("/")
85    public String mensagem(){
86        return "Hello World";
87    }
88
89    http://127.0.0.1:8080/boasVindas
90    @GetMapping("/boasVindas")
```

Criando o metodo e a rota

The screenshot shows the Visual Studio Code interface with the title "Controle.java - APIS-SPRING-BOOT - Visual Studio Code". The left sidebar contains icons for file operations, search, and project navigation. The Explorer view shows the project structure under "APIS-SPRING-BOOT", including ".vscode", "api", "src" (with "main" and "java" subfolders), "resources", "test", "target", and various configuration files like ".gitignore" and "HELP.md". The main editor area displays Java code for the "Controle.java" class:

```
67     }
68
69     @GetMapping("/api/iniciaCom")
70     public List<Pessoa> iniciaCom(){
71         return acao.findByNomeStartsWith(termo: "a");
72     }
73
74     @GetMapping("/api/terminaCom")
75     public List<Pessoa> terminaCom(){
76         return acao.findByNomeEndsWith(termo: "a");
77     }
78
79     @GetMapping("/api/somaIdades")
80     public int somaIdades(){
81         return acao.somaIdades();
82     }
83
84     http://127.0.0.1:8080/
85     @GetMapping("/")
86     public String mensagem(){
87         return "Hello World";
88     }
89
90     http://127.0.0.1:8080/boasVindas
91     @GetMapping("/boasVindas")
92     public String boasVindas(){
93         return "Seja bem vindo(a) ";
94     }
```

**Salvando e passando a rota no navegador temos**



#27  
**Utilizando query com parametros**

**A ideia é vamos passar uma idade e queremos retornar as pessoas com aquela idade ou superior**



EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

v api

&gt; .mvn

&gt; .vscode

v src

v main

v java

v br

v com

v projeto

v api

v controle

Controle.java

v modelo

Pessoa.java

v repository

Repositorio.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

... ApiApplication.java

Repositorio.java

Con

:

||

?

▼

▲

↻

□

▼

```
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > ...  
13     // @Repository  
14     @Transactional  
15     → Controle | -- 3 beans  
16     public interface Repositorio extends JpaRepository<Pessoa, Integer>{  
17         List<Pessoa> findAll();  
18  
19         Pessoa findByCodigo(int codigo);  
20  
21         List<Pessoa> findByOrderByNameDesc();  
22  
23         List<Pessoa> findByNomeOrderByIdade(String nome);  
24  
25         List<Pessoa> findByNomeContaining(String termo);  
26  
27         List<Pessoa> findByNomeStartsWith(String termo);  
28  
29         List<Pessoa> findByNomeEndsWith(String termo);  
30  
31         @Query(value = "SELECT SUM(idade) FROM pessoas", nativeQuery = true)  
32         int somaIdades();  
33  
34         List<Pessoa> idadeMaiorIgual(int idade); ←  
35     }  
36 }
```

Criamos um metodo que retorna uma lista de pessoas e recebe como parametro um inteiro idade

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Pessoa.java
                - repositorio
                  - Repository.java
                  - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
- Search View:** A magnifying glass icon.
- Problems View:** A lightbulb icon.
- Terminal View:** A terminal window icon.
- Help View:** A gear icon.
- Code Editor:** Displays the Java code for the Repository interface.

```
13     // @Repository
14     @Transactional
15     public interface Repository extends JpaRepository<Pessoa, Integer>{
16
17         List<Pessoa> findAll();
18
19         Pessoa findByCodigo(int codigo);
20
21         List<Pessoa> findByOrderByNomeDesc();
22
23         List<Pessoa> findByNomeOrderByIdade(String nome);
24
25         List<Pessoa> findByNomeContaining(String termo);
26
27         List<Pessoa> findByNomeStartsWith(String termo);
28
29         List<Pessoa> findByNomeEndsWith(String termo);
30
31         @Query(value = "SELECT SUM(idade) FROM pessoas", nativeQuery = true)
32         int somaIdades();
33
34         @Query(value = "SELECT * FROM pessoas WHERE idade >= :idade")
35         List<Pessoa> idadeMaiorIgual(int idade);
36
37 }
```

Depois dos : colocamos o nome do parametro que queremos pegar

File Edit Selection View Go Run Terminal Help Repository.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER ... ApiApplication.java Repository.java Con II ↴ ↑ ↵ □ ▾

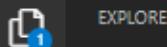
APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
              - modelo
                - Pessoa.java
              - repository
                - Repository.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

api > src > main > java > br > com > projeto > api > repository > Repository.java > ...

```
13     // @Repository
14     @Transactional
→ Controle | ← 3 beans
15     public interface Repository extends JpaRepository<Pessoa, Integer>{
16
17         List<Pessoa> findAll();
18
19         Pessoa findByCodigo(int codigo);
20
21         List<Pessoa> findByOrderByNomeDesc();
22
23         List<Pessoa> findByNomeOrderByIdade(String nome);
24
25         List<Pessoa> findByNomeContaining(String termo);
26
27         List<Pessoa> findByNomeStartsWith(String termo);
28
29         List<Pessoa> findByNomeEndsWith(String termo);
30
31         @Query(value = "SELECT SUM(idade) FROM pessoas", nativeQuery = true)
32         int somaIdades();
33
34         @Query(value = "SELECT * FROM pessoas WHERE idade >= :idade", nativeQuery = true) ←
35         List<Pessoa> idadeMaiorIgual(int idade);
36     }
37 }
```

Podemos passar quantos parametros quisermos



EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

api

&gt; .mvn

&gt; .vscode

&gt; src

main

java

br

com

projeto

api

controle

Controle.java

2

modelo

Pessoa.java

repositorio

Repository.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

ApiApplication.java

Repository.java

Controle.java

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > idadeMaiorIgual()  
67 }  
68  
69 @GetMapping("/api/iniciaCom")  
70 public List<Pessoa> iniciaCom(){  
71     return acao.findByNomeStartsWith(termo: "a");  
72 }  
73  
74 @GetMapping("/api/terminaCom")  
75 public List<Pessoa> terminaCom(){  
76     return acao.findByNomeEndsWith(termo: "a");  
77 }  
78  
79 @GetMapping("/api/somaIdades")  
80 public int somaIdades(){  
81     return acao.somaIdades();  
82 }  
83  
84 @GetMapping("/api/idadeMaiorIgual")  
85     public List<Pessoa> idadeMaiorIgual(){  
86         return acao.i  
87         ↳ idadeMaiorIgual(int idade) : List<Pessoa> ← →  
88         http://127.0.0.1:8080/  
89         @GetMapping("/")  
90         public String men  
91             return "Hello"  
92         }  
93         http://127.0.0.1:8080/ba  
94         @GetMapping("/boa")  
95         public String boasVindas()  
96             return "Bem-vindo ao projeto API"  
97         }
```

```
Repositorio.idadeMaiorIgual(int idade) : List<Pessoa>  
somaIdades() : int  
getById(Integer id) : Pessoa  
findById(Integer id) : Optional<Pessoa>  
deleteInBatch(Iterable<Pessoa> entities) : void  
getOne(Integer id) : Pessoa  
saveAll(Iterable<S> entities) : List<S>  
existsById(Integer id) : boolean  
deleteById(Integer id) : void  
findAllById(Iterable<Integer> ids) : List<Pessoa>
```



EXPLORER



APIS-SPRING-BOOT



.vscode



api



.mvn



.vscode



src



main



java



br



com



projeto



api



controle



Controle.java



modelo



Pessoa.java



repositorio



Repositorio.java



ApiApplication.java



resources



test



target



.gitignore



HELP.md



mvnw



mvnw.cmd



pom.xml



README.md

ApiApplication.java

Repositorio.java

Con

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; controle &gt; Controle.java &gt; Language Support for Java(TM) by Red Hat &gt; Controle.java

```
    67     }
    68
    69     @GetMapping("/api/iniciaCom")
    70     public List<Pessoa> iniciaCom(){
    71         return acao.findByNomeStartsWith(termo: "a");
    72     }
    73
    74     @GetMapping("/api/terminaCom")
    75     public List<Pessoa> terminaCom(){
    76         return acao.findByNomeEndsWith(termo: "a");
    77     }
    78
    79     @GetMapping("/api/somaIdades")
    80     public int somaIdades(){
    81         return acao.somaIdades();
    82     }
    83
    84     @GetMapping("/api/idadeMaiorIgual")
    85     public List<Pessoa> idadeMaiorIgual(){
    86         return acao.idadeMaiorIgual(idade: 18);
    87     }
    88
    89     @GetMapping("/")
    90     public String mensagem(){
    91         return "Hello World";
    92     }
    93
    94     @GetMapping("/boasVindas")
```

Salve e va no navegador  
adicone a rota

localhost:8080/api/idadeMaiorIgual X +

localhost:8080/api/idadeMaiorIgual

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel.org Linux kernel gcc archlinux die.net devdocsc

```
[{"codigo":3,"nome":"Carlos","idade":65}, {"codigo":4,"nome":"Joyce","idade":33}, {"codigo":5,"nome":"Cristiano","idade":33}, {"codigo":6,"nome":"Luke","idade":34}, {"codigo":7,"nome":"Darth Vader","idade":41}, {"codigo":9,"nome":"Obi-Wan","idade":48}, {"codigo":10,"nome":"Madara","idade":100}, {"codigo":11,"nome":"Naruto","idade":20}, {"codigo":12,"nome":"Sasuke","idade":20}, {"codigo":13,"nome":"Nanu","idade":90}, {"codigo":14,"nome":"Luna","idade":23}, {"codigo":15,"nome":"Joao","idade":34}, {"codigo":16,"nome":"Edson","idade":67}, {"codigo":17,"nome":"Eva","idade":45}, {"codigo":18,"nome":"Adao","idade":700}, {"codigo":19,"nome":"Moises","idade":1500}, {"codigo":20,"nome":"Noe","idade":100}, {"codigo":21,"nome":"Maria","idade":65}, {"codigo":22,"nome":"Akilles","idade":65}]
```

```
82     }
83
84     @GetMapping("/api/idadeMaiorIgual")
85     public List<Pessoa> idadeMaiorIgual(){
86         return acao.idadeMaiorIgual(idade: 65); ←
87     }
88
89     @GetMapping("/")
90     public String mensagem(){
91         return "Hello World";
92     }
93
94     http://127.0.0.1:8080/boasVindas
```

# Troquei a idade pra testar

localhost:8080/api/idadeMaiorIgual x +

localhost:8080/api/idadeMaiorIgual

c++ C++ C++ br-C.org C++ VISUAL STUDIO Domestika ASSEMBLY ss64 kernel org Linux kernel gcc archlinux die.net dev

```
[{"codigo":3,"nome":"Carlos","idade":65}, {"codigo":10,"nome":"Madara","idade":100}, {"codigo":13,"nome":"Nanu","idade":90}, {"codigo":16,"nome":"Edson","idade":67}, {"codigo":18,"nome":"Adao","idade":700}, {"codigo":19,"nome":"Moises","idade":1500}, {"codigo":20,"nome":"Noe","idade":100}, {"codigo":21,"nome":"Maria","idade":65}, {"codigo":22,"nome":"Akilles","idade":65}]
```

Agora eu recebo todas as idades acima de 65 anos

#28

ResponseEntity

Responsavel por customizar os estatus das nossas requisições



File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code



EXPLORER

APIS-SPRING-BOOT  
> .vscode  
  < api  
    > .mvn  
    > .vscode  
  < src  
    < main  
      < java  
        < br  
          < com  
            < projeto  
              < api  
                < controle  
                  Controle.java  
                model  
                  Pessoa.java  
                repository  
                  Repository.java  
                ApiApplication.java  
          > resources  
          > test  
          > target  
            .gitignore  
          HELP.md  
          mvnw  
          mvnw.cmd

...

ApiApplication.java

Repositorio.java

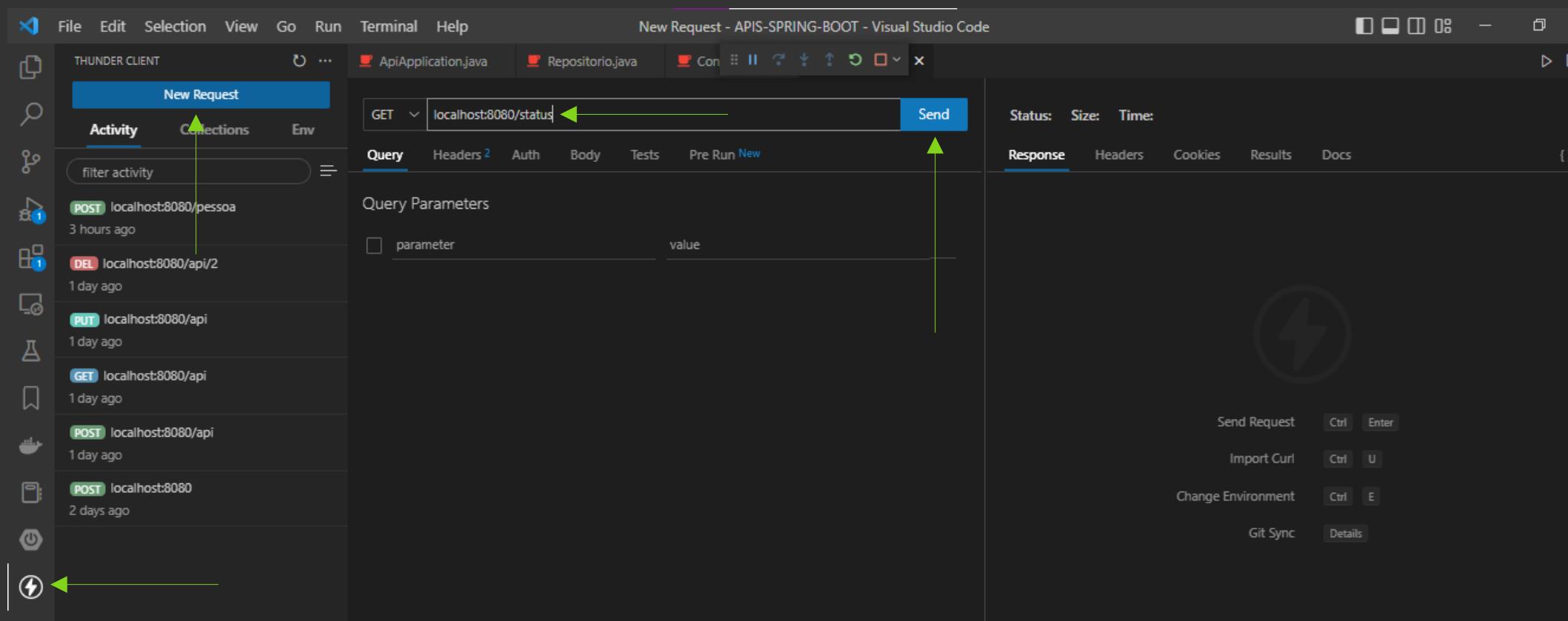
Controle.java

api > src > main > java > br > com > projeto > api > controle > Controle.java > ...

```
93  
94  @GetMapping("/boasVindas")  
95  public String boasVindas(){  
96      return "Seja bem vindo(a) ";  
97  }  
98  
99  @GetMapping("/boasVindas/{nome}")  
100 public String boasVindas(@PathVariable String nome){  
101     return "Seja bem vindo(a) " + nome;  
102 }  
103  
104 @PostMapping("/pessoa")  
105 public Pessoa pessoa(@RequestBody Pessoa p){  
106     return p;  
107 }  
108  
109 @GetMapping("/status")  
110 public String status(){  
111     return "Configurando status";  
112 }  
113  
114  
115 }  
116 }
```



Começamos criando uma rota simples que nos retorna um texto



Depois de salvar abrimos o thunder client e colocamos a rota que criamos e damos send

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/status just now

POST localhost:8080/pessoa 3 hours ago

DEL localhost:8080/api/2 1 day ago

PUT localhost:8080/api 1 day ago

GET localhost:8080/api 1 day ago

POST localhost:8080/api 1 day ago

POST localhost:8080 2 days ago

ApiApplication.java Repository.java Con

Send

Status: 200 OK Size: 19 Bytes Time: 56 ms

Response Headers 4 Cookies Results Docs

1 Configurando status

A screenshot of the Thunder Client extension in Visual Studio Code. The interface shows a sidebar with activity logs and a main panel for making API requests. In the main panel, a GET request to 'localhost:8080/status' is being sent. The response is shown with a status of 200 OK, a size of 19 bytes, and a time of 56 ms. The response body contains the text 'Configurando status'. A green arrow points upwards from the text 'A resposta é o texto que enviamos' at the bottom to the 'Configurando status' text in the response.

A resposta é o texto que enviamos

**Implementando um status diferente personalizado**

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

**GET** localhost:8080/status just now

**POST** localhost:8080/pessoa 3 hours ago

**DEL** localhost:8080/api/2 1 day ago

**PUT** localhost:8080/api 1 day ago

**GET** localhost:8080/api 1 day ago

**POST** localhost:8080/api 1 day ago

**POST** localhost:8080 2 days ago

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Oracle
```

93 http://127.0.0.1:8080/boasVindas  
94 @GetMapping("/boasVindas")  
95 public String boasVindas(){  
96 return "Seja bem vindo(a) ";  
97 }  
  
98 http://127.0.0.1:8080/boasVindas/{nome}  
99 @GetMapping("/boasVindas/{nome}")  
100 public String boasVindas(@PathVariable String nome){  
101 return "Seja bem vindo(a) " + nome;  
102 }  
  
103 http://127.0.0.1:8080/pessoa  
104 @PostMapping("/pessoa")  
105 public Pessoa pessoa(@RequestBody Pessoa p){  
106 return p;  
107 }  
  
108 @GetMapping("/status")  
109 public ResponseEntity<String> status(){  
110 return "Configuração OK";  
111 }

org.springframework.http.ResponseEntity ×

Extension of HttpEntity that adds an HttpStatusCode status code. Used in RestTemplate as well as in @Controller methods.

In RestTemplate , this class is returned by getForEntity() and exchange():

```
ResponseEntity<String> entity = template.get();  
String body = entity.getBody();  
MediaType contentType = entity.getHeaders().getContentType();  
HttpStatus statusCode = entity.getStatusCode();
```

This can also be used in Spring MVC as the return value from an @Controller method:

```
@RequestMapping("/handle")  
public ResponseEntity<String> handle() {  
    URI location = ...;  
    HttpHeaders responseHeaders = new HttpHeaders();  
    responseHeaders.setLocation(location);  
    responseHeaders.set("MyResponseHeader", "MyValue");  
    return new ResponseEntity<String>("Hello World!", responseHeaders, HttpStatus.OK);  
}
```

Or, by using a builder accessible via static methods:

```
@RequestMapping("/handle")  
public ResponseEntity<String> handle() {  
    URI location = ...;  
    return ResponseEntity.created(location);  
}
```

• Type Parameters:

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
  - APIS-SPRING-BOOT
  - .vscode
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
              - controle
    - Controle.java
    - modelo
    - Pessoa.java
    - repositorio
    - Repository.java
    - ApiApplication.java
    - resources
    - test
    - target
      - .gitignore
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
      - README.md
- Terminal View:** Shows the current file path: api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM).
- Code Editor View:** Displays the Java code for Controle.java. The code defines two endpoints for "boasVindas" and one for "pessoa". The "pessoa" endpoint uses @RequestBody Pessoa p. A cursor is positioned at the end of the "status" endpoint's return value.

```
94     http://127.0.0.1:8080/boasVindas
95     @GetMapping("/boasVindas")
96     public String boasVindas(){
97         return "Seja bem vindo(a) ";
98     }
99
100    http://127.0.0.1:8080/boasVindas/{nome}
101    @GetMapping("/boasVindas/{nome}")
102    public String boasVindas(@PathVariable String nome){
103        return "Seja bem vindo(a) " + nome;
104    }
105    http://127.0.0.1:8080/pessoa
106    @PostMapping("/pessoa")
107    public Pessoa pessoa(@RequestBody Pessoa p){
108        return p;
109    }
110
111    @GetMapping("/status")
112    public ResponseEntity<?> status(){
113        return ResponseEntity.ok("Configurando status");
114    }
115
116
117 }
```
- Bottom Status Bar:** PROBLEMS 1, OUTPUT, TERMINAL.

Removemos o String e colocamos  
ResponseEntity

<?>

**É o objeto que estamos deixando隐式**



EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
  - > .mvn
  - > .vscode
- < src
  - < main
    - < java
      - < br
        - < com
          - < projeto
            - < api
              - < controle
                - Controle.java
  - > resources
  - > test
  - > target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md



ApiApplication.java



Repositorio.java



Controle.java



Con



||



?



□



▼



↑



↻



✖

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat, Inc.
```

94     http://127.0.0.1:8080/boasVindas  
95     @GetMapping("/boasVindas")  
96     public String boasVindas(){  
97         return "Seja bem vindo(a) ";  
98     }  
  
99     http://127.0.0.1:8080/boasVindas/{nome}  
100    @GetMapping("/boasVindas/{nome}")  
101    public String boasVindas(@PathVariable String nome){  
102         return "Seja bem vindo(a) " + nome;  
103    }  
  
104    http://127.0.0.1:8080/pessoa  
105    @PostMapping("/pessoa")  
106    public Pessoa pessoa(@RequestBody Pessoa p){  
107         return p;  
108    }  
  
109    http://127.0.0.1:8080/status  
110    @GetMapping("/status")  
111    public ResponseEntity<?> status(){  
112         return new ResponseEntity<?>();  
113    }  
114    }  
115    }  
116    }  
117 }

```
106     public Pessoa pessoa(@RequestBody Pessoa p){  
107         return p;  
108     }  
109  
110     @GetMapping("/status")  
111     public ResponseEntity<?> status(){  
112         return new ResponseEntity<?>([Htt]);  
113     }  
114 }  
115  
116 }  
117
```

• o [HttpStatusCode - org.springframework.http](#) >  
↳ [Http - org.springframework.boot.actuate.autoconfigure](#)  
↳ [Http11InputBuffer - org.apache.coyote.http11](#)  
↳ [Http11Nio2Protocol - org.apache.coyote.http11](#)  
↳ [Http11NioProtocol - org.apache.coyote.http11](#)  
↳ [Http11OutputBuffer - org.apache.coyote.http11](#)  
↳ [Http11Processor - org.apache.coyote.http11](#)  
↳ [Http2 - org.springframework.boot.web.server](#)  
↳ [Http2AsyncUpgradeHandler - org.apache.coyote.http2](#)  
↳ [Http2OutputBuffer - org.apache.coyote.http2](#)  
↳ [Http2Protocol - org.apache.coyote.http2](#)

org.springframework.http.HttpStatusCode x

Represents an HTTP response status code.  
Implemented by HttpStatus, but defined as an  
interface to allow for values not in that  
enumeration.

- Since:
  - 6.0
- Author:
  - Arjen Poutsma
- See Also:
  - [HTTP Status Code Registry](#)
  - [List of HTTP status codes - Wikipedia](#)

PROBLEMS 1 OUTPUT TERMINAL

TERMINAL

```
2023-01-24T17:48:26.601-03:00 INFO 8048 --- [nio-8048-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'  
2023-01-24T17:48:26.602-03:00 INFO 8048 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
```

[?] Java Build Status

[?] Run: ApiApplication

```
107     public Pessoa pessoa(@RequestBody Pessoa p){  
108         return p;  
109     }  
110  
111     @GetMapping("/status")  
112     public ResponseEntity<?> status(){  
113         return new ResponseEntity<?>(HttpStatus.ACCEPTED);  
114     }  
115 }  
116  
117 }  
118 }
```

1/5 ResponseEntity(HttpStatusCode status)

- ACCEPTED : HttpStatus
- ALREADY\_REPORTED : HttpStatus
- BAD\_GATEWAY : HttpStatus
- BAD\_REQUEST : HttpStatus
- BANDWIDTH\_LIMIT\_EXCEEDED : HttpStatus
- CHECKPOINT : HttpStatus
- CONFLICT : HttpStatus
- CONTINUE : HttpStatus
- CREATED : HttpStatus
- DESTINATION\_LOCKED : HttpStatus
- EXPECTATION\_FAILED : HttpStatus
- FAILED\_DEPENDENCY : HttpStatus

org.springframework.http.HttpStatus.ACCEPTED : HttpStatus  
202 Accepted .

- See Also:
  - [HTTP/1.1: Semantics and Content, section 6.3.3](#)

PROBLEMS 3 OUTPUT TERMINAL

TERMINAL

```
2023-01-24T17:48:26.601-03:00 INFO 8048 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]  
: Initializing Spring DispatcherServlet 'dispatcherServlet'  
2023-01-24T17:48:26.602-03:00 INFO 8048 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
```

POWERBIAN

Java Build Status

Run: ApiApplication

DEBUG CONSOLE

```
109
110     }
111
112     @GetMapping("/status")
113     public ResponseEntity<?> status(){
114         return new ResponseEntity<?>(HttpStatus.OK);
115     }
116
117 }
```

The code editor shows a Java method `status()` that returns a `ResponseEntity<?>` with the status code `HttpStatus.OK`. A tooltip for `HttpStatus.OK` is open, showing the class definition and various static fields representing HTTP status codes. The `CREATED` field is highlighted.

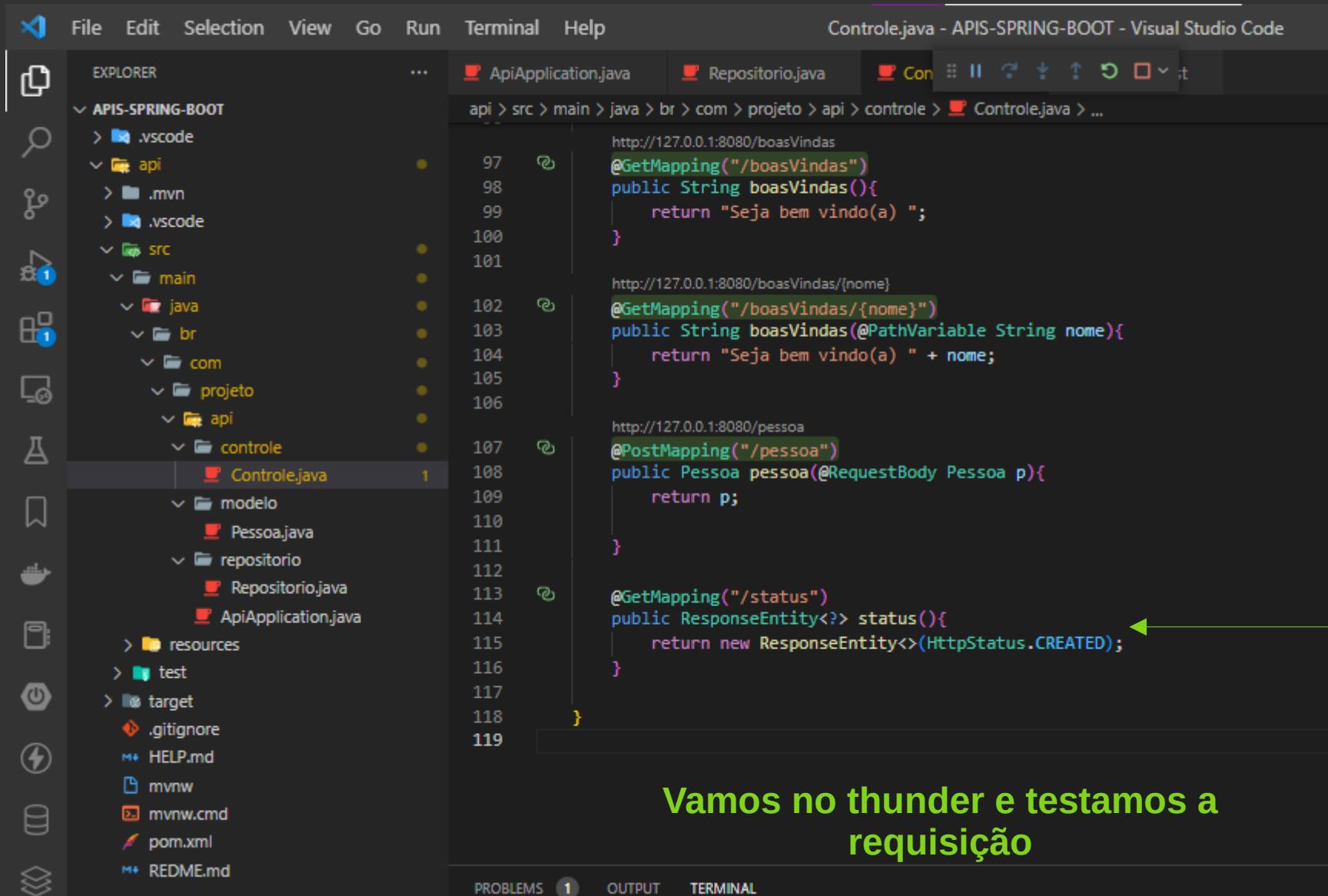
PROBLEMS 1 OUTPUT TERMINAL

TERMINAL

```
2023-01-24T17:48:26.601-03:00 INFO 8048 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-24T17:48:26.602-03:00 INFO 8048 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
```

org.springframework.http.HttpStatus  
CREATED : HttpStatus  
201 Created .  
• See Also:  
◦ HTTP/1.1: Semantic Versioning 6.3.2

DEBUG CON



# Vamos no thunder e testamos a requisição

New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/status

Send

Status: 201 Created Size: 0 Bytes Time: 41 ms

Query Headers 2 Auth Body Tests Pre Run New Response Headers 3 Cookies Results Docs

1

Query Parameters

parameter value

A screenshot of the Thunder Client extension in Visual Studio Code. The interface shows a sidebar with activity logs and a main panel for making API requests. In the main panel, a new request is being prepared for the endpoint 'localhost:8080/status' using a GET method. The response summary indicates a successful '201 Created' status with 0 bytes size and a duration of 41 ms. The 'Response' tab is selected, showing a single item labeled '1'. Below the request section, there's a 'Query Parameters' section with a checkbox and two input fields for 'parameter' and 'value'. The overall theme is dark, and a green arrow points downwards from the top center towards the 'Response' tab.

#29  
**CRIANDO SERVIÇOS**

**O que é o serviço?  
E onde ficam as regras de negocio do projeto**

**Regras de negócios geralmente não devem ficar em rotas**

The screenshot shows the Visual Studio Code interface with a Java Spring Boot project named "APIS-SPRING-BOOT". The project structure in the Explorer sidebar includes ".vscode", "api", "src" (containing "main" and "java"), and "test". The "java" folder contains "br", "com", and "projeto" packages, with "api" being the currently selected folder. In the main editor area, the file "ApiApplication.java" is open, showing imports for java.util.List and org.springframework.web.bind.annotation.\*. A context menu is open over the "api" folder, with the "New Folder..." option highlighted. Other options in the menu include "New File...", "Reveal in File Explorer", "Open in Integrated Terminal", and "Add Folder to Java Source Path".

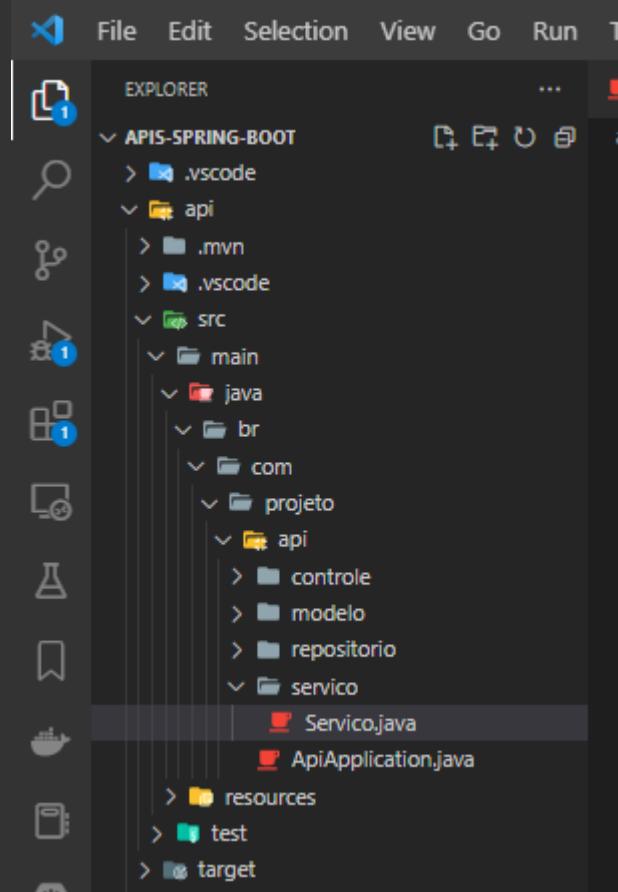
```
2
3     import java.util.List;
4
5     import org.springframework.bea
6     import org.springframework.htt
7     import org.springframework.htt
8     import org.springframework.web
9     import org.springframework.web
10    import org.springframework.web
11    import org.springframework.web
12    import org.springframework.web
13    import org.springframework.web
14    import org.springframework.web
```

The screenshot shows the Visual Studio Code interface with a Java project named "APIS-SPRING-BOOT". The project structure in the Explorer sidebar includes ".vscode", "api", ".mvn", ".vscode", "src", "main", "java", "br", "com", "projeto", "api", "controle", "modelo", "repositorio", and "servico". The "servico" folder is currently selected, as indicated by a green arrow icon in the sidebar.

The code editor displays a file named "ApiApplication.java" with the following content:

```
1 import java.util.List;
2
3 import org.springframework.web.bind.annotation.*;
4
5 import org.springframework.web.bind.annotation.RestController;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestMethod;
8 import org.springframework.web.bind.annotation.RequestParam;
9 import org.springframework.web.bind.annotation.RestController;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RestController;
13 import org.springframework.web.bind.annotation.RequestMapping;
14 import org.springframework.web.bind.annotation.RequestMethod;
15
16 import br.com.projeto.api.servico.Controle;
17 import br.com.projeto.api.servico.Repositorio;
18
```

A context menu is open over the "servico" folder, listing options: "New File...", "New Folder...", "Reveal in File Explorer", "Open in Integrated Terminal", and "Add Folder to Java Source Path".



**Criamos a pasta serviço e dentro dela o arquivo  
Serviço.java**



EXPLORER

## APIS-SPRING-BOOT

&gt; .vscode

&lt; api

&gt; .mvn

&gt; .vscode

&lt; src

&lt; main

&lt; java

&lt; br

&lt; com

&lt; projeto

&lt; api

&gt; controle

&gt; modelo

&gt; repositorio

&lt; servico

&lt; Servico.java

&lt; ApiApplication.java

&gt; resources

&gt; test

&gt; target

&lt; .gitignore

&lt; HELP.md

&lt; mvnw

&lt; mvnw.cmd

&lt; pom.xml

&lt; README.md

...

ApiApplication.java

Repositorio.java

Con



TC New Request

TC New Request

```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Servico
1 package br.com.projeto.api.servico;
2
3 public class Servico {
4
5 }
```

## Para dizer que é um serviço adicionamos a annotation e seu import

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Code Editor:** Displays a Java file with the following code:

```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Servico
1 package br.com.projeto.api.servico;
2
3 @Service
4 public class Servico {
5     ...
6 }
7 }
```
- Code Completion:** A tooltip for the `@Service` annotation is open, listing its possible types:
  - ServerEndpoint - jakarta.websocket.server
  - Service - org.springframework.stereotype
  - ServletComponentScan - org.springframework.boot.w...
  - ServletEndpoint - org.springframework.boot.actuat...
  - ServletSecurity - jakarta.servlet.annotation
  - SecondaryTable - jakarta.persistence
  - SecondaryTables - jakarta.persistence
  - SqlResultSetMapping - jakarta.persistence
  - SqlResultSetMappings - jakarta.persistence
  - WebServlet - jakarta.servlet.annotation
  - JsonSerializableSchema - com.fasterxml.jackson.da...
  - JsonSerialize - com.fasterxml.jackson.databind.an...
- Documentation:** To the right of the completion list, there is a detailed description of the `org.springframework.stereotype.Service` annotation.

**Description of the Service Annotation:**

- Indicates that an annotated class is a "Service", originally defined by Domain-Driven Design (Evans, 2003) as "an operation offered as an interface that stands alone in the model, with no encapsulated state."
- May also indicate that a class is a "Business Service Facade" (in the Core J2EE patterns sense), or something similar. This annotation is a general-purpose stereotype and individual teams may narrow their semantics and use as appropriate.
- This annotation serves as a specialization of `@Component`, allowing for implementation classes to be autodetected through classpath scanning.

**Since:**

- 2.5



EXPLORER

## APIS-SPRING-BOOT

&gt; .vscode

&gt; api

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

&gt; modelo

&gt; repositorio

&gt; servico

&gt; Servico.java

&gt; ApiApplication.java

&gt; resources

&gt; test

&gt; target

&gt; .gitignore

&gt; HELP.md

&gt; mvnw

&gt; mvnw.cmd

&gt; pom.xml

&gt; README.md

...

ApiApplication.java

Repositorio.java

Con

II

?

↓

↑

○

□

▼

×

x

New Request

New Request

```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java(TM) by Red Hat > Servico.java
1      package br.com.projeto.api.servico;
2
3      import org.springframework.stereotype.Service; ←
4      ^
5      @Service ←
6      public class Servico {
7
8      }
9
```

#30  
**Annotation Component**

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left displays the project structure:

- APIS-SPRING-BOOT
- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle
- modelo
- repositorio
- servico
- Serviço
- ApiAppli
- resources

The code editor on the right shows `ApiApplication.java` with the following content:

```
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @Service
7 public class Serviço {
8 }
9
```

A context menu is open over the "servico" folder in the Explorer sidebar, with the "New File..." option highlighted.

- New File...
- New Folder...
- Reveal in File Explorer Shift+Alt+R
- Open in Integrated Terminal

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Mensagem.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - Root: .vscode, api, .mvn, .vscode, src.
  - src: main, java, br, com, projeto, api, controle, modelo.
  - modelo: Mensagem.java, Pessoa.java.
  - servico: Servico.java.
  - resources.
  - test.
  - target: .gitignore, HELP.md, mvnw, mvnw.cmd, pom.xml, README.md.
- Code Editor:** Displays the content of Mensagem.java.

```
1 package br.com.projeto.api.modelo;
2
3 public class Mensagem {
4
5 }
```

A code completion dropdown is open at the end of the class definition, listing various suggestions:
  - class Mensagem
  - interface Mensagem
  - enum Mensagem
  - record Mensagem()
  - abstract class Mensagem
  - @interface Mensagem
- Terminal:** At the bottom of the interface.

**Dentro da pasta modelo criamos o arquivo Mensagem.java**

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Mensagem.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
- Code Editor:** Displays the following Java code:

```
1 package br.com.projeto.api.modelo;
2
3 @Com
4 public class Mensagem {
5     ...
6 }
7 }
```
- IntelliSense Tooltip:** A tooltip for the `@Component` annotation is displayed, providing the following information:
  - Indicates that an annotated class is a "component". Such classes are considered as candidates for auto-detection when using annotation-based configuration and classpath scanning.
  - Other class-level annotations may be considered as identifying a component as well, typically a special kind of component e.g. the `@Repository` annotation or AspectJ's `@Aspect` annotation.
- Status Bar:** Shows "Mensagem.java 1" and "New Request".

Adicionamos a anotation e seu import

**Inicialmente o annotation @Componente faz uma busca por tudo que esta anotado nessa classe e depois usamos o @Autowired que nossa injeção de dependencia para instanciar um objeto dessa classe**

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Mensagem.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including .vscode, api, .mvn, .vscode, src, main, java, br, com, projeto, api, controle, and modelo folders.
- Code Editor:** Displays Mensagem.java with the following code:

```
1 package br.com.projeto.api.modelo;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class Mensagem {
7
8     private String mensagem; ←
9
10 }
11
```
- Bottom Left:** Shows a list of files: Mensagem.java, Pessoa.java, repositorio, servico, Servico.java, ApiApplication.java, resources, and test.

**Selecione a linha toda e click na  
lampada pra gerar os getters e setters**

```
4
5  @Component
6  public class Mensagem {
7
8      private String mensagem;
9
10 }
11
```

Quick Fix...

Remove 'mensagem', keep assignments with side effects

More Actions...

Add Javadoc for 'mensagem'

**Generate Getter and Setter for 'mensagem'**

Generate Getter for 'mensagem'

Generate Setter for 'mensagem'

Generate Constructors...

Add final modifier for 'mensagem'



EXPLORER

...

ApiApplication.java

Repositorio.java

Con

II

?

▼

▲

○

●

Mensagem.java

APIS-SPRING-BOOT

&gt; .vscode

&lt; api

&gt; .mvn

&gt; .vscode

&lt; src

&lt; main

&lt; java

&lt; br

&lt; com

&lt; projeto

&lt; api

&gt; controle

&lt; modelo

Mensagem.java

Pessoa.java

&gt; repositorio

&lt; servico

Servico.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; modelo &gt; Mensagem.java &gt; ...

```
1 package br.com.projeto.api.modelo;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class Mensagem {
7
8     private String mensagem;
9
10    public String getMensagem() {
11        return mensagem;
12    }
13
14    public void setMensagem(String mensagem) {
15        this.mensagem = mensagem;
16    }
17
18 }
19 }
```

salve

**Depois de salvo você pode ir na classe de serviço  
E la podemos criar o @Autowired pra automaticamente instanciar objetos**



EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

&gt; api

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

&gt; modelo

&gt; Mensagem.java

&gt; Pessoa.java

&gt; repository

&gt; servico

&gt; Servico.java

&gt; ApiApplication.java

&gt; resources

&gt; test

&gt; target

&gt; .gitignore

&gt; HELP.md

&gt; mvnw

&gt; mvnw.cmd

&gt; pom.xml

&gt; README.md

ApiApplication.java

Repositorio.java

Con

Mensagem.java

New Request

New Request

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; servico &gt; Servico.java &gt; Language Support for Java(TM) by Red Hat &gt; Servico

1 package br.com.projeto.api.servico;

2

3 import org.springframework.stereotype.Service;

4

5 @Service

6 public class Servico {

7

8 @Auu

9 } 10 }

11

- **AutoConfiguration** - org.springframework.boot.autoconfigure
- **AutoConfigurationPackage** - org.springframework.boot.autoconfigure
- **AutoConfigureAfter** - org.springframework.boot.autoconfigure
- **AutoConfigureBefore** - org.springframework.boot.autoconfigure
- **AutoConfigureOrder** - org.springframework.boot.autoconfigure
- **Autowired** - org.springframework.beans.factory.annotation.Autowired
- **AttributeAccessor** - org.hibernate.annotations
- **AttributeBinderType** - org.hibernate.annotations
- **AttributeOverride** - jakarta.persistence
- **AttributeOverrides** - jakarta.persistence
- **NoAutoStart** - ch.qos.logback.core.joran.spi
- **JsonAutoDetect** - com.fasterxml.jackson.annotation

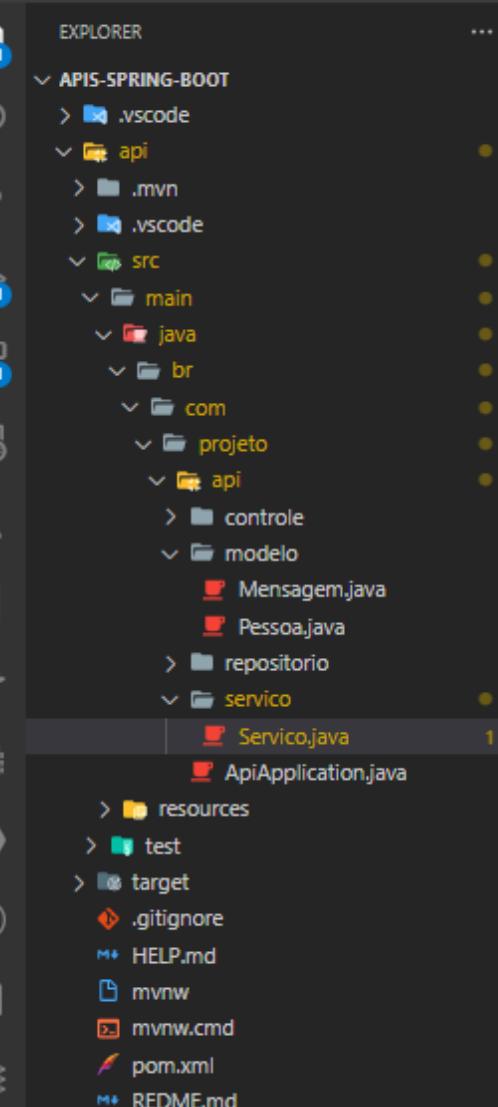
org.springframework.beans.factory.annotation.Autowired

Marks a constructor, field, setter method, or config method as to be autowired by Spring's dependency injection facilities. This is an alternative to the JSR-330 jakarta.inject.Inject annotation, adding required-vs-optional semantics.

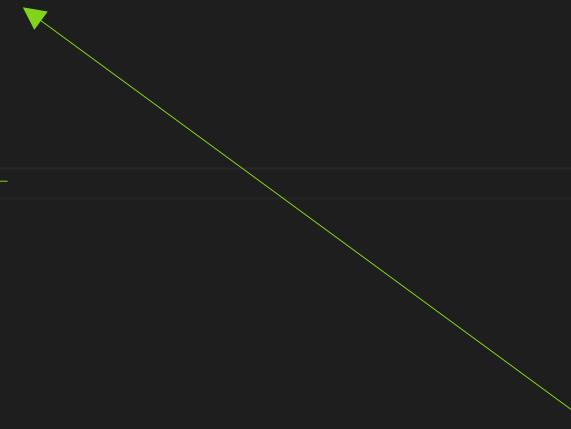
### Autowired Constructors

Only one constructor of any given bean class may declare this annotation with the required attribute set to `true`, indicating the constructor to autowire when used as a Spring bean. Furthermore, if the required attribute is set to `true`, only a single constructor may be annotated with `@Autowired`. If multiple non-required constructors declare the annotation, they will be considered as candidates for autowiring. The constructor with the greatest number of dependencies that can be satisfied by matching beans in the Spring container will be chosen. If none of the candidates can be satisfied, then a primary/default constructor (if present) will

Adicione a notation e seu  
import



```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java(TM) by Red Hat > Servico > mensagem.java  
1 package br.com.projeto.api.servico;  
2  
3 import org.springframework.beans.factory.annotation.Autowired;  
4 import org.springframework.stereotype.Service;  
5  
6 import br.com.projeto.api.modelo.Mensagem;  
7  
8 @Service  
9 public class Servico {  
10  
11     @Autowired  
12     private Mensagem mensagem; // Error here  
13 }  
14  
15
```



Criamos um objeto de nome mensagem não  
esqueça de conferir os imports

**Agora toda vez que precisarmos passar uma mensagem só passamos esse objeto  
não precisaremos instanciar um objeto**

#31  
**IMPLEMENTANDO SERVIÇOS**

**Vamos cadastrar e validar**

File Edit Selection View Go Run Terminal Help

• Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- < controle
- < modelo
  - Mensagem.java
  - Pessoa.java
- < repositorio
  - Repository.java
- < servico
  - Servico.java
  - ApiApplication.java

Servico.java 1

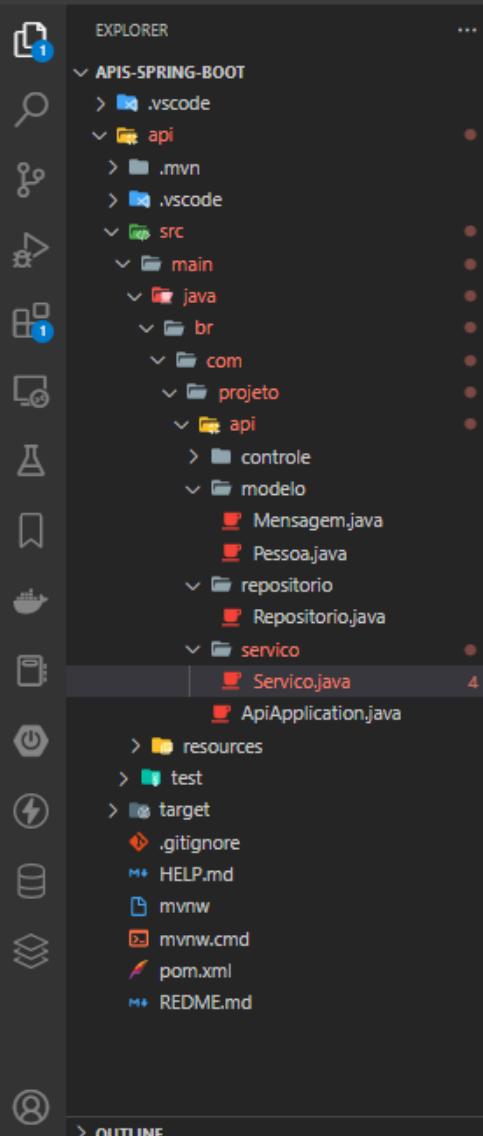
```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java
1 package br.com.projeto.api.servico;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Service;
5
6 import br.com.projeto.api.modelo.Mensagem;
7
8 @Service
9 public class Servico {
10
11     @Autowired
12     private Mensagem mensagem;
13
14     @Autowired
15     private Repository acao;
16
17 }
18
```

PROBLEMS 1 OUTPUT TERMINAL

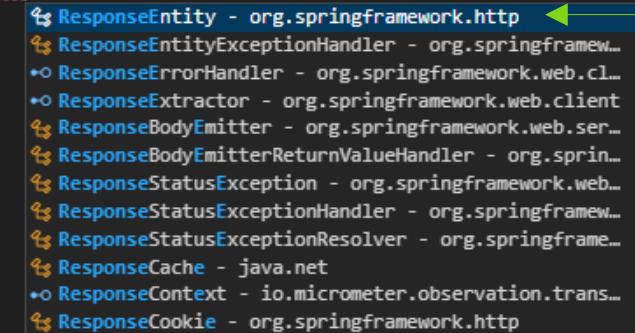
• Servico.java api\src\main\java\br\com\projeto\api\servico 1

Repository cannot be resolved to a type Java(16777218) [Ln 15, Col 13]

Como boa pratica vamos deixar o CRUD no servico



```
api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java(TM) by Red Hat > ☰ Servico > acao
 1 package br.com.projeto.api.servico;
 2
 3 import org.springframework.beans.factory.annotation.Autowired;
 4 import org.springframework.stereotype.Service;
 5
 6 import br.com.projeto.api.modelo.Mensagem;
 7
 8 @Service
 9 public class Servico {
10
11     @Autowired
12     private Mensagem mensagem;
13
14     @Autowired
15     private Repositorio acao;
16
17     public ResponseEntity<Mensagem> acao(ResponseEntity<Mensagem> responseEntity) {
18         return responseEntity;
19     }
20 }
```



PROBLEMS 4 OUTPUT TERMINAL

Service.java api\src\main\java\br\com\projeto\api\service

Filter (e.g. text: \*\*/

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
              - modelo
                - Mensagem.java
                - Pessoa.java
              - repositorio
                - Repositorio.java
              - servico
                - Servico.java
  - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Servico.java 1

api > src > main > java > br > com > projeto > api > servico > Servico.java > ...

```
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.stereotype.Service;
6
7 import br.com.projeto.api.modelo.Mensagem;
8
9 @Service
10 public class Servico {
11
12     @Autowired
13     private Mensagem mensagem;
14
15     @Autowired
16     private Repositorio acao;
17
18     public ResponseEntity<?> cadastrar(){
19
20
21
22 }
23 }
```

PROBLEMS 1 OUTPUT TERMINAL

• Servico.java api\src\main\java\br\com\projeto\api\servico 1

File Edit Selection View Go Run Terminal Help • Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- > controle
- < modelo
- < Mensagem.java
- < Pessoa.java
- < repositorio
- < Repositorio.java
- < servico
- < Servico.java

Servico.java 1

api > src > main > java > br > com > projeto > api > servico > Servico.java > ...

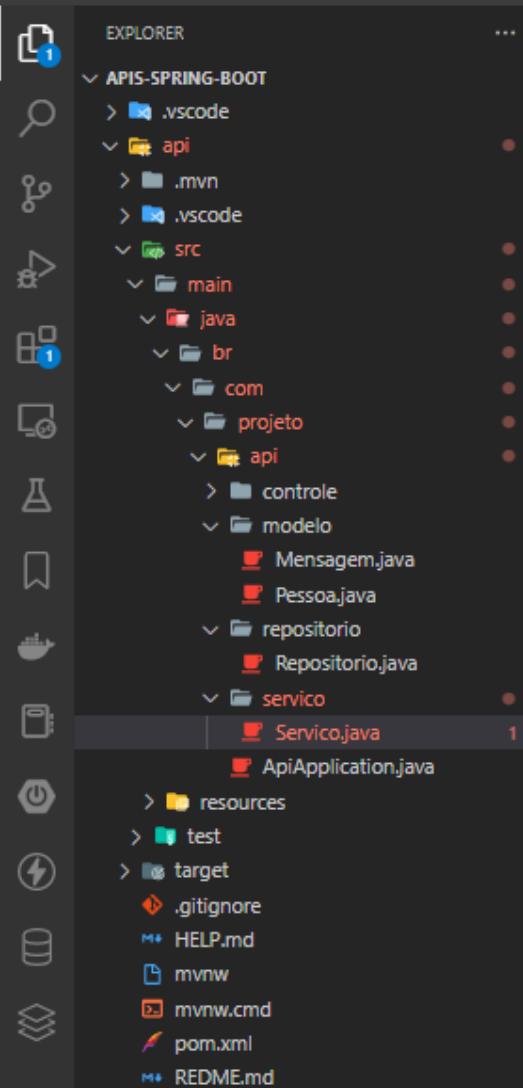
```
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.stereotype.Service;
6
7 import br.com.projeto.api.modelo.Mensagem;
8 import br.com.projeto.api.modelo.Pessoa;
9
10 @Service
11 public class Servico {
12
13     @Autowired
14     private Mensagem mensagem;
15
16     @Autowired
17     private Repositorio acao;
18
19     public ResponseEntity<?> cadastrar(Pessoa obj){
20
21     }
22
23 }
24 }
```

PROBLEMS 1 OUTPUT TERMINAL

• Servico.java api\src\main\java\br\com\projeto\api\servico 1

Repository cannot be resolved to a type Java(16777218) [Ln 17, Col 13]

Temos que ter um objeto do tipo pessoa pra cadastrar



```
 3 import org.springframework.beans.factory.annotation.Autowired;
 4 import org.springframework.http.ResponseEntity;
 5 import org.springframework.stereotype.Service;
 6
 7 import br.com.projeto.api.modelo.Mensagem;
 8 import br.com.projeto.api.modelo.Pessoa;
 9
10 @Service
11 public class Servico {
12
13     @Autowired
14     private Mensagem mensagem;
15
16     @Autowired
17     private Repositorio acao;
18
19     public ResponseEntity<?> cadastrar(Pessoa obj){
20
21         if(obj.getNome().equals("")){ ←
22
23     }
24
25 }
26
27 }
28 }
```

Estamos verificando se o nome  
informado é vazio se for vazio

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
    - br
      - com
        - projeto
        - api
          - controle
          - modelo
            - Mensagem.java
            - Pessoa.java
          - repositorio
            - Repository.java
          - servico
            - Servico.java
            - ApiApplication.java

PROBLEMS 5

OUTPUT TERMINAL

Filter (e.g. text, \*\*/\*.ts, !\*\*/node)

OUTLINE

TIMELINE

JAVA PROJECTS

Servico.java 3

```
1 package br.com.projeto.api.servico;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.HttpStatus;
5 import org.springframework.http.ResponseEntity;
6 import org.springframework.stereotype.Service;
7
8 import br.com.projeto.api.modelo.Mensagem;
9 import br.com.projeto.api.modelo.Pessoa;
10
11 @Service
12 public class Servico {
13
14     @Autowired
15     private Mensagem mensagem;
16
17     @Autowired
18     private Repository acao;
19
20     public ResponseEntity<?> cadastrar(Pessoa obj){
21
22         if(obj.getNome().equals("")){
23             mensagem.setMensagem("O nome precisa ser preenchido");
24             return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
25         }else if(obj.getIdade() < 0){
26             mensagem.setMensagem("Informe uma idade valida");
27             return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
28         }
29     }
30 }
```

- ✖ The project was not built since its build path is incomplete. Cannot find the class file for java.util.Optional. Fix the build path then try building this project Java(0) [Ln 1, Col 1]
- ✖ Controle.java api\src\main\java\br\com\projeto\api\controle 1
- ✖ The type java.util.Optional cannot be resolved. It is indirectly referenced from required type java.lang.Class Java(16777563) [Ln 32, Col 16]
- ✖ Servico.java api\src\main\java\br\com\projeto\api\servico 3
- ✖ The type java.util.Optional cannot be resolved. It is indirectly referenced from required type org.springframework.http.ResponseEntity Java(16777563) [Ln 1, Col 1]
- ✖ Repository cannot be resolved to a type Java(16777218) [Ln 18, Col 13]
- ✖ The operator < is undefined for the argument type(s) String, int Java(536871072) [Ln 25, Col 18]



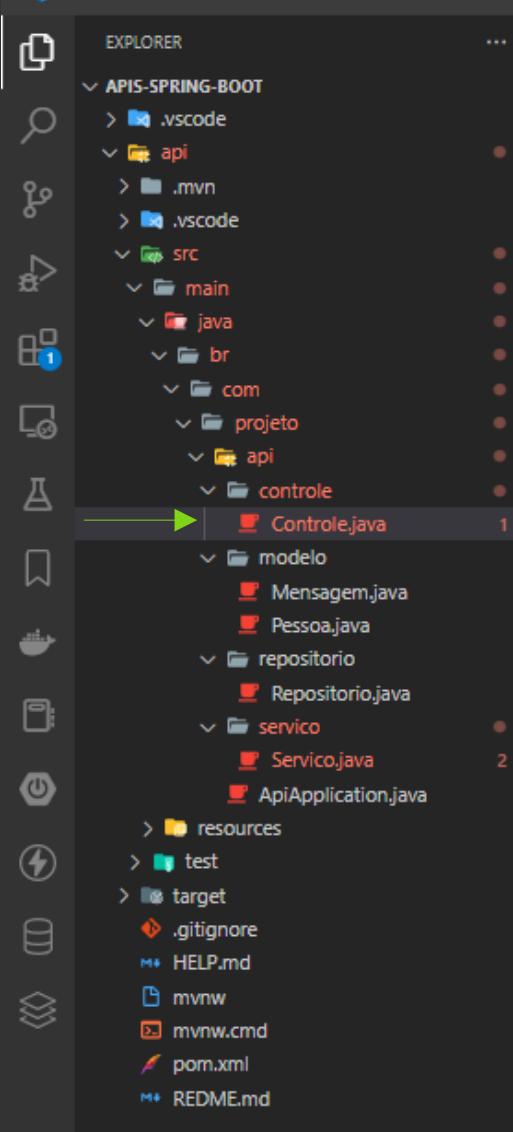
EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
              - modelo
                - Mensagem.java
                - Pessoa.java
              - repositorio
                - Repositorio.java
              - servico
                - Servico.java
    - ApiApplication.java
  - resources
  - test
  - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md

Servico.java 2 x

```
api > src > main > java > br > com > projeto > api > servico > Servico.java > ...
  0   import org.springframework.stereotype.Service;
  1
  2   import br.com.projeto.api.modelo.Mensagem;
  3   import br.com.projeto.api.modelo.Pessoa;
  4   import br.com.projeto.api.repositorio.Repositorio;
  5
  6   @Service
  7   public class Servico {
  8
  9       @Autowired
 10      private Mensagem mensagem;
 11
 12      @Autowired
 13      private Repositorio acao;
 14
 15
 16      public ResponseEntity<?> cadastrar(Pessoa obj){
 17
 18          if(obj.getNome().equals("")){
 19              mensagem.setMensagem("O nome precisa ser preenchido");
 20              return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 21          }else if(obj.getIdade() < 0){
 22              mensagem.setMensagem("Informe uma idade valida");
 23              return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 24          }else{
 25              return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);
 26          }
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36 }
```



```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle >
```

```
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.DeleteMapping;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.PostMapping;
12 import org.springframework.web.bind.annotation.PutMapping;
13 import org.springframework.web.bind.annotation.RequestBody;
14 import org.springframework.web.bind.annotation.RestController;
```

```
15
16 import br.com.projeto.api.modelo.Pessoa;
17 import br.com.projeto.api.repositorio.Repositorio;
18 import br.com.projeto.api.servico.Servico;
```

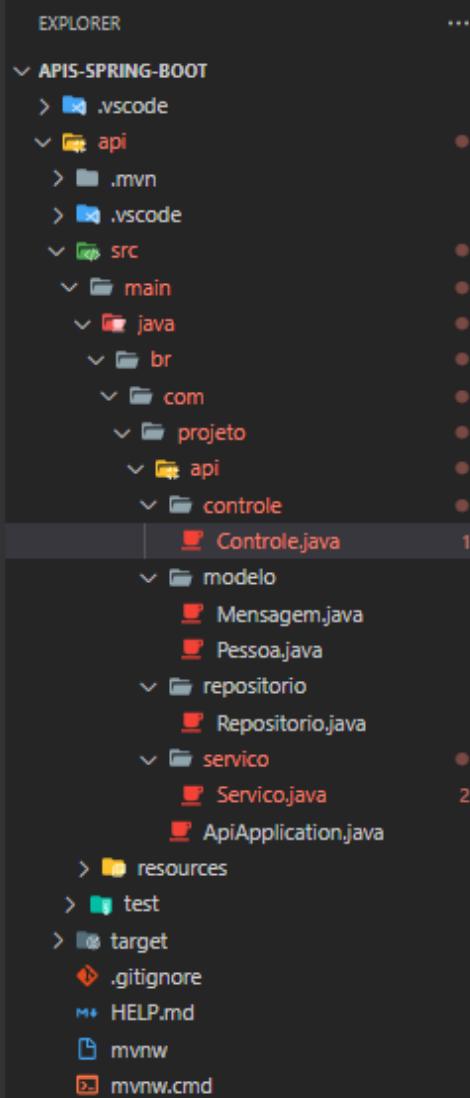
```
19
20 @RestController
21 public class Controle {
22
23     @Autowired
24     private Repositorio acao;
25
26     @Autowired
27     private Servico servico; ◀
28
29     @PostMapping("/api")
30     public Pessoa cadastrar(@RequestBody Pessoa obj){
31         return acao.save(obj);
32     }
33 }
```

PROBLEMS 4 OUTPUT TERMINAL

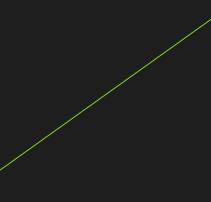
api 1

The project was not built since its build path is incomplete. Cannot find the class file for java.util.Optional. Fix the build path then try building.

Controle.java api\src\main\java\br\com\projeto\api\controle 1



```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle.java  
17     import br.com.projeto.api.repositorio.Repositorio;  
18     import br.com.projeto.api.servico.Servico;  
19  
20     @RestController  
21     public class Controle {  
22         @Autowired  
23         private Repositorio acao;  
24  
25         @Autowired  
26         private Servico servico;  
27  
28         @PostMapping("/api")  
29         public Pessoa cadastrar(@RequestBody Pessoa obj){  
30             return acao.save(obj);  
31         }  
32  
33         @GetMapping("/api")  
34         public List<Pessoa> selecionar(){  
35             return acao.findAll();  
36         }  
37  
38         @GetMapping("/api/{codigo}")  
39         public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
40             return acao.findByCodigo(codigo);  
41         }  
42  
43         @PutMapping("/api")  
44         public Pessoa editar(@RequestBody Pessoa obj){  
45             return acao.save(obj);  
46         }
```



Vamos mudar essa rota

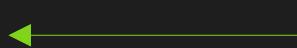


EXPLORER  
APIS-SPRING-BOOT  
.vscode  
api  
.mvn  
.vscode  
src  
main  
java  
br  
com  
projeto  
api  
controle  
Controle.java  
1  
modelo  
Mensagem.java  
Pessoa.java  
repositorio  
Repositorio.java  
servico  
Servico.java  
2  
ApiApplication.java  
resources  
test  
target  
.gitignore  
HELP.md  
mvnw  
mvnw.cmd  
pom.xml  
REDME.md

Servico.java 2 Controle.java 1

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle

```
17     import br.com.projeto.api.repositorio.Repositorio;
18     import br.com.projeto.api.servico.Servico;
19
20     @RestController
21     public class Controle {
22
23         @Autowired
24         private Repositorio acao;
25
26         @Autowired
27         private Servico servico;
28
29         @PostMapping("/api")
30         public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){
31             return servico.cadastrar(obj);
32         }
33
34         @GetMapping("/api")
35         public List<Pessoa> selecionar(){
36             return acao.findAll();
37         }
38
39         @GetMapping("/api/{codigo}")
40         public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
41             return acao.findByCodigo(codigo);
42         }
43
44         @PutMapping("/api")
45         public Pessoa editar(@RequestBody Pessoa obj){
46             return acao.save(obj);
```



PROBLEMS 4 OUTPUT TERMINAL

api 1

The project was not built since its build path is incomplete. Cannot find the class file for java.util.Optional. Fix the build path then try building this

## **Resolvendo o erro**

File Edit Selection View Go Run Terminal Help Servico.java - APIS-SPRING-BOOT - Visual Studio Code

APPS api

SERVICO DASHBOARD

Servico.java 1 x Controle.java

api > src > main > java > br > com > projeto > api > servico > Servico.java Language Support for Java(TM) by Red Hat > Servico > cadastrar(Pessoa)

```
import br.com.projeto.api.modelo.Pessoa;
import br.com.projeto.api.repositorio.Repositorio;

@Service
public class Servico {

    @Autowired
    private Mensagem mensagem;

    @Autowired
    private Repositorio acao;

    public ResponseEntity<?> cadastrar(Pessoa obj){

        if(obj.getNome().equals("")){
            mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");
            return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
        }else if(obj.getIdade() < 0){
            mensagem.setMensagem(mensagem: "Informe uma idade valida");
            return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
        }else{
            return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);
        }
    }
}
```

Apenas fechei e reabri o vscode

PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

TERMINAL

```
2023-01-25T12:47:12.072-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-25T12:47:12.078-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet
: Initializing Servlet 'dispatcherServlet'
2023-01-25T12:47:12.111-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet
: Completed initialization in 3 ms
```

DEBUG CONSOLE

Debug: ApiApplication  
Run: ApiApplication

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Ln 26, Col 30 (8 selected) Spaces: 4 UTF-8 CRLF { Java Go Live

29°C Nublado

File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

Spring Boot Dashboard

APPS api

BEANS

ENDPOINT MAPPINGS

MEMORY

Heap Memory  
Size 62 MB / Used 40 MB / Max 2122 MB  
MB 0 1247.56 1248.01 1248.06 1248.11 1248.16 1248.21 1248.26 1248.31 1248.36 1248.41

Unknown (pid: 8972)

Servico.java 1 Controle.java X

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > selecionar()

```
1 public class Controle {  
2     -- Repositorio  
3     @Autowired  
4     private Repositorio acao;  
5     -- Servico  
6     @Autowired  
7     private Servico servico;  
8     http://127.0.0.1:8080/api  
9     @PostMapping("/api")  
10    public ResponseEntity<Pessoa> cadastrar(@RequestBody Pessoa obj){  
11        return servico.cadastrar(obj);  
12    }  
13     http://127.0.0.1:8080/api  
14     @GetMapping("/api")  
15    public List<Pessoa> selecionar(){  
16        return acao.findAll();  
17    }  
18     http://127.0.0.1:8080/api/{codigo}  
19     @GetMapping("/api/{codigo}")  
20    public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
21        return acao.findByCodigo(codigo);  
22    }  
23     http://127.0.0.1:8080/api  
24     @PutMapping("/api")  
25    public Pessoa editar(@RequestBody Pessoa obj){  
26        return acao.save(obj);  
27    }
```

PROBLEMS 1 OUTPUT TERMINAL

TERMINAL

```
2023-01-25T12:47:12.072-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/]  
: Initializing Spring DispatcherServlet 'dispatcherServlet'  
2023-01-25T12:47:12.078-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet  
: Initializing Servlet 'dispatcherServlet'  
2023-01-25T12:47:12.111-03:00 INFO 8972 --- [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet  
: Completed initialization in 3 ms
```

DEBUG CONSOLE

Debug: ApiApplication  
Run: ApiApplication

Ln 36, Col 30 (14 selected) Spaces: 4 UTF-8 CRLF (8 Java Go Live

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Windows Taskbar Icons

29°C Nublado 12:48

Único erro que aparece no terminal

The screenshot shows the VS Code interface with the Problems panel open. The Problems panel has a dark theme and displays one error. The error message is: "Servico.java api\src\main\java\br\com\projeto\api\servico 1: getidade cannot be resolved or is not a field Java(33554502) [Ln 26, Col 22]". The code editor above shows a snippet of Java code:

```
44     @
45     @PutMapping("/api")
46     public Pessoa editar(@RequestBody Pessoa obj){
        return acao.save(obj);
```

Vamos testar pra ver se vai dar certo

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/status  
19 hours ago

POST localhost:8080/pessoa  
22 hours ago

DEL localhost:8080/api/2  
2 days ago

PUT localhost:8080/api  
2 days ago

GET localhost:8080/api  
2 days ago

POST localhost:8080/api  
2 days ago

POST localhost:8080  
3 days ago

Query Headers 2 Auth Body Tests Pre Run New

GET https://www.thunderclient.com/welcome

Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Parameters

parameter value

Send Request Ctrl Enter

Import Curl Ctrl U

Change Environment Ctrl E

Git Sync Details

The screenshot shows the Thunder Client extension in Visual Studio Code. The interface includes a left sidebar with activity logs, a top bar with file operations like File, Edit, Selection, View, Go, Run, Terminal, and Help, and a title bar indicating "New Request - APIS-SPRING-BOOT - Visual Studio Code". The main area is titled "New Request" and contains a request configuration panel. The request method is set to "POST" and the URL is "http://localhost:8080/api". The "Body" tab is selected, showing JSON content:

```
1 {  
2   "nome": "",  
3   "idade": 22  
4 }
```

Below the body, there are tabs for Query, Headers, Auth, Body, Tests, Pre Run, and New. To the right, there are sections for Response, Headers, Cookies, Results, and Docs. A "Send" button is located at the top right of the request panel. On the far right, there are several keyboard shortcuts: Send Request (Ctrl + Enter), Import Curl (Ctrl + U), Change Environment (Ctrl + E), and Git Sync (Details). A large green arrow points upwards from the JSON code towards the "Send" button.

Criamos um objeto json e enviamos olhe os dados acima

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java 1 Controle.java New Requests ...

New Request Activity Collections Env filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "",  
3   "idade": 22  
4 }
```

Esta dando esse erro  
jogaremos no google pra  
saber o que é o erro 500

Status: 500 Internal Server Error Size: 5.63 KB Time: 226 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "timestamp": "2023-01-25T15:53:41.608+00:00",  
3   "status": 500,  
4   "error": "Internal Server Error",  
5   "trace": "java.lang.Error: Unresolved compilation problem: \n\tgetIdade  
cannot be resolved or is not a field\n\r\n\tat br.com.projeto.api  
.servico.Servico.cadastrar(Servico.java:26)\r\n\tat br.com.projeto  
.api.controle.Controle.cadastrar(Controle.java:31)\r\n\tat java.base  
/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
\r\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl  
.invoke(Unknown Source)\r\n\tat java.base/jdk.internal.reflect  
.DelegatingMethodAccessorImpl.invoke(Unknown Source)\r\n\tat java  
.base/java.lang.reflect.Method.invoke(Unknown Source)\r\n\tat org  
.springframework.web.method.support.InvocableHandlerMethod.  
doInvoke(InvocableHandlerMethod.java:207)\r\n\tat org.springframework.web  
.method.support.InvocableHandlerMethod.invokeForRequest  
(InvocableHandlerMethod.java:152)\r\n\tat org.springframework.web  
.servlet.mvc.method.annotation.ServletInvocableHandlerMethod  
.invokeAndHandle(ServletInvocableHandlerMethod.java:117)\r\n\tat org  
.springframework.web.servlet.mvc.method.annotation  
.RequestMappingHandlerAdapter.invokeHandlerMethod  
(RequestMappingHandlerAdapter.java:884)\r\n\tat org.springframework  
.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter  
.handleInternal(RequestMappingHandlerAdapter.java:797)\r\n\tat org  
.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter  
.handle(AbstractHandlerMethodAdapter.java:87)\r\n\tat org  
.springframework.web.servlet.DispatcherServlet.doDispatch  
()
```

PROBLEMS 1 OUTPUT TERMINAL

Service.java api\src\main\java\br\com\projeto\api\servico 1  
getidade cannot be resolved or is not a field Java(33554502) [Ln 26, Col 22]

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

1 0 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

29°C Nublado ⚡ ☰ 🔍 12:53

[←](#)[→](#)[C](#)[google.com/search?q=gooel+erro+500&oq=gooel+erro+500&aq=chrome..69i57j0i13i512j0i22i30l8.3672j0j9&sourceid=chrome&ie=UTF-8](#)

Aproximadamente 19.300.000 resultados (0,55 segundos)

Exibindo resultados para [google erro 500](#)

Em vez disso, pesquisar por [gooel erro 500](#)

**Erro 500 é um código apresentado em sites para indicar problemas com a estrutura da página que o usuário deseja acessar.** Em outras palavras, não se trata de um bug no navegador do usuário, tampouco com sua conexão de Internet. há 2 dias



<https://www.techtudo.com.br/noticias/2023/01/google-erro-500-causa-problemas-sites>

[Google Ads fora do ar: site caiu e apresenta 'erro 500'](#)

[?](#) Sobre trechos em destaque • [Feedback](#)

As pessoas também perguntam

O que é erro 500 no Google?



**O erro 500 é um status de erro HTTP que indica uma dificuldade de processamento do servidor, a partir de uma incompatibilidade ou configuração incorreta.**



<https://king.host/artigo/erro-500-causas-e-correcoes>

[O que está gerando erro 500 em seu site - KingHost](#)

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Servico.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle (Controle.java)
                - modelo (Mensagem.java, Pessoa.java)
                - repositorio (Repositorio.java)
    - resources
  - Code Editor:** The current file is "Servico.java". The code is as follows:

```
13     public class Servico {
14
15         -- Mensagem
16         @Autowired
17         private Mensagem mensagem;
18
19         -- Repositorio
20         @Autowired
21         private Repositorio acao;
22
23         public ResponseEntity<?> cadastrar(Pessoa obj){
24
25             if(obj.getNome().equals("")){
26                 mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");
27                 return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
28             }else if(Integer.parseInt(obj.getIdade()) < 0){
29                 mensagem.setMensagem(mensagem: "Informe uma idade valida");
30                 return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
31             }else{
32                 return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);
33             }
34         }
35     }
```

Problema resolvido usando parseInt()



Cerca de

Produtos

Para equipes

Procurar...

Conecte-se

Inscrever-se

Casa

PÚBLICO

Perguntas

1 Tag

Usuários

Empresas

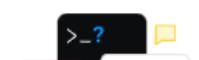
COLETIVOS

Explorar Coletivos

EQUIPES

**Stack Overflow for**

**Teams** – Comece a colaborar e compartilhar conhecimento organizacional.



**Junte-se ao Stack Overflow** para encontrar a melhor resposta para sua pergunta técnica, ajude outras pessoas a responderem às suas.

um crédito de USD200



Report this ad

Você precisa analisar a String como um inteiro.

1

```
String age = input.nextLine();
if (Integer.parseInt(age) < 50) {
    System.out.println("You are young");
}
```

Compartilhar

Seguir

respondido em 29 de maio de 2015 às 18h17



David M

841 ▪ 6 ▪ 23

Adicione um comentário

0

```
Scanner input = new Scanner(System.in);
int age = input.nextInt();
// Comparisons
```

A classe scanner na verdade tem um método de entrada chamado nextInt() que você pode usar:

Inscra-se com e-mail

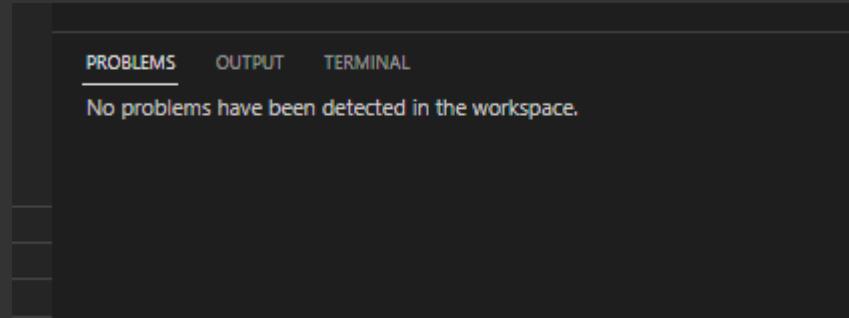
Inscra-se com o Google

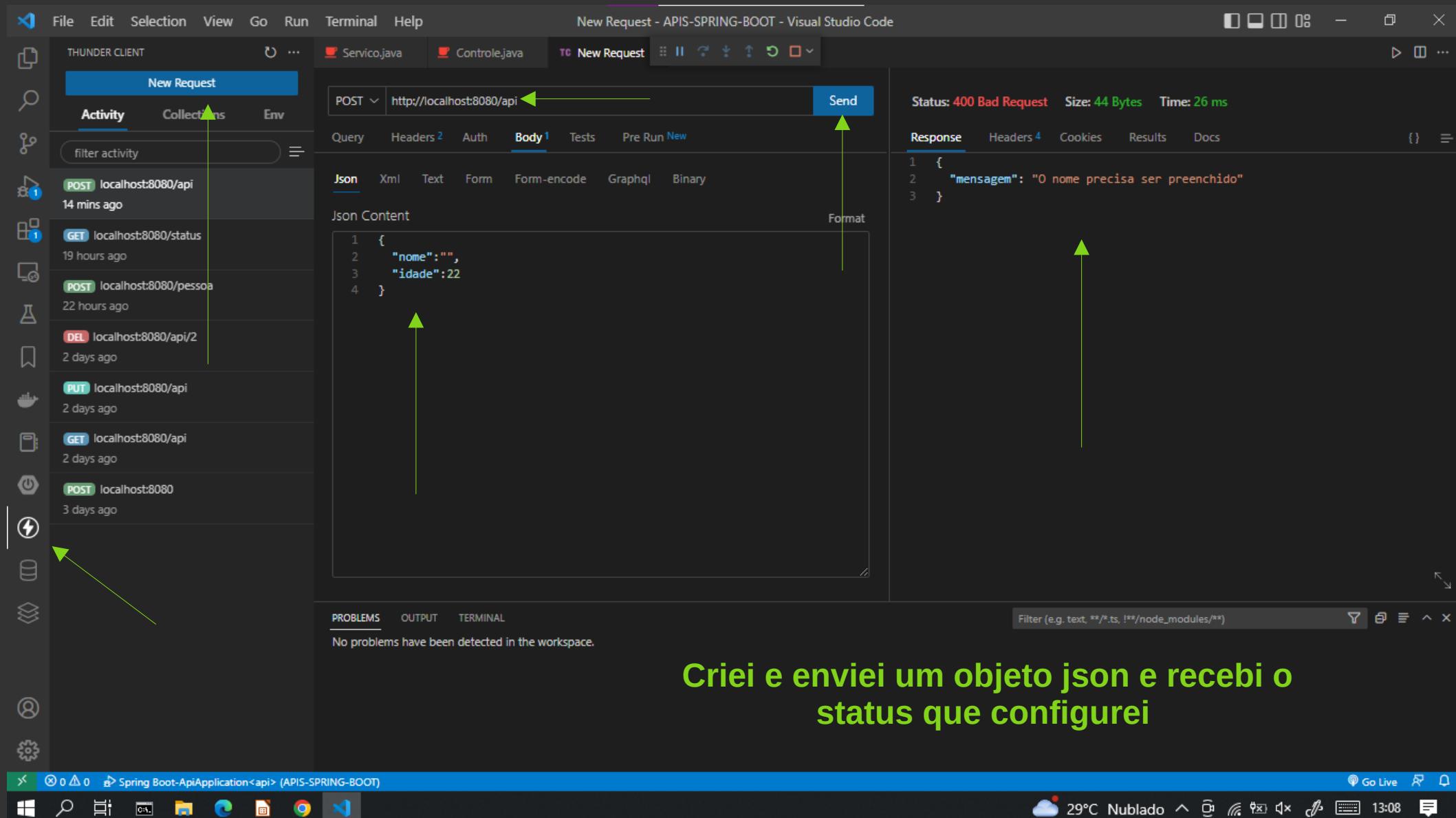
Inscra-se no GitHub

Inscra-se com o Facebook

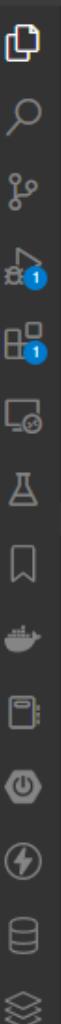
X

## Nenhum problema no terminal





**Criei e enviei um objeto json e recebi o status que configurei**



EXPLORER

- APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - proj
              - api
              - c
              - m
              - r
              - s
              - se
              - A
        - resource
      - test
    - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md

```
Controle.java
public class Controle {
    private Repositorio acao;
    private Servico servico;
    @PostMapping("/api")
    public ResponseEntity<Pessoa> cadastrar(@RequestBody Pessoa obj){
        return servico.cadastrar(obj);
    }
    @GetMapping("/api")
    public List<Pessoa> selecionar(){
        return acao.findAll();
    }
    @GetMapping("/api/{codigo}")
    public Pessoa selecionarPeloCodigo(@PathVariable int codigo){
        return acao.findByCodigo(codigo);
    }
    @PutMapping("/api")
    public Pessoa editar(@RequestBody Pessoa obj){
        return acao.save(obj);
    }
}
```

# Só pra mostrar como esta

## PROBLEMS      OUTPUT      TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules)

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Servico.java Controle.java New Request

New Request Activity Collections Env

filter activity

POST localhost:8080/api 18 mins ago

GET localhost:8080/status 19 hours ago

POST localhost:8080/pessoa 23 hours ago

DEL localhost:8080/api/2 2 days ago

PUT localhost:8080/api 2 days ago

GET localhost:8080/api 2 days ago

POST localhost:8080 3 days ago

POST http://localhost:8080/api

Status: 400 Bad Request Size: 44 Bytes Time: 26 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "mensagem": "O nome precisa ser preenchido"  
3 }
```

Json Content Format

```
1 {  
2   "nome": "",  
3   "idade": 22  
4 }
```

Validou o nome pois conferiu se ele foi preenchido ou não validação é isso conferir se os dados estão corretos ou foram digitados corretamente

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

29°C Nublado

Go Live

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request ...

New Request Activity Collections Env filter activity

POST http://localhost:8080/api Send

Status: 400 Bad Request Size: 39 Bytes Time: 8 ms

Response Headers 4 Cookies Results Docs

1 {  
2 "mensagem": "Informe uma idade valida"  
3 }

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

1 {  
2 "nome": "Cristiano", ←  
3 "idade": -5  
4 }

Inserimos novos dados e enviamos mas com a idade errada e a idade foi validade pois ele informa que esta errada

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

29°C Nublado ⚡ ☰ 🔍 ⌂ 13:15

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Servico.java Controle.java New Request

New Request Activity Collections Env filter activity

POST http://localhost:8080/api Send

Status: 201 Created Size: 43 Bytes Time: 344 ms

Response Headers 4 Cookies Results Docs

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "Prisila",  
3   "idade": 27  
4 }
```

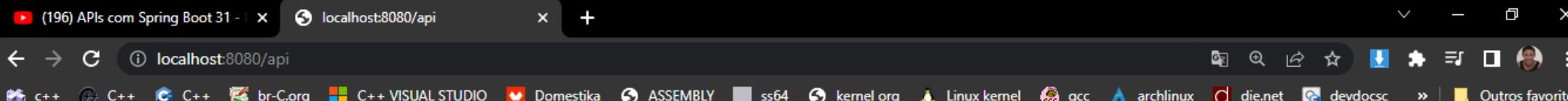
Por ultimo temos um novo usuario cadastrado até aqui tudo ok

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

29°C Nublado



```
[{"code":1,"name":"Akilles","age":"11"}, {"code":3,"name":"Carlos","age":"65"}, {"code":4,"name":"Joyce","age":"33"}, {"code":5,"name":"Cristiano","age":"33"}, {"code":6,"name":"Luke","age":"34"}, {"code":7,"name":"Darth Vader","age":"41"}, {"code":8,"name":"AnKIN","age":"9"}, {"code":9,"name":"Obi-Wan","age":"48"}, {"code":10,"nome":"Madara","idade":"100"}, {"codigo":11,"name":"Naruto","age":"20"}, {"code":12,"name":"Sasuke","age":"20"}, {"code":13,"name ":"Nanu","age":"90"}, {"code":14,"name":"Luna","age":"23"}, {"code":15,"name": "João","age":"34"}, {"code":16,"name":"Edson","age":"67"}, {"code":17,"name": "Eva ","idade":45}, {"codigo":18,"nome": "Adão","idade":700}, {"codigo":19,"nome": "Moises", "idade":1500}, {"codigo":20,"nome": "Não","age":100}, {"code":21,"name": "Maria","age":65}, {"code":22,"name": "Akilles","age":65}, {"code":23,"name": "Prisila","age":27}]
```

#32  
**IMPLEMENTANDO SERVIÇOS**  
**PT2**

File Edit Selection View Go Run Terminal Help Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER ... Servico.java X Controle.java New Request II ? ^ ^ ^ ^

APIS-SPRING-BOOT  
  > .vscode  
  < api  
  > .mvn  
  > .vscode  
  < src  
    > main  
      > java  
        > br  
          > com  
            > projeto  
              > api  
                > controle  
                  Controle.java  
                > modelo  
                  Mensagem.java  
                  Pessoa.java  
                > repositorio  
                  Repositorio.java  
                > servico  
                  Servico.java  
                  ApiApplication.java  
    > resources  
    > test  
  > target  
  > .gitignore  
  > HELP.md  
  > mvnw  
  > mvnw.cmd  
  > pom.xml  
  > README.md

api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java(TM) by Red Hat > Servico > cadastrar(Pessoa)

```
6 import org.springframework.stereotype.Service;  
7  
8 import br.com.projeto.api.modelo.Mensagem;  
9 import br.com.projeto.api.modelo.Pessoa;  
10 import br.com.projeto.api.repositorio.Repositorio;  
11  
12 @Service  
13 public class Servico {  
14  
15     @Autowired  
16     private Mensagem mensagem;  
17  
18     @Autowired  
19     private Repositorio acao;  
20  
21     // METODO PARA CADASTRAR PESSOAS  
22     public ResponseEntity<?> cadastrar(Pessoa obj){  
23  
24         if(obj.getNome().equals("")){  
25             mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");  
26             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);  
27         }else if(Integer.parseInt(obj.getIdade()) < 0){  
28             mensagem.setMensagem(mensagem: "Informe uma idade valida");  
29             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);  
30         }else{  
31             return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);  
32         }  
33     }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/n)

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** • Servico.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "servico" folder contains "Servico.java" (marked with a red error icon) and "ApiApplication.java". Other files like ".gitignore", "mvnw", "mvnw.cmd", "pom.xml", and "REDME.md" are also listed.
- Code Editor:** The "Servico.java" file is open. The code implements a service layer for managing people. It includes a dependency injection for a repository and two methods: "cadastrar" and "selecionar". The "cadastrar" method checks if the name is empty or the age is less than zero, returning an error response. Otherwise, it saves the person and returns a success response. The "selecionar" method is currently being typed and has a red underline, indicating a syntax error.
- Terminal:** Shows the command "api > src > main > java > br > com > projeto > api > servico > Servico.java > Language Support for Java(TM) by Red Hat > Servico.java".
- Bottom Status Bar:** PROBLEMS 1, OUTPUT, TERMINAL. A tooltip below the status bar says: "This method must return a result of type ResponseEntity<?> Java(603979884) [Ln 37, Col 30]."

```
17
18     -- Repositorio
19     @Autowired
20     private Repositorio acao;
21
22     // METODO PARA CADASTRAR PESSOAS
23     public ResponseEntity<?> cadastrar(Pessoa obj){
24
25         if(obj.getNome().equals("")){
26             mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");
27             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
28         }else if(Integer.parseInt(obj.getIdade()) < 0){
29             mensagem.setMensagem(mensagem: "Informe uma idade valida");
30             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
31         }else{
32             return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);
33         }
34     }
35
36     // METODO PARA SELECIONAR PESSOAS
37     public ResponseEntity<?> selecionar(){
38
39     }
40
41
42 }
```

## Criando metodo para selecionar pessoas



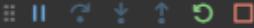
EXPLORER

...

Serviço.java X

Controle.java

New Request



APIS-SPRING-BOOT

&gt; .vscode

api

&gt; .mvn

&gt; .vscode

src

main

java

br

com

projeto

api

controle

Controle.java

modelo

Mensagem.java

Pessoa.java

repositorio

Repositorio.java

servico

Serviço.java

ApiApplication.java

&gt; resources

test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

17

← Repositorio

@Autowired

private Repositorio acao;

20

// METODO PARA CADASTRAR PESSOAS

public ResponseEntity&lt;?&gt; cadastrar(Pessoa obj){

23

if(obj.getNome().equals("")){

mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");

return new ResponseEntity&lt;?&gt;(mensagem, HttpStatus.BAD\_REQUEST);

}else if(Integer.parseInt(obj.getIdade()) &lt; 0){

mensagem.setMensagem(mensagem: "Informe uma idade valida");

return new ResponseEntity&lt;?&gt;(mensagem, HttpStatus.BAD\_REQUEST);

}else{

return new ResponseEntity&lt;?&gt;(acao.save(obj), HttpStatus.CREATED);

}

35

// METODO PARA SELECIONAR PESSOAS

public ResponseEntity&lt;?&gt; selecionar(){

38

return new ResponseEntity&lt;?&gt;(acao.findAll(), HttpStatus.OK);

39

}

40

}

41

}

42

}

PROBLEMS

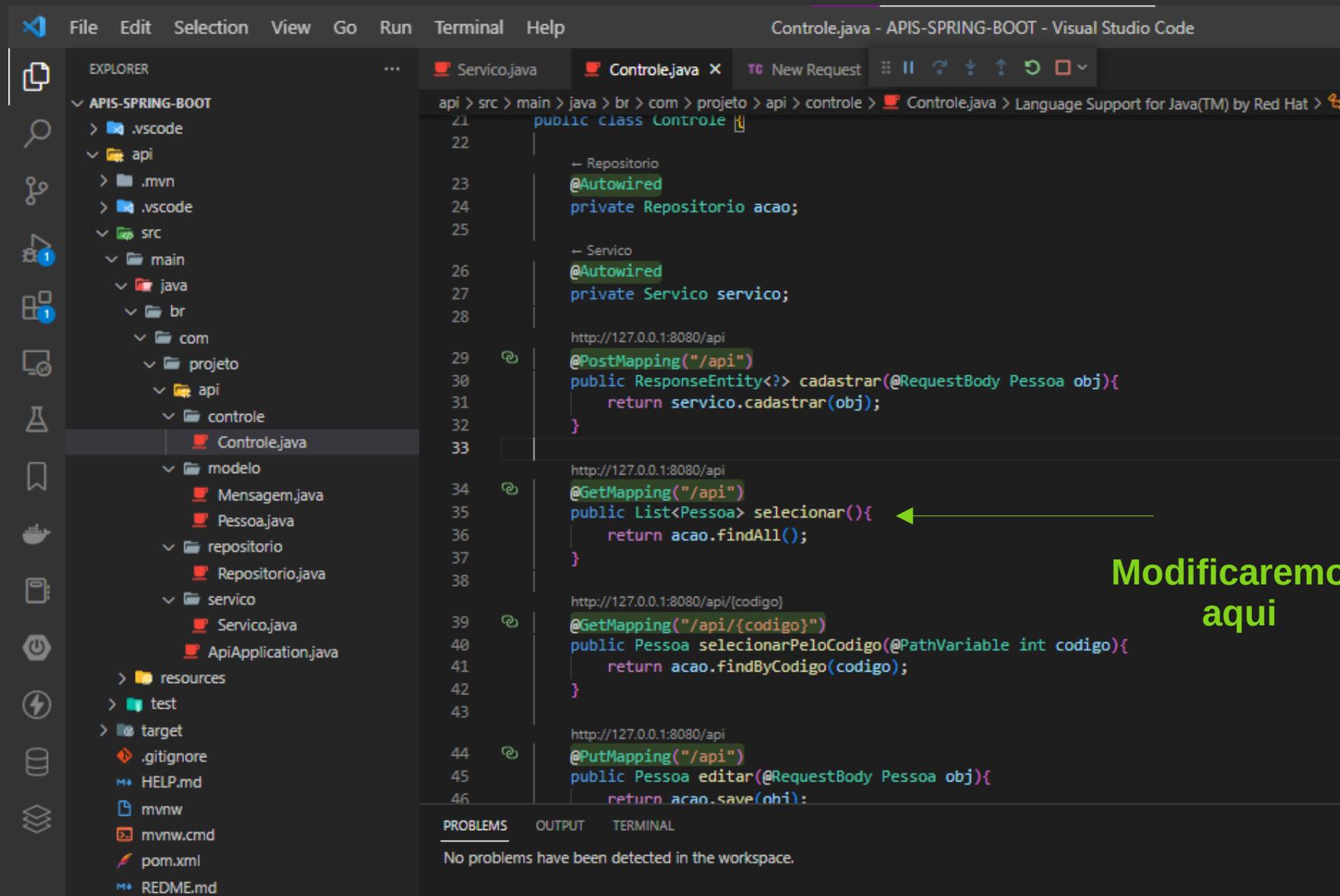
OUTPUT

TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/no)

**Agora no controle implementar nosso metodo criado**



File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
          - controle
            - Controle.java
  - modelo
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repositorio.java
  - servico
    - Servico.java
    - ApiApplication.java
- resources
- test
- target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- README.md

Servico.java    Controle.java X    New Request

Controle.java

```
21 public class Controle {
22     ...
23     ...
24     ...
25     ...
26     ...
27     ...
28     ...
29     ...
30     ...
31     ...
32     ...
33     ...
34     ...
35     ...
36     ...
37     ...
38     ...
39     ...
40     ...
41     ...
42     ...
43     ...
44     ...
45     ...
46     ...
```

http://127.0.0.1:8080/api  
@PostMapping("/api")  
public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){  
 return servico.cadastrar(obj);  
}  
  
http://127.0.0.1:8080/api  
@GetMapping("/api")  
public ResponseEntity<?> selecionar(){  
 return servico.selecionar();  
}  
  
http://127.0.0.1:8080/api/{codigo}  
@GetMapping("/api/{codigo}")  
public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
 return acao.findByCodigo(codigo);  
}  
  
http://127.0.0.1:8080/api  
@PutMapping("/api")  
public Pessoa editar(@RequestBody Pessoa obj){  
 return acao.save(obj);  
}

PROBLEMS    OUTPUT    TERMINAL

No problems have been detected in the workspace.

Trocamos pra  
isso

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/api 2 days ago

GET localhost:8080/api just now

POST localhost:8080/api 21 mins ago

GET localhost:8080/status 20 hours ago

POST localhost:8080/pessoa 23 hours ago

DEL localhost:8080/api/2 2 days ago

PUT localhost:8080/api 2 days ago

POST localhost:8080 3 days ago

Query Headers 2 Auth Body Tests Pre Run New

Send

Status: 200 OK Size: 935 Bytes Time: 32 ms

Response Headers 4 Cookies Results Docs

Certinho até agora

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

30°C Nublado

13:39

#33  
**IMPLEMENTANDO SERVIÇOS**  
**PT3**

EXPLORER    ...    Servico.java    Controle.java    Repository.java    New Request

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
  - < main
    - < java
      - < br
        - < com
          - < projeto
            - < api
              - Controle.java
            - < controle
              - Controle.java
- < repository
  - Repository.java

- < servico
- Servico.java
- ApiApplication.java
- > resources
- > test
- > target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- README.md

EXPLORER

- APIS-SPRING-BOOT
  - .vscode
  - api
    - .mvn
    - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
            - api
              - controle
              - Modelo
              - Repositorio
              - servico
                - Servico.java
                - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md
  - OUTLINE
  - TIMELINE
  - JAVA PROJECTS

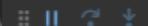
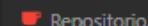
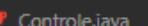
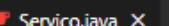
```
 17
 18     ← Repositorio
 19 @Autowired
 20     private Repositorio acao;
 21
 22 // METODO PARA CADASTRAR PESSOAS
 23 public ResponseEntity<?> cadastrar(Pessoa obj){
 24
 25     if(obj.getNome().equals("")){
 26         mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");
 27         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 28     }else if(Integer.parseInt(obj.getIdade()) < 0){
 29         mensagem.setMensagem(mensagem: "Informe uma idade valida");
 30         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 31     }else{
 32         return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);
 33     }
 34 }
 35
 36 // METODO PARA SELECIONAR PESSOAS
 37 public ResponseEntity<?> selecionar(){
 38     return new ResponseEntity<?>(acao.findAll(), HttpStatus.OK);
 39 }
 40
 41 // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO
 42 public ResponseEntity<?> selecionarPeloCodigo(int codigo){}
 43
 44
 45
 46 }
 47 }
```



EXPLORER

APIS-SPRING-BOOT

- > vscode
- < API
- > mvn
- > vscode
- < src
  - < main
    - < java
      - < br
        - < com
          - < projeto
            - < api
              - < controle
                - Controle.java
              - < modelo
                - Mensagem.java
                - Pessoa.java
            - < repositorio
              - Repositorio.java
          - < servico
            - Servico.java
            - ApiApplication.java
        - > resources
        - > test
      - < target
        - .gitignore
        - HELP.md
        - mvnw
        - mvnw.cmd
        - pom.xml
        - REDME.md



api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; servico &gt; Servico.java &gt; Language Support for Java(TM) by Red Hat &gt; Servico

```
20  
21 // METODO PARA CADASTRAR PESSOAS  
22 public ResponseEntity<?> cadastrar(Pessoa obj){  
23  
24     if(obj.getNome().equals("")){  
25         mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");  
26         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);  
27     }else if(Integer.parseInt(obj.getIdade()) < 0){  
28         mensagem.setMensagem(mensagem: "Informe uma idade valida");  
29         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);  
30     }else{  
31         return new ResponseEntity<?>(acao.save(obj), HttpStatus.CREATED);  
32     }  
33  
34 }  
35  
36 // METODO PARA SELECIONAR PESSOAS  
37 public ResponseEntity<?> selecionar(){  
38     return new ResponseEntity<?>(acao.findAll(), HttpStatus.OK);  
39 }  
40  
41 // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO  
42 public ResponseEntity<?> selecionarPeloCodigo(int codigo){  
43  
44     if(acao.countByCodigo(codigo) == 0){  
45         mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");  
46         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);  
47     }else{  
48         return new ResponseEntity<?>(acao.findByCodigo(codigo), HttpStatus.OK);  
49     }  
50  
51 }  
52  
53 }
```



&gt; OUTLINE

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g.

**Se não encontrar exibe uma mensagem de BAD\_REQUEST  
E se encontrar ele retorna uma pessoa contendo o código o nome e a idade  
E o status OK**

File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER ... Servico.java Controle.java Repositorio.java New Request

APIS-SPRING-BOOT

.vscode .mvn .vscode

api

src

main

java

br

com

projeto

api

controle

Controle.java

modelo

Mensagem.java

Pessoa.java

repositorio

Repositorio.java

servico

Servico.java

ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

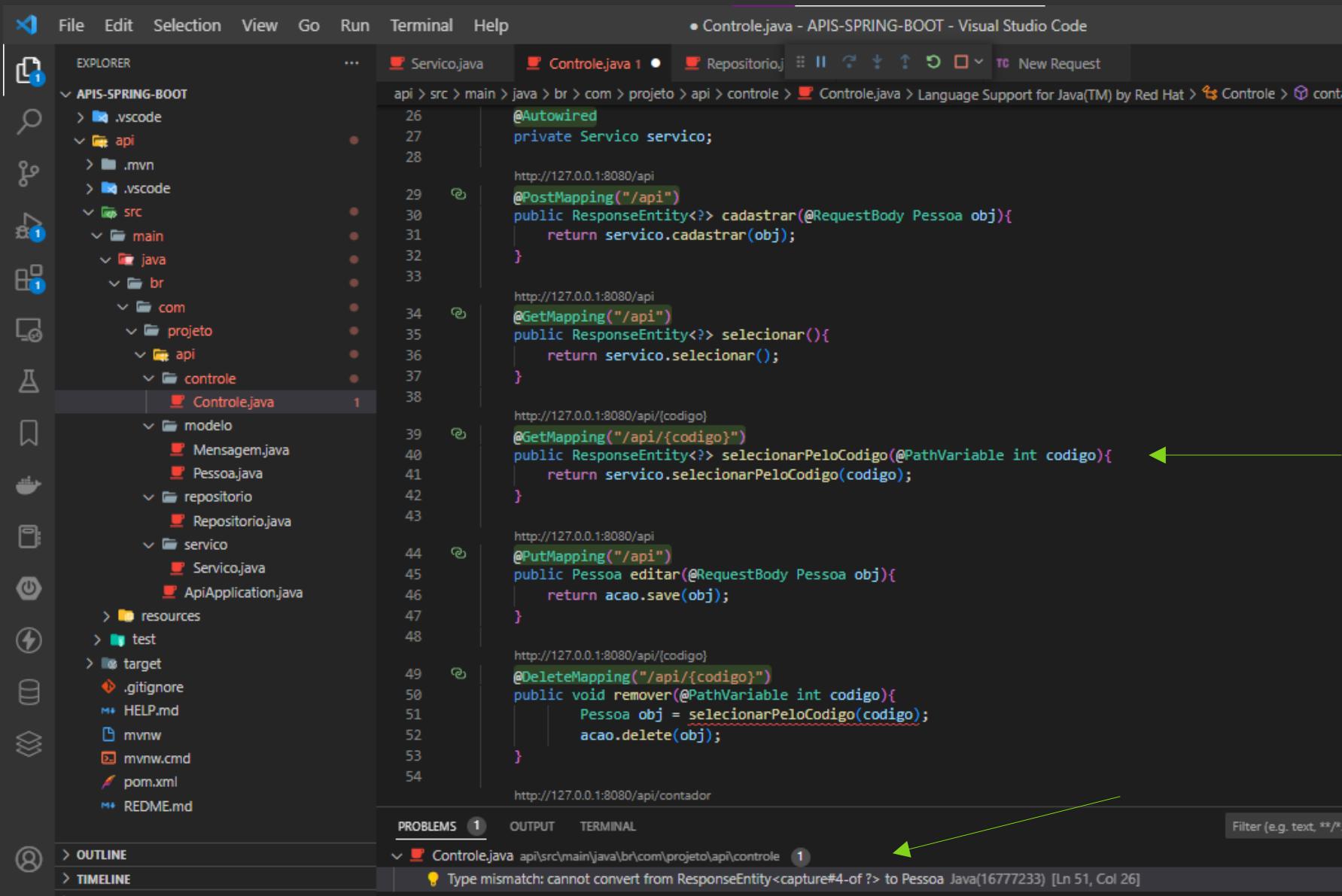
59

http://127.0.0.1:8080/api  
@GetMapping("/api")  
public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){  
 return servico.cadastrar(obj);  
}  
  
http://127.0.0.1:8080/api  
@GetMapping("/api")  
public ResponseEntity<?> selecionar(){  
 return servico.selecionar();  
}  
  
http://127.0.0.1:8080/api/{codigo}  
... @GetMapping("/api/{codigo}")  
... public Pessoa selecionarPeloCodigo(@PathVariable int codigo){  
... return acao.findByCodigo(codigo);  
... }  
  
http://127.0.0.1:8080/api  
@PutMapping("/api")  
public Pessoa editar(@RequestBody Pessoa obj){  
 return acao.save(obj);  
}  
  
http://127.0.0.1:8080/api/{codigo}  
@DeleteMapping("/api/{codigo}")  
public void remover(@PathVariable int codigo){  
 Pessoa obj = selecionarPeloCodigo(codigo);  
 acao.delete(obj);  
}  
  
http://127.0.0.1:8080/api/contador  
@GetMapping("/api/contador")  
public Long contador(){  
 return acao.count();  
}

Mudaremos aqui

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.



**Vamos resolver esse erro**



EXPLORER

APIS-SPRING-BOOT

&gt; .vscode

&gt; api

&gt; .mvn

&gt; .vscode

&gt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

Controle.java 1

&gt; modelo

Mensagem.java

Pessoa.java

&gt; repositorio

Repositorio.java

&gt; servico

Servico.java

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

Servico.java

Controle.java 1

Repositorio.j

New Request

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; controle &gt; Controle.java &gt; Language Support for Java(TM) by Red Hat &gt; Controle

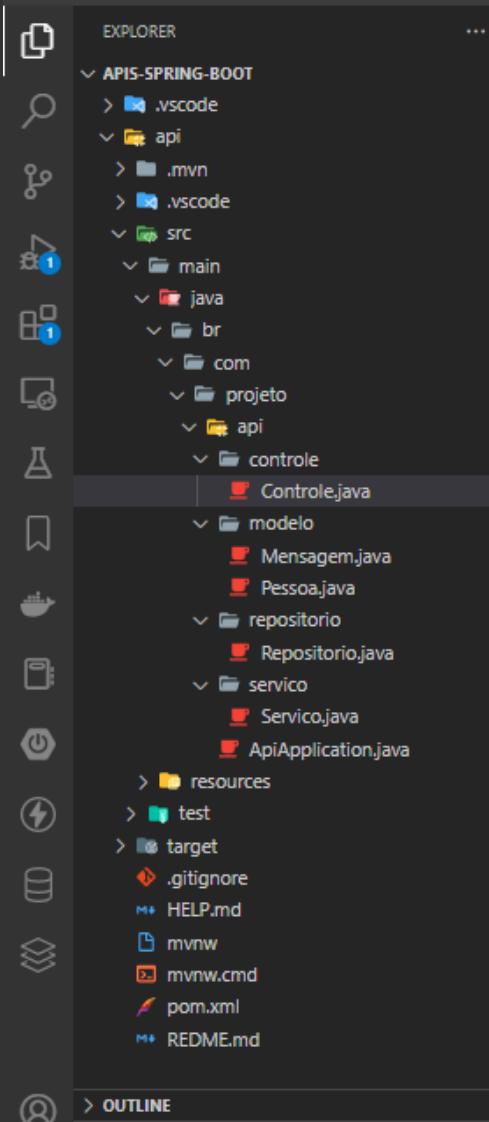
```
29  @PostMapping("/api")
30  public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){
31      return servico.cadastrar(obj);
32  }
33
34  http://127.0.0.1:8080/api
35  @GetMapping("/api")
36  public ResponseEntity<?> selecionar(){
37      return servico.selecionar();
38  }
39
40  http://127.0.0.1:8080/api/{codigo}
41  @GetMapping("/api/{codigo}")
42  public ResponseEntity<?> selecionarPeloCodigo(@PathVariable int codigo){
43      return servico.selecionarPeloCodigo(codigo);
44  }
45
46  http://127.0.0.1:8080/api
47  @PutMapping("/api")
48  public Pessoa editar(@RequestBody Pessoa obj){
49      return acao.save(obj);
50  }
51
52  http://127.0.0.1:8080/api/{codigo}
53  @DeleteMapping("/api/{codigo}")
54  public void remover(@PathVariable int codigo){
55      Pessoa obj = selecionarPeloCodigo(codigo);
56      acao.delete(obj);
57  }
58
59  http://127.0.0.1:8080/api/contador
60  @GetMapping("/api/contador")
61  public Long contador(){
62      return acao.count();
63  }
```

PROBLEMS 1 OUTPUT TERMINAL

Controle.java api\src\main\java\br\com\projeto\api\controle 1

Type mismatch: cannot convert from ResponseEntity&lt;capture#4-of ?&gt; to Pessoa Java(16777233) [Ln 51, Col 26]

Filter (e.g. text, \*\*/)



```
29  @PostMapping("/api")
30  public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){
31      return servico.cadastrar(obj);
32  }
33
34  http://127.0.0.1:8080/api
35  @GetMapping("/api")
36  public ResponseEntity<?> selecionar(){
37      return servico.selecionar();
38  }
39
40  http://127.0.0.1:8080/api/{codigo}
41  @GetMapping("/api/{codigo}")
42  public ResponseEntity<?> selecionarPeloCodigo(@PathVariable int codigo){
43      return servico.selecionarPeloCodigo(codigo);
44  }
45
46  http://127.0.0.1:8080/api
47  @PutMapping("/api")
48  public Pessoa editar(@RequestBody Pessoa obj){
49      return acao.save(obj);
50  }
51
52  http://127.0.0.1:8080/api/{codigo}
53  @DeleteMapping("/api/{codigo}")
54  public void remover(@PathVariable int codigo){
55      // Pessoa obj = selecionarPeloCodigo(codigo);
56      // acao.delete(obj);
57  }
58
59  http://127.0.0.1:8080/api/contador
60  @GetMapping("/api/contador")
61  public Long contador(){
62      return acao.count();
63  }
```

Comentando essas duas linhas o erro some

O erro era porque ele usava o metodo selecionarPeloCodigo();  
E como alteramos devemos comentar depois vamos concertar isso

The screenshot shows the Thunder Client extension in Visual Studio Code. The interface is divided into several sections:

- Left Sidebar:** Displays a list of activity logs for the "THUNDER CLIENT". The logs include various API requests (GET, POST, PUT, DELETE) made to "localhost:8080/api" and "localhost:8080/status" at different times (e.g., 2 days ago, 25 mins ago, 47 mins ago). A green arrow points from the sidebar to the "Activity" tab in the main request editor.
- Main Request Editor:** This is the central area where a new request is being configured.
  - Request Type:** Set to "GET".
  - URL:** "localhost:8080/api/30".
  - Buttons:** "Send" (blue), "Cancel" (red), and "Stop" (green).
  - Query Parameters:** A table with one row: "parameter" (empty) and "value" (empty). A green arrow points from the "Activity" log to this table.
- Response View:** Shows the result of the failed request.
  - Status:** 400 Bad Request
  - Size:** 50 Bytes
  - Time:** 406 ms
  - Response Content:** A JSON object: { "mensagem": "Não foi encontrada nenhuma pessoa." }

Testando enviamos uma pessoa que não existe e ele respondeu com uma BAD\_REQUEST

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Servico.java Controle.java New Request New Request

New Request Activity Collections Env filter activity

GET localhost:8080/api/1 Send

Status: 200 OK Size: 42 Bytes Time: 112 ms

Response Headers Cookies Results Docs

```
1 {  
2   "codigo": 1,  
3   "nome": "Akilles",  
4   "idade": "11"  
5 }
```

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters parameter value

Activity

filter activity

GET localhost:8080/api/1 just now

GET localhost:8080/api 2 days ago

GET localhost:8080/api 25 mins ago

POST localhost:8080/api 47 mins ago

GET localhost:8080/status 20 hours ago

POST localhost:8080/pessoa 23 hours ago

DEL localhost:8080/api/2 2 days ago

PUT localhost:8080/api 2 days ago

POST localhost:8080 3 days ago

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*\*)

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

29°C Nublado

Go Live

14:06

Agora enviamos um código que existe e ele retornou status OK  
Até aqui certinho  
Assim conseguimos validar se existe ou não um código na nossa tabela de pessoas

#34  
IMPLEMENTANDO SERVIÇOS  
PT4

**VAMOS IMPLEMENTAR O METODO EDITAR NA CLASSE DE SERVIÇO**

File Edit Selection View Go Run Terminal Help • Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER ... Servico.java 1 Controle.java New Requests New Request New

APIS-SPRING-BOOT .vscode api .mvn .vscode src main java br com projeto api servico Servico.java Language Support for Java(TM) by Red Hat

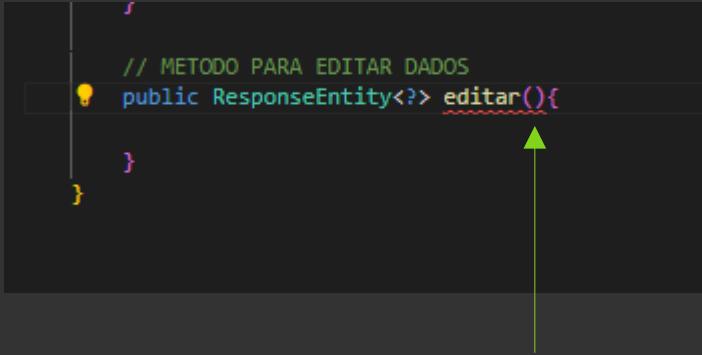
.vscode  
api  
.mvn  
.vscode  
src  
main  
java  
br  
com  
projeto  
api  
controle  
Mensagem.java  
Pessoa.java  
repositorio  
Repositorio.java  
servico  
Servico.java  
ApiApplication.java  
resources  
test  
target  
.gitignore  
HELP.md  
mvnw  
mvnw.cmd  
pom.xml  
REDME.md

```
25     mensagem.setMensagem(mensagem: "O nome precisa ser preenchido");
26     return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
27 }else if(Integer.parseInt(obj.getIdade()) < 0){
28     mensagem.setMensagem(mensagem: "Informe uma idade valida");
29     return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
30 }else{
31     return new ResponseEntity<>(acao.save(obj), HttpStatus.CREATED);
32 }
33
34 }
35
36 // METODO PARA SELECIONAR PESSOAS
37 public ResponseEntity<> selecionar(){
38     return new ResponseEntity<>(acao.findAll(), HttpStatus.OK);
39 }
40
41 // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO
42 public ResponseEntity<> selecionarPeloCodigo(int codigo){
43
44     if(acao.countByCodigo(codigo) == 0){
45         mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");
46         return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
47     }else{
48         return new ResponseEntity<>(acao.findByCodigo(codigo), HttpStatus.OK);
49     }
50 }
51
52 // METODO PARA EDITAR DADOS
53 public ResponseEntity<> editar(){
54
55 }
56
57 }
```

PROBLEMS 1 OUTPUT TERMINAL

Servico.java api\src\main\java\br\com\projeto\api\servico 1

This method must return a result of type ResponseEntity<> Java(603979884) [Ln 53, Col 30]

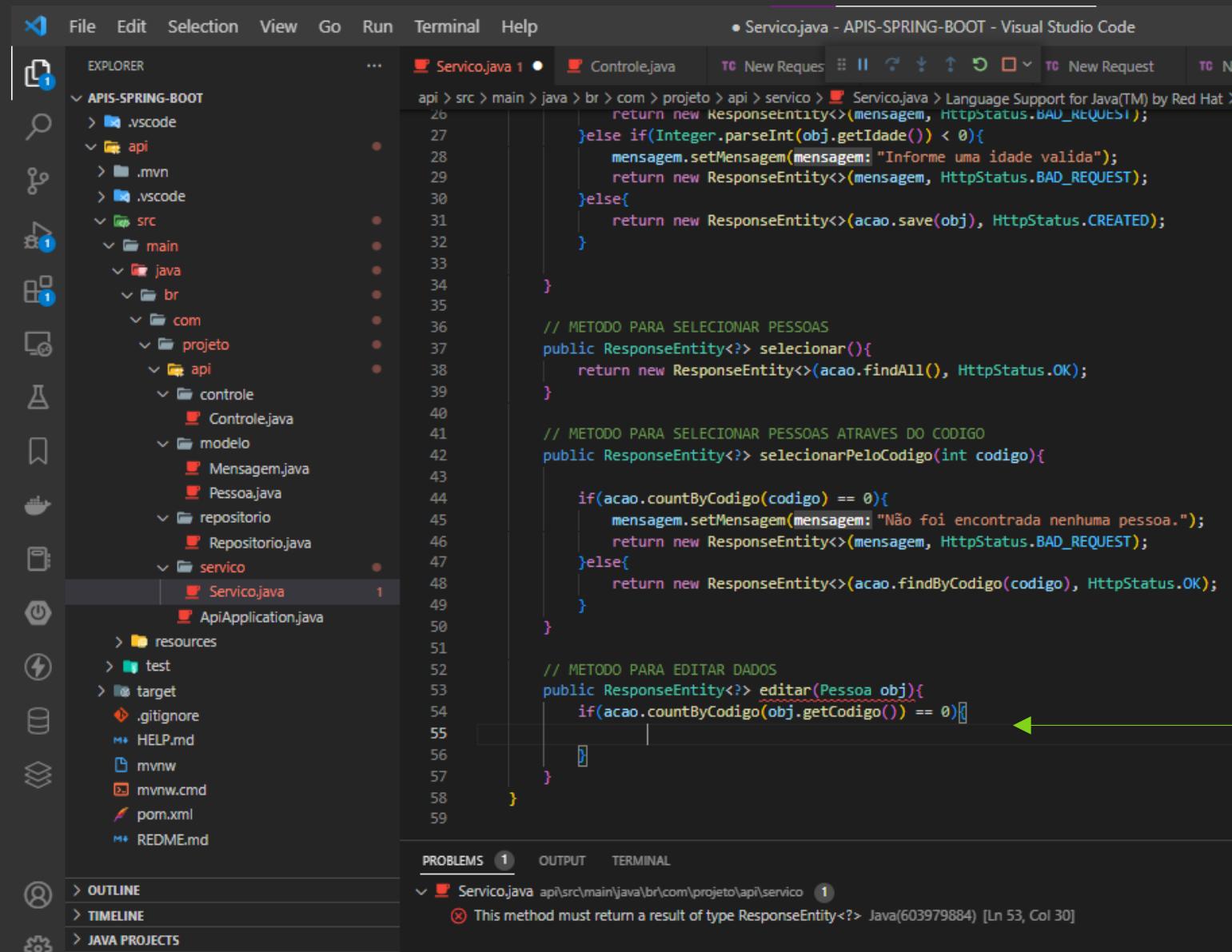


A screenshot of a Java code editor showing a portion of a class definition. The code is:

```
// METODO PARA EDITAR DADOS
public ResponseEntity<?> editar(){
}
```

The word "editar" is underlined with a pink squiggly line, indicating a potential error or suggestion. A green arrow points upwards from the bottom of the slide towards the underlined word.

Dentro dos parentesis precisamos de um objeto que tenha o código o nome e a idade





EXPLORER

APIS-SPRING-BOOT

.vscode

api

.mvn

.vscode

src

main

java

br

com

projeto

api

controle

Controle.java

modelo

Mensagem.java

Pessoa.java

repositorio

Repositorio.java

servico

Servico.java

ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

...  
Servico.java 1 ●

Controle.java

New Reques

New Request

New Req

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; servico &gt; Servico.java

Language Support for Java(TM) by Red Hat &gt; Se

26 return new ResponseEntity&lt;&gt;(mensagem, HttpStatus.BAD\_REQUEST);

27 }else if(Integer.parseInt(obj.getIdade()) &lt; 0){

28 mensagem.setMensagem(mensagem: "Informe uma idade valida");

29 return new ResponseEntity&lt;&gt;(mensagem, HttpStatus.BAD\_REQUEST);

30 }else{

31 return new ResponseEntity&lt;&gt;(acao.save(obj), HttpStatus.CREATED);

32 }

33 }

34 }

35 }

36 // METODO PARA SELECIONAR PESSOAS

37 public ResponseEntity&lt;&gt; selecionar(){

38 return new ResponseEntity&lt;&gt;(acao.findAll(), HttpStatus.OK);

39 }

40

41 // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO

42 public ResponseEntity&lt;&gt; selecionarPeloCodigo(int codigo){

43

44 if(acao.countByCodigo(codigo) == 0){

45 mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");

46 return new ResponseEntity&lt;&gt;(mensagem, HttpStatus.BAD\_REQUEST);

47 }else{

48 return new ResponseEntity&lt;&gt;(acao.findByCodigo(codigo), HttpStatus.OK);

49 }

50 }

51

52 // METODO PARA EDITAR DADOS

53 public ResponseEntity&lt;&gt; editar(Pessoa obj){

54 if(acao.countByCodigo(obj.getCodigo()) == 0){

55 mensagem.setMensagem(mensagem: "O codigo informado não existe.");

56 return new ResponseEntity&lt;&gt;(mensagem, HttpStatus.NOT\_FOUND);

57 }

58 }

59 }

60 }

PROBLEMS 1

OUTPUT

TERMINAL

&gt; OUTLINE Servico.java api\src\main\java\br\com\projeto\api\servico 1

&gt; TIMELINE &gt; Timeline

&gt; PROBLEMS

1

This method must return a result of type ResponseEntity&lt;&gt; Java(603979884) [Ln 53, Col 30]



EXPLORER

APIS-SPRING-BOOT



vscode



api



.mvn



.vscode



src



main



java



br



com



projeto



api



controle



Controle.java



modelo



Mensagem.java



Pessoa.java



repositorio



Repositorio.java



servico



Servico.java



ApiApplication.java



resources



test



target



.gitignore



HELP.md



mvnw



mvnw.cmd



pom.xml



README.md

```
 33
 34     }
 35
 36     // METODO PARA SELECIONAR PESSOAS
 37     public ResponseEntity<?> selecionar(){
 38         return new ResponseEntity<?>(acao.findAll(), HttpStatus.OK);
 39     }
 40
 41     // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO
 42     public ResponseEntity<?> selecionarPeloCodigo(int codigo){
 43
 44         if(acao.countByCodigo(codigo) == 0){
 45             mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");
 46             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 47         }else{
 48             return new ResponseEntity<?>(acao.findByCodigo(codigo), HttpStatus.OK);
 49         }
 50     }
 51
 52     // METODO PARA EDITAR DADOS
 53     public ResponseEntity<?> editar(Pessoa obj){
 54         if(acao.countByCodigo(obj.getCodigo()) == 0){
 55             mensagem.setMensagem(mensagem: "O codigo informado não existe.");
 56             return new ResponseEntity<?>(mensagem, HttpStatus.NOT_FOUND);
 57         }else if(obj.getNome().equals("")){
 58             mensagem.setMensagem(mensagem: "É necessário informar um nome");
 59             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
 60         }
 61     }
 62 }
```

PROBLEMS

OUTPUT

TERMINAL

Servico.java api\src\main\java\br\com\projeto\api\servico

This method must return a result of type ResponseEntity&lt;?&gt; Java(603979884) [Ln 53, Col 30]

File Edit Selection View Go Run Terminal Help • Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle
- Controle.java
- modelo
- Mensagem.java
- Pessoa.java
- repositorio
- Repositorio.java
- servico
- Servico.java
- ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

Servico.java 1

Controle.java

New Request

New Request

```
33
34     }
35
36     // METODO PARA SELECIONAR PESSOAS
37     public ResponseEntity<?> selecionar(){
38         return new ResponseEntity<?>(acao.findAll(), HttpStatus.OK);
39     }
40
41     // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO
42     public ResponseEntity<?> selecionarPeloCodigo(int codigo){
43
44         if(acao.countByCodigo(codigo) == 0){
45             mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");
46             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
47         }else{
48             return new ResponseEntity<?>(acao.findByCodigo(codigo), HttpStatus.OK);
49         }
50     }
51
52     // METODO PARA EDITAR DADOS
53     public ResponseEntity<?> editar(Pessoa obj){
54         if(acao.countByCodigo(obj.getCodigo()) == 0){
55             mensagem.setMensagem(mensagem: "O código informado não existe.");
56             return new ResponseEntity<?>(mensagem, HttpStatus.NOT_FOUND);
57         }else if(obj.getNome().equals("")){
58             mensagem.setMensagem(mensagem: "É necessário informar um nome");
59             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
60         }else if(obj.getIdade() < 0){
61             mensagem.setMensagem(mensagem: "Informe uma idade válida.");
62             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
63         }
64     }
65 }
66 }
```

PROBLEMS 1

OUTPUT TERMINAL

Filter (e.g.)

Servico.java api\src\main\java\br\com\projeto\api\servico 1

The operator < is undefined for the argument type(s) String, int Java(536871072) [Ln 60, Col 18]

**Vamos resolver o erro de parse**

File Edit Selection View Go Run Terminal Help • Servico.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
        - api
          - controle
  - modelo
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repository.java
  - servico
    - Servico.java
    - ApiApplication.java
- resources
- test
- target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

Servico.java 1

Controle.java

New Requests

New Request

New

api > src > main > java > br > com > projeto > api > servico > Servico.java > ...

```
36     // METODO PARA SELECIONAR PESSOAS
37     public ResponseEntity<?> selecionar(){
38         return new ResponseEntity<>(acao.findAll(), HttpStatus.OK);
39     }
40
41     // METODO PARA SELECIONAR PESSOAS ATRAVES DO CODIGO
42     public ResponseEntity<?> selecionarPeloCodigo(int codigo){
43
44         if(acao.countByCodigo(codigo) == 0){
45             mensagem.setMensagem(mensagem: "Não foi encontrada nenhuma pessoa.");
46             return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
47         }else{
48             return new ResponseEntity<>(acao.findByCodigo(codigo), HttpStatus.OK);
49         }
50     }
51
52     // METODO PARA EDITAR DADOS
53     public ResponseEntity<?> editar(Pessoa obj){
54         if(acao.countByCodigo(obj.getCodigo()) == 0){
55             mensagem.setMensagem(mensagem: "O codigo informado não existe.");
56             return new ResponseEntity<>(mensagem, HttpStatus.NOT_FOUND);
57         }else if(obj.getNome().equals("")){
58             mensagem.setMensagem(mensagem: "É necessário informar um nome");
59             return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
60         }else if(Integer.parseInt(obj.getIdade()) < 0){ ←
61             mensagem.setMensagem(mensagem: "Informe uma idade válida.");
62             return new ResponseEntity<>(mensagem, HttpStatus.BAD_REQUEST);
63         }
64     }
65 }
66 }
```

PROBLEMS 1

OUTPUT

TERMINAL

Servico.java api\src\main\java\br\com\projeto\api\servico 1

This method must return a result of type ResponseEntity<?> Java(603979884) [Ln 53, Col 30]

**Erro de parse resolvido**

EXPLORER    ...    Servico.java    Controle.java    New Request    New Request    New Request

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
              - modelo
                - Mensagem.java
                - Pessoa.java
              - repositorio
                - Repositorio.java
              - servico
                - Servico.java
  - ApiApplication.java
  - resources
  - test
  - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md

PROBLEMS    OUTPUT    TERMINAL

No problems have been detected in the workspace.

Filter

**Vamos no controle implementar essa ação**

File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - modelo
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repository.java
  - servico
    - Servico.java
  - ApiApplication.java
- resources
- test
- target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- README.md

Servico.java

Controle.java X TC New Request

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat http://127.0.0.1:8080/api

```
29 @PostMapping("/api")
30     public ResponseEntity<?> cadastrar(@RequestBody Pessoa obj){
31         return servico.cadastrar(obj);
32     }
33
34 http://127.0.0.1:8080/api
35 @GetMapping("/api")
36     public ResponseEntity<?> selecionar(){
37         return servico.selecionar();
38     }
39 http://127.0.0.1:8080/api/{codigo}
40 @GetMapping("/api/{codigo}")
41     public ResponseEntity<?> selecionarPeloCodigo(@PathVariable int codigo){
42         return servico.selecionarPeloCodigo(codigo);
43     }
44 http://127.0.0.1:8080/api
45 @PutMapping("/api")
46     public Pessoa editar(@RequestBody Pessoa obj){
47         acao.save(obj);
48     }
49 http://127.0.0.1:8080/api/{codigo}
50 @DeleteMapping("/api/{codigo}")
51     public void remover(@PathVariable int codigo){
52         // Pessoa obj = selecionarPeloCodigo(codigo);
53         // acao.delete(obj);
54     }
55 http://127.0.0.1:8080/api/contador
56 @GetMapping("/api/contador")
57     public Long contador(){
58         return acao.count();
59     }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Mudaremos aqui

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
  - modelo
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repositorio.java
  - servico
    - Serviço.java
    - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Servico.java Controle.java New Request New Request

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat > Controle > editar(Pessoa)

```
29     @PostMapping("/api")
30     public ResponseEntity<> cadastrar(@RequestBody Pessoa obj){
31         return servico.cadastrar(obj);
32     }
33
34     http://127.0.0.1:8080/api
35     @GetMapping("/api")
36     public ResponseEntity<> selecionar(){
37         return servico.selecionar();
38     }
39
40     http://127.0.0.1:8080/api/{codigo}
41     @GetMapping("/api/{codigo}")
42     public ResponseEntity<> selecionarPeloCodigo(@PathVariable int codigo){
43         return servico.selecionarPeloCodigo(codigo);
44     }
45
46     http://127.0.0.1:8080/api
47     @PutMapping("/api")
48     public ResponseEntity<> editar(@RequestBody Pessoa obj){ ←
49         return servico.editar(obj);
50     }
51
52     http://127.0.0.1:8080/api/{codigo}
53     @DeleteMapping("/api/{codigo}")
54     public void remover(@PathVariable int codigo){
55         // Pessoa obj = selecionarPeloCodigo(codigo);
56         //acao.delete(obj);
57     }
58
59     http://127.0.0.1:8080/api/contador
60     @GetMapping("/api/contador")
61     public Long contador(){
62         returnacao.count();
63     }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, \*\*/node\_modules/\*\*)



File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Servico.java Controle.java New Request Repositorio.java New Request New Request

New Request

Activity Collections Env filter activity

PUT localhost:8080/api Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Headers 2 Auth Body Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

1

Send Request Ctrl + Enter Import CUrl Ctrl + U Change Environment Ctrl + E Git Sync Details

Agora preenchemos o json com os dados do usuario que queremos mudar

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

29°C Nublado

localhost:8080/api

localhost:8080/api

```
[{"code":1,"name":"Akilles","age":11}, {"code":3,"name":"Carlos","age":65}, {"code":4,"name":"Joyce","age":33}, {"code":5,"name":"Cristiano","age":33}, {"code":6,"name":"Luke","age":34}, {"code":7,"name":"Darth Vader","age":41}, {"code":8,"name":"AnKIN","age":9}, {"code":9,"name":"Obi-Wan","age":48}, {"code":10,"name":"Madara","idade":100}, {"codigo":11,"name":"Naruto","age":20}, {"code":12,"name":"Sasuke","age":20}, {"code":13,"name":"Nanu","age":90}, {"code":14,"name":"Luna","age":23}, {"code":15,"name":"João","age":34}, {"code":16,"name":"Edson","age":67}, {"code":17,"name":"Eva","idade":45}, {"codigo":18,"name":"Adão","idade":700}, {"codigo":19,"name":"Moises","idade":1500}, {"codigo":20,"name":"Não","age":100}, {"code":21,"name":"Maria","age":65}, {"code":22,"name":"Akilles","age":65}, {"code":23,"name":"Prisila","age":27}]
```

Copiei e vou colar pra mudar os dados desse usuário

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request ... Repository.java New Request New Request

New Request

Activity Collections Env filter activity

GET localhost:8080/api/1 31 mins ago  
GET localhost:8080/api 2 days ago  
GET localhost:8080/api 58 mins ago  
POST localhost:8080/api 1 hour ago  
GET localhost:8080/status 21 hours ago  
POST localhost:8080/pessoa 24 hours ago  
DEL localhost:8080/api/2 2 days ago  
PUT localhost:8080/api 2 days ago  
POST localhost:8080 3 days ago

PUT localhost:8080/api Send

Status: Size: Time: Response Headers Cookies Results Docs

Json Content Format

```
1 {"code":23,"name":"Prisila","age":27}
```

Send Request Ctrl + Enter  
Import CUrl Ctrl + U  
Change Environment Ctrl + E  
Git Sync Details

Colei os dados que vou editar PUT  
Agora vamos fazer os testes

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

URGENTE 14:40

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request ... Repository.java New Request New Request

New Request

Activity Collections Env filter activity

PUT localhost:8080/api Send

Status: 404 Not Found Size: 46 Bytes Time: 65 ms

Response Headers 4 Cookies Results Docs

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {"code":40,"name":"Prisila","age":"27"}
```

Informei um codigo que não existe deu certo

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

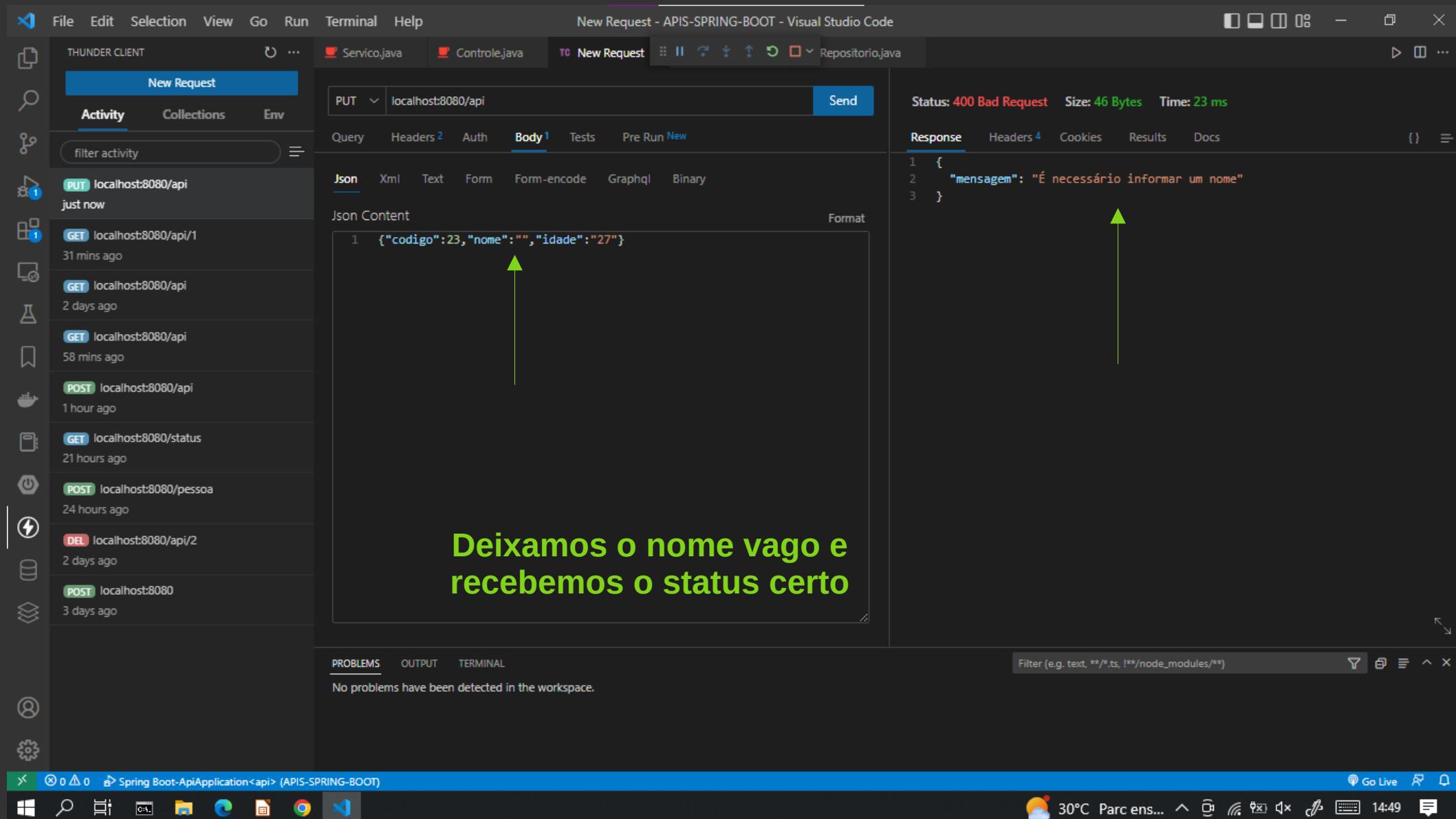
Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

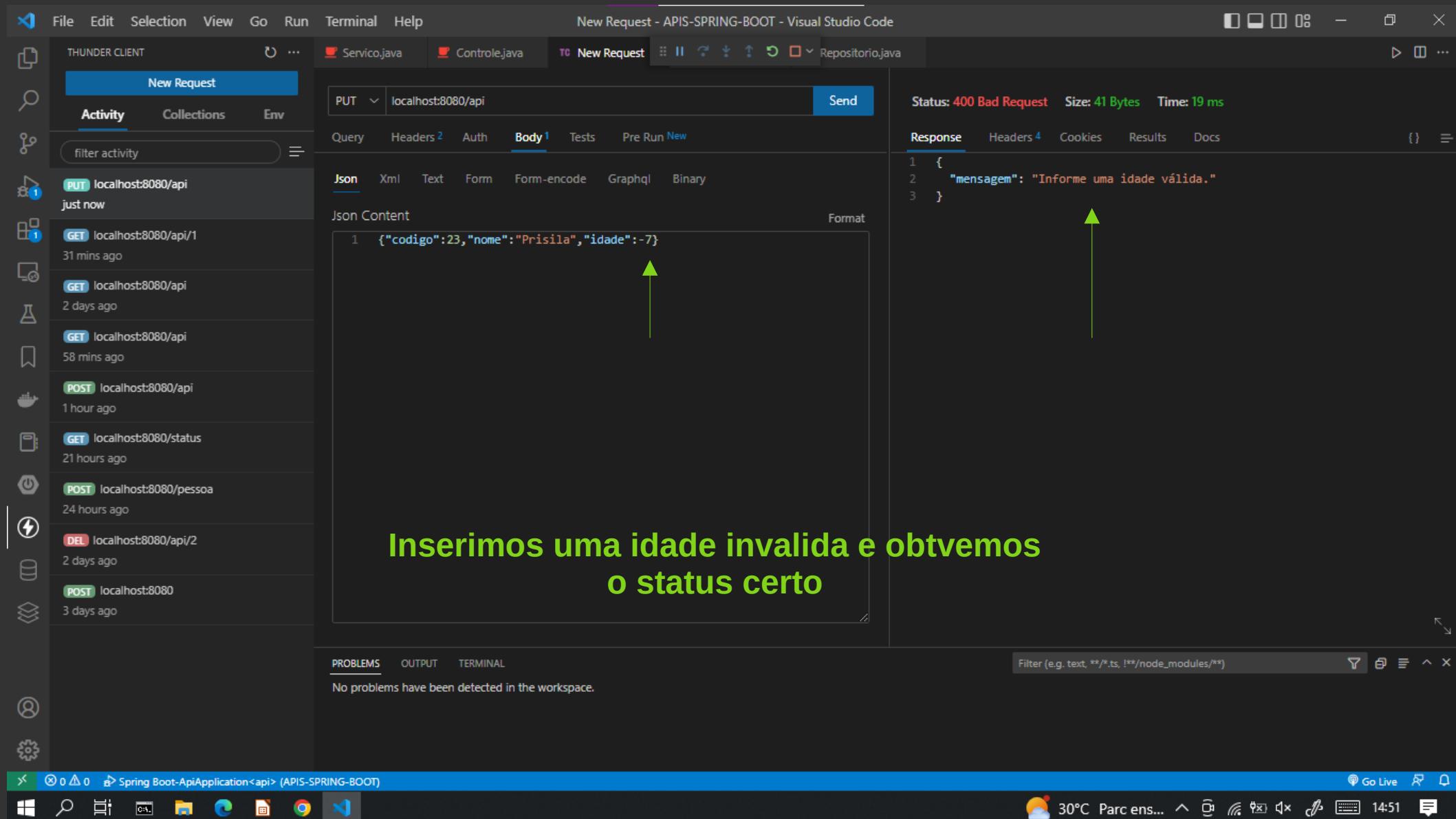
30°C Parc ens... 14:42

localhost:8080/api

```
[{"codigo":1,"nome":"Akilles","idade":11}, {"codigo":3,"nome":"Carlos","idade":65}, {"codigo":4,"nome":"Joyce","idade":33}, {"codigo":5,"nome":"Cristiano","idade":33}, {"codigo":6,"nome":"Luke","idade":34}, {"codigo":7,"nome":"Darth Vader","idade":41}, {"codigo":8,"nome":"AnKIN","idade":9}, {"codigo":9,"nome":"Obi-Wan","idade":48}, {"codigo":10,"nome":"Madara","idade":100}, {"codigo":11,"nome":"Naruto","idade":20}, {"codigo":12,"nome":"Sasuke","idade":20}, {"codigo":13,"nome":"Nanu","idade":90}, {"codigo":14,"nome":"Luna","idade":23}, {"codigo":15,"nome":"Joao","idade":34}, {"codigo":16,"nome":"Edson","idade":67}, {"codigo":17,"nome":"Eva","idade":45}, {"codigo":18,"nome":"Adao","idade":700}, {"codigo":19,"nome":"Moises","idade":1500}, {"codigo":20,"nome":"Noe","idade":100}, {"codigo":21,"nome":"Maria","idade":65}, {"codigo":22,"nome":"Akilles","idade":65}, {"codigo":23,"nome":"Prisila","idade":27}]
```

Tinha dado um erro preste atenção na tradução  
continuando





File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Servico.java Controle.java New Request Repositorio.java

New Request Activity Collections Env filter activity

PUT localhost:8080/api Send

Status: 200 OK Size: 44 Bytes Time: 384 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "codigo": 23,  
3   "nome": "Prisilla",  
4   "idade": "77"  
5 }
```

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

Inserimos dados validos agora e obtivemos o status certo

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

```
[{"codigo":1,"nome":"Akilles","idade":11}, {"codigo":3,"nome":"Carlos","idade":65}, {"codigo":4,"nome":"Joyce","idade":33}, {"codigo":5,"nome":"Cristiano","idade":33}, {"codigo":6,"nome":"Luke","idade":34}, {"codigo":7,"nome":"Darth Vader","idade":41}, {"codigo":8,"nome":"AnKIN","idade":9}, {"codigo":9,"nome":"Obi-Wan","idade":48}, {"codigo":10,"nome":"Madara","idade":100}, {"codigo":11,"nome":"Naruto","idade":20}, {"codigo":12,"nome":"Sasuke","idade":20}, {"codigo":13,"nome":"Nanu","idade":90}, {"codigo":14,"nome":"Luna","idade":23}, {"codigo":15,"nome":"Joao","idade":34}, {"codigo":16,"nome":"Edson","idade":67}, {"codigo":17,"nome":"Eva","idade":45}, {"codigo":18,"nome":"Adao","idade":700}, {"codigo":19,"nome":"Moises","idade":1500}, {"codigo":20,"nome":"Noe","idade":100}, {"codigo":21,"nome":"Maria","idade":65}, {"codigo":22,"nome":"Akilles","idade":65}, {"codigo":23,"nome":"Prisilla","idade":77}]
```

**Atualizando o navegador os dados da priscilla foram alterados**

#35  
**IMPLEMENTANDO SERVIÇOS**  
PT5

## **IMPLEMENTANDO O METODO REMOVER NA CLASSE DE SERVIÇO**

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Servico.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT".
  - api
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
          - api
            - controle
            - Servico.java
  - resources
  - test
  - target
    - .gitignore
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
    - REDME.md
- Code Editor:** Displays Java code for the Servico.java file.

```
48     return new ResponseEntity<?>(acao.findByCodigo(codigo), HttpStatus.OK);
49 }
50 }
51
52 // METODO PARA EDITAR DADOS
53 public ResponseEntity<?> editar(Pessoa obj){
54     if(acao.countByCodigo(obj.getCodigo()) == 0){
55         mensagem.setMensagem(mensagem: "O código informado não existe.");
56         return new ResponseEntity<?>(mensagem, HttpStatus.NOT_FOUND);
57     }else if(obj.getNome().equals("")){
58         mensagem.setMensagem(mensagem: "É necessário informar um nome");
59         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
60     }else if(Integer.parseInt(obj.getIdade()) < 0){
61         mensagem.setMensagem(mensagem: "Informe uma idade válida.");
62         return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
63     }else{
64         return new ResponseEntity<?>(acao.save(obj), HttpStatus.OK);
65     }
66 }
67
68 // METODO PARA REMOVER REGISTROS
69 public ResponseEntity<?> remover(){
70 }
71 }
72 }
73 }
```
- Bottom Status Bar:** PROBLEMS 1, OUTPUT, TERMINAL.
- Bottom Error List:** Shows an error for Servico.java: "This method must return a result of type ResponseEntity<?> Java(603979884) [Ln 69, Col 30]."

**Text Overlay:** Esse metodo vai ter parametros

The screenshot shows a Java code editor with a dark theme. On the left is the project structure:

- Pessoa.java
- repository

  - Repository.java

- servico

  - Servico.java (highlighted)
  - ApiApplication.java

- resources
- test
- target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

The main editor area contains the following code:

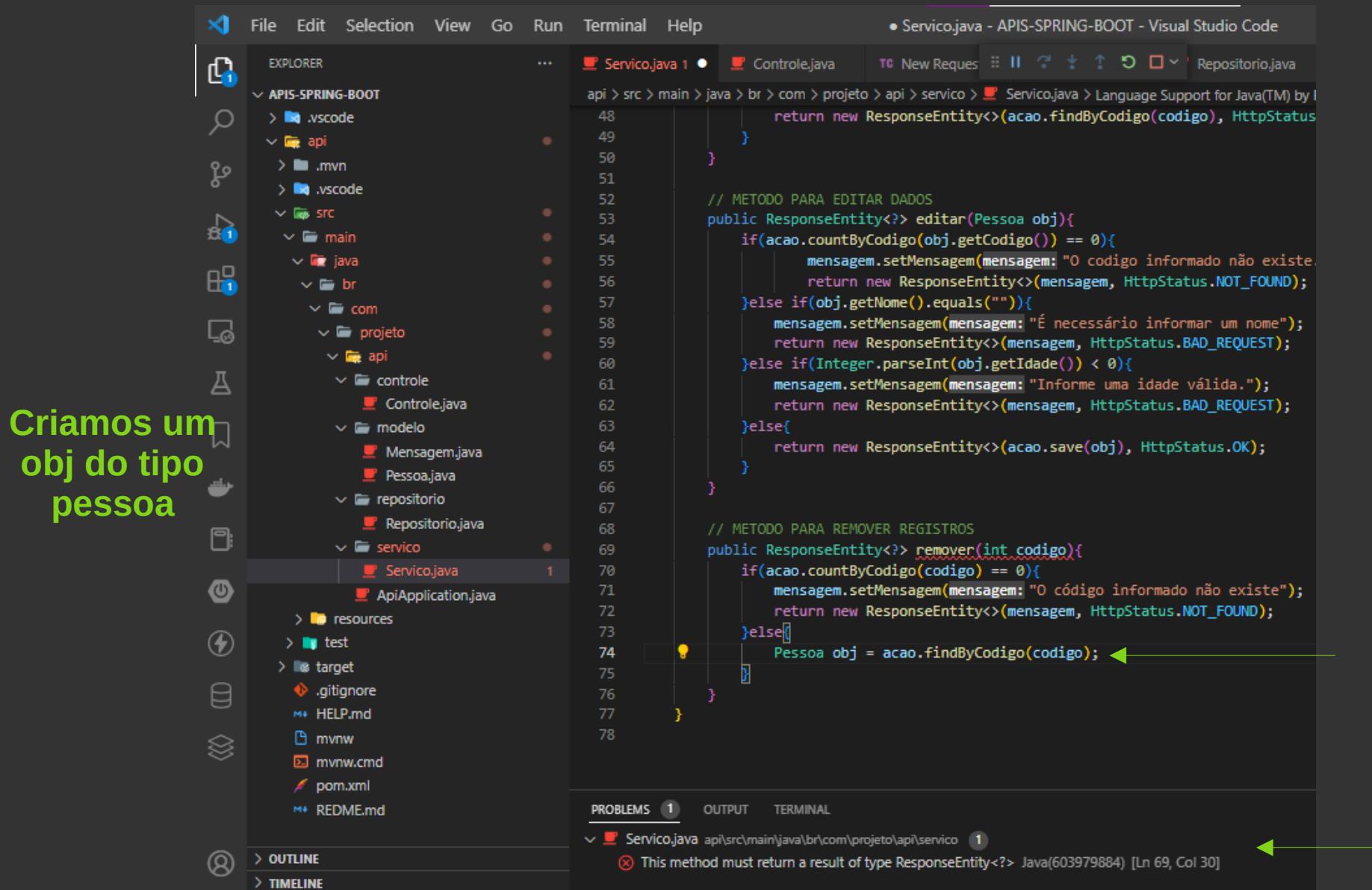
```
65     }
66 }
67
68 // METODO PARA REMOVER REGISTROS
69 public ResponseEntity<?> remover(int codigo){
70
71 }
72 }
73 }
```

A light blue arrow points upwards from the code editor towards the parameter "codigo" in the method signature. Below the editor is a navigation bar with tabs: PROBLEMS (1), OUTPUT, and TERMINAL.

The PROBLEMS tab shows one error:

- Servico.java api\src\main\java\br\com\projeto\api\servico 1  
This method must return a result of type ResponseEntity<?> Java(603979884) [Ln 69, Col 30]

O parametro vai ter um inteiro codigo  
Usaremos pra saber se o codigo existe ou não na tabela





EXPLORER



APIS-SPRING-BOOT



.vscode



api



.mvn



.vscode



src



main



java



br



com



projeto



api



controle



Controle.java



modelo



Mensagem.java



Pessoa.java



repository



Repository.java



servico



Servico.java



ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

README.md

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; servico &gt; Servico.java &gt; ...

```
51
52     // METODO PARA EDITAR DADOS
53     public ResponseEntity<?> editar(Pessoa obj){
54         if(acao.countByCodigo(obj.getCodigo()) == 0){
55             mensagem.setMensagem(mensagem: "O código informado não existe.");
56             return new ResponseEntity<?>(mensagem, HttpStatus.NOT_FOUND);
57         }else if(obj.getNome().equals("")){
58             mensagem.setMensagem(mensagem: "É necessário informar um nome");
59             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
60         }else if(Integer.parseInt(obj.getIdade()) < 0){
61             mensagem.setMensagem(mensagem: "Informe uma idade válida.");
62             return new ResponseEntity<?>(mensagem, HttpStatus.BAD_REQUEST);
63         }else{
64             return new ResponseEntity<?>(acao.save(obj), HttpStatus.OK);
65         }
66     }
67
68     // METODO PARA REMOVER REGISTROS
69     public ResponseEntity<?> remover(int codigo){
70         if(acao.countByCodigo(codigo) == 0){
71             mensagem.setMensagem(mensagem: "O código informado não existe");
72             return new ResponseEntity<?>(mensagem, HttpStatus.NOT_FOUND);
73         }else{
74             Pessoa obj = acao.findByCodigo(codigo);
75             acao.delete(obj);
76
77             mensagem.setMensagem(mensagem: "Pessoa removida com sucesso!");
78             return new ResponseEntity<?>(mensagem, HttpStatus.OK);
79         }
80     }
81 }
82 }
```

PROBLEMS

OUTPUT

TERMINAL

Filter (e.g.



&gt; OUTLINE

File Edit Selection View Go Run Terminal Help

Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
          - api
          - controle
  - modelo
  - Mensagem.java
  - Pessoa.java
  - repositorio
  - Repository.java
  - servico
  - Servico.java
  - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

Service.java Controle.java New Request Repository.java

api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM) by Red Hat http://127.0.0.1:8080/api/{codigo}

```
39 @GetMapping("/api/{codigo}")
40     public ResponseEntity<?> selecionarPeloCodigo(@PathVariable int codigo){
41         return servico.selecionarPeloCodigo(codigo);
42     }
43
44     http://127.0.0.1:8080/api
45     @PutMapping("/api")
46     public ResponseEntity<?> editar(@RequestBody Pessoa obj){
47         return servico.editar(obj);
48     }
49     http://127.0.0.1:8080/api/{codigo}
50     @DeleteMapping("/api/{codigo}")
51     public void remover(@PathVariable int codigo){
52         //Pessoa obj = selecionarPeloCodigo(codigo);
53         //acao.delete(obj);
54     }
55     http://127.0.0.1:8080/api/contador
56     @GetMapping("/api/contador")
57     public Long contador(){
58         return acao.count();
59     }
60     http://127.0.0.1:8080/api/ordenarNomes
61     @GetMapping("/api/ordenarNomes")
62     public List<Pessoa> ordenarNomes(){
63         return acao.findByOrderByNomeDesc();
64     }
65     http://127.0.0.1:8080/api/ordenarNomes2
66     @GetMapping("/api/ordenarNomes2")
67     public List<Pessoa> ordenarNomes2(){
```

Mudaremos aqui

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
  - controle
    - Controle.java
  - modelo
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repository.java
  - servico
    - Servico.java
    - ApiApplication.java
- resources
- test
- target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - REDME.md

```
Controle.java X New Request Repository.java
```

http://127.0.0.1:8080/api/{codigo}

```
@GetMapping("/api/{codigo}")
public ResponseEntity<?> selecionarPeloCodigo(@PathVariable int codigo){
    return servico.selecionarPeloCodigo(codigo);
}
```

http://127.0.0.1:8080/api

```
@PutMapping("/api")
public ResponseEntity<?> editar(@RequestBody Pessoa obj){
    return servico.editar(obj);
}
```

http://127.0.0.1:8080/api/{codigo}

```
@DeleteMapping("/api/{codigo}")
public ResponseEntity<?> remover(@PathVariable int codigo){
    return servico.remover(codigo);
}
```

http://127.0.0.1:8080/api/contador

```
@GetMapping("/api/contador")
public Long contador(){
    return acao.count();
}
```

http://127.0.0.1:8080/api/ordenarNomes

```
@GetMapping("/api/ordenarNomes")
public List<Pessoa> ordenarNomes(){
    return acao.findByOrderByNomeDesc();
}
```

http://127.0.0.1:8080/api/ordenarNomes2

```
@GetMapping("/api/ordenarNomes2")
public List<Pessoa> ordenarNomes2(){
    return acao.findByNomeOrderByIdade(nome: "Akilles");
}
```

Salvamos e  
vamos no  
thunder

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request New Request Repositorio.java

New Request Activity Collections Env filter activity

PUT localhost:8080/api 21 mins ago  
GET localhost:8080/api/1 1 hour ago  
GET localhost:8080/api 2 days ago  
GET localhost:8080/api 2 hours ago  
POST localhost:8080/api 2 hours ago  
GET localhost:8080/status 21 hours ago  
POST localhost:8080/pessoa 1 day ago  
DEL localhost:8080/api/2 2 days ago  
POST localhost:8080 3 days ago

DELETE localhost:8080/api/18 Send

Status: Size: Time: Response Headers Cookies Results Docs

Query Parameters parameter value

Send Request Ctrl + Enter Import CUrl Ctrl + U Change Environment Ctrl + E Git Sync Details

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

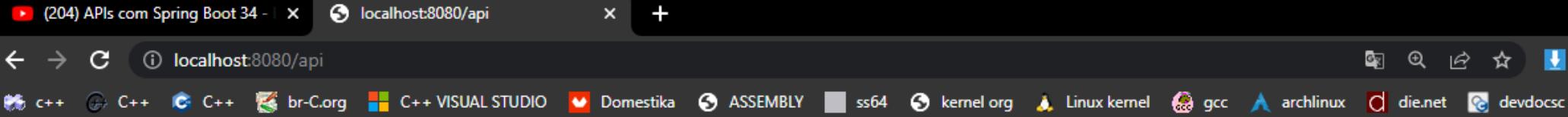
No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

Ao vivo

15:15

Escolhi o codigo 18 que é do Adão



```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":3,"nome":"Carlos","idade":"65"}, {"codigo":4,"nome":"Joyce","idade":"33"}, {"codigo":5,"nome":"Cristiano","idade":"33"}, {"codigo":6,"nome":"Luke","idade":"34"}, {"codigo":7,"nome":"Darth Vader","idade":"41"}, {"codigo":8,"nome":"AnKIN","idade":"9"}, {"codigo":9,"nome":"Obi-Wan","idade":"48"}, {"codigo":10,"nome":"Madara","idade":"100"}, {"codigo":11,"nome":"Naruto","idade":"20"}, {"codigo":12,"nome":"Sasuke","idade":"20"}, {"codigo":13,"nome":"Nanu","idade":"90"}, {"codigo":14,"nome":"Luna","idade":"23"}, {"codigo":15,"nome":"Joao","idade":"34"}, {"codigo":16,"nome":"Edson","idade":"67"}, {"codigo":17,"nome":"Eva","idade":"45"}, {"codigo":18,"nome":"Adao","idade":"700"}, {"codigo":19,"nome":"Moises","idade":"1500"}, {"codigo":20,"nome":"Noe","idade":"100"}, {"codigo":21,"nome":"Maria","idade":"65"}, {"codigo":22,"nome":"Akilles","idade":"65"}, {"codigo":23,"nome":"Prisilla","idade":"77"}]
```

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request New Request Repositorio.java

New Request Activity Collections Env filter activity

DELE localhost:8080/api/18 just now

PUT localhost:8080/api 21 mins ago

GET localhost:8080/api/1 1 hour ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 2 hours ago

POST localhost:8080/api 2 hours ago

GET localhost:8080/status 21 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 days ago

POST localhost:8080 3 days ago

Query Parameters

parameter value

DELETE localhost:8080/api/18 Send

Status: 200 OK Size: 43 Bytes Time: 211 ms

Response Headers 4 Cookies Results Docs

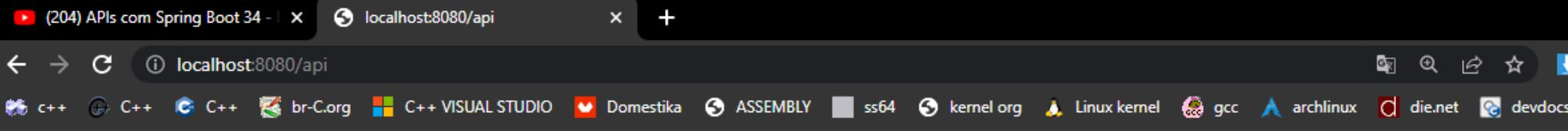
```
1 {  
2   "mensagem": "Pessoa removida com sucesso!"  
3 }
```

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

Ao vivo ^ ☰ 🔍 15:16



```
[{"codigo":1,"nome":"Akilles","idade":"11"}, {"codigo":3,"nome":"Carlos","idade":"65"}, {"codigo":4,"nome":"Joyce","idade":"33"}, {"codigo":5,"nome":"Cristiano","idade":"33"}, {"codigo":6,"nome":"Luke","idade":"34"}, {"codigo":7,"nome":"Darth Vader","idade":"41"}, {"codigo":8,"nome":"AnKIN","idade":"9"}, {"codigo":9,"nome":"Obi-Wan","idade":"48"}, {"codigo":10,"nome":"Madara","idade":"100"}, {"codigo":11,"nome":"Naruto","idade":"20"}, {"codigo":12,"nome":"Sasuke","idade":"20"}, {"codigo":13,"nome":"Nanu","idade":"90"}, {"codigo":14,"nome":"Luna","idade":"23"}, {"codigo":15,"nome":"Joao","idade":"34"}, {"codigo":16,"nome":"Edson","idade":"67"}, {"codigo":17,"nome":"Eva","idade":"45"}, {"codigo":19,"nome":"Moises","idade":"1500"}, {"codigo":20,"nome":"Noe","idade":"100"}, {"codigo":21,"nome":"Maria","idade":"65"}, {"codigo":22,"nome":"Akilles","idade":"65"}, {"codigo":23,"nome":"Prisilla","idade":"77"}]
```

Atualizando o navegador Adão foi removido

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request New Request Repositorio.java

New Request Activity Collections Env filter activity

DEL localhost:8080/api/18 2 mins ago

PUT localhost:8080/api 21 mins ago

GET localhost:8080/api/1 1 hour ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 2 hours ago

POST localhost:8080/api 2 hours ago

GET localhost:8080/status 21 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 days ago

POST localhost:8080 3 days ago

DELETE localhost:8080/api/18

Query Headers 2 Auth Body Tests Pre Run New

Send

Status: 404 Not Found Size: 46 Bytes Time: 20 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "mensagem": "O código informado não existe"  
3 }
```

Se enviar mais uma vez como o código não existe mais recebemos um status de erro

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

30°C Parc ens... 15:18

#36  
Validando por JPA

File Edit Selection View Go Run Terminal Help

New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT ... Servico.java Controle.java New Request Repository.java

New Request

Activity Collections Env filter activity

DEL localhost:8080/api/18 2 mins ago  
PUT localhost:8080/api 21 mins ago  
GET localhost:8080/api/1 1 hour ago  
GET localhost:8080/api 2 days ago  
GET localhost:8080/api 2 hours ago  
POST localhost:8080/api 2 hours ago

DELETE localhost:8080/api/18 Send

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

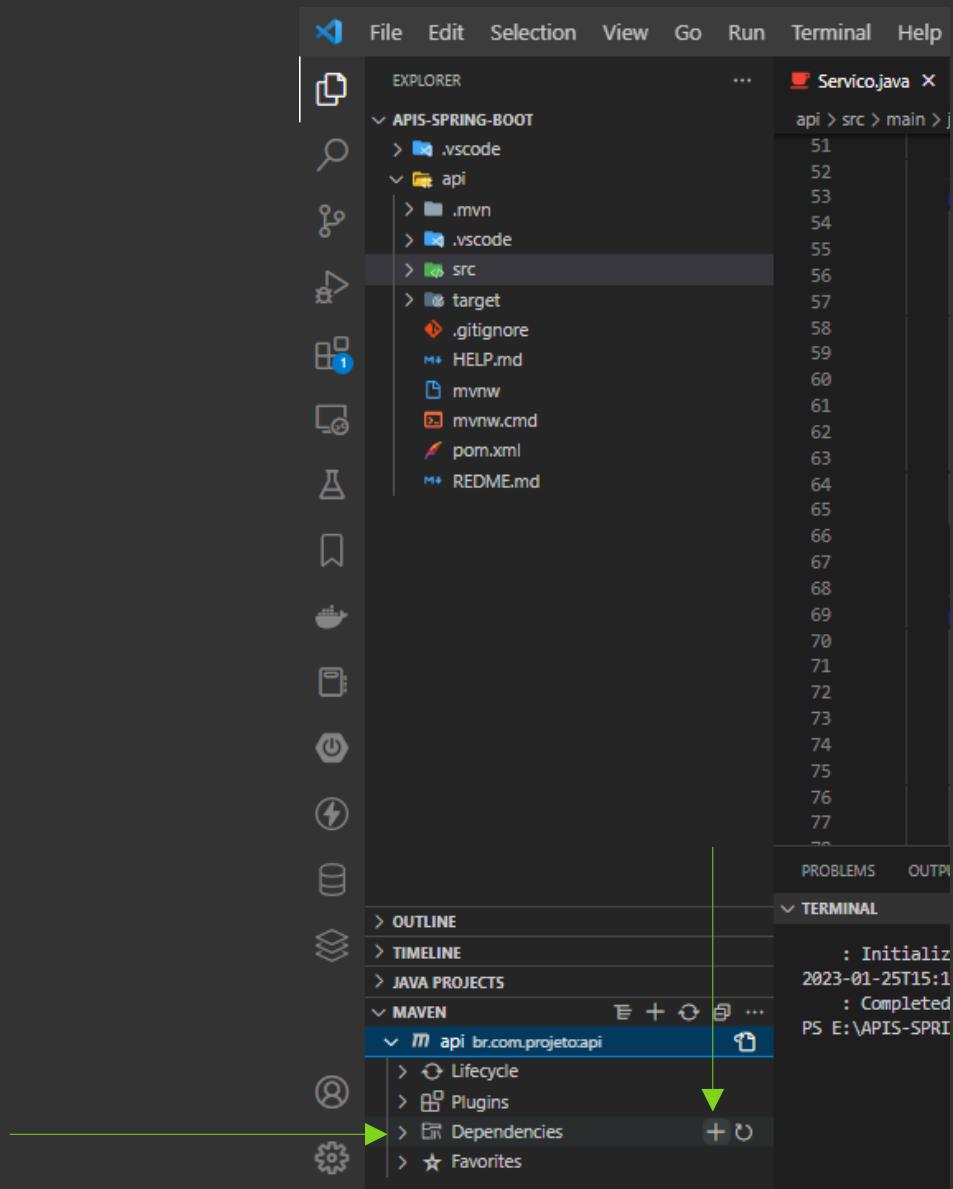
| parameter | value |
|-----------|-------|
|-----------|-------|

Status: 404 Not Found Size: 46 Bytes

Response Headers 4 Cookies R

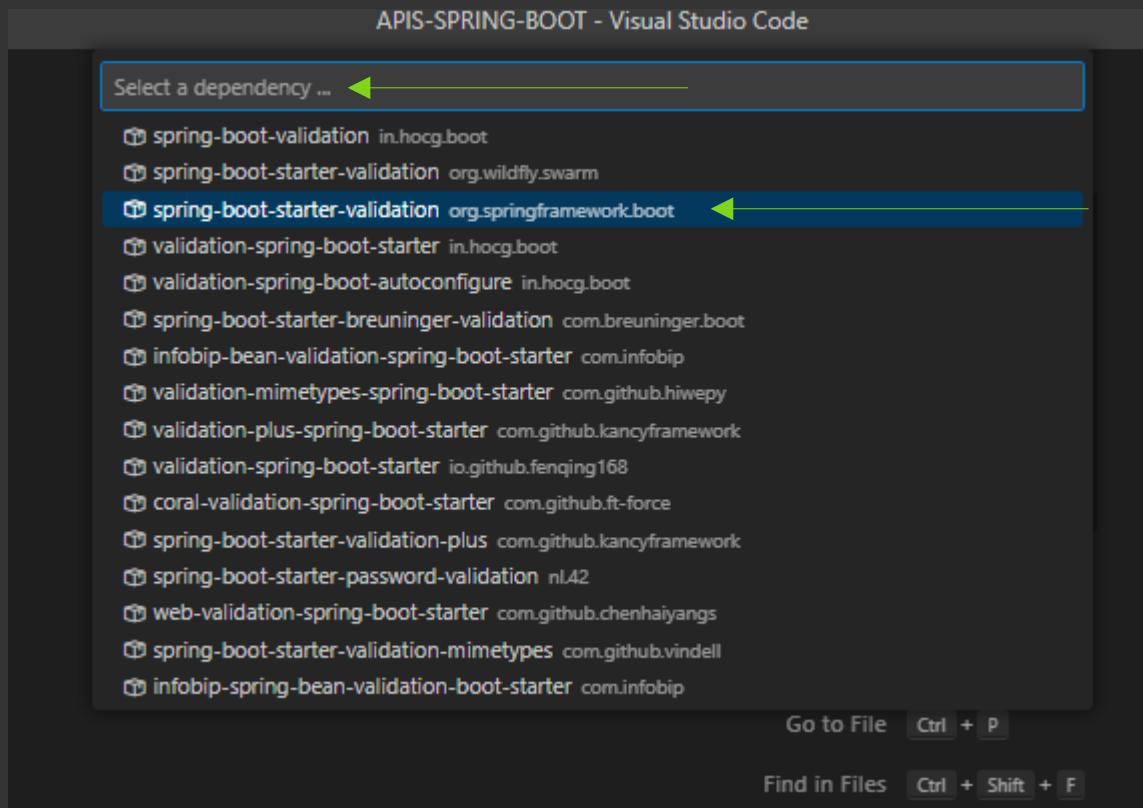
```
1 {  
2   "mensagem": "O código informado não existe."  
3 }
```

FINALIZE O SERVIDOR PORQUE CRIAREMOS UMA NOVA ENTIDADE



CLICK NO  
SINAL DE +

**Digite spring-boot-validation e tecle enter depois escolha  
essa dependencia desse pacote ai e tecle enter**



EXPLORER

APIS-SPRING-BOOT

- .vscode
- src
- main
  - java
    - br
      - com
        - projeto
          - api
            - controle  
Controle.java
            - modelo
            - repositorio  
Repositorio.java
            - servico
            - ApiApplication.java
      - resources
      - test
    - target
      - .gitignore
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
      - REDME.md

pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6       <groupId>org.springframework.boot</groupId>
7       <artifactId>spring-boot-starter-parent</artifactId>
8       <version>3.0.2</version>
9       <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>br.com.projeto</groupId>
12    <artifactId>api</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>api</name>
15    <description>Demo project for Spring Boot</description>
16    <properties>
17      <java.version>17</java.version>
18    </properties>
19    <dependencies>
20
21      <dependency>
22        <groupId>org.springframework.boot</groupId>
23        <artifactId>spring-boot-starter-validation</artifactId>
24        <version>3.0.2</version>
25      </dependency>
26
27    </dependencies>

```

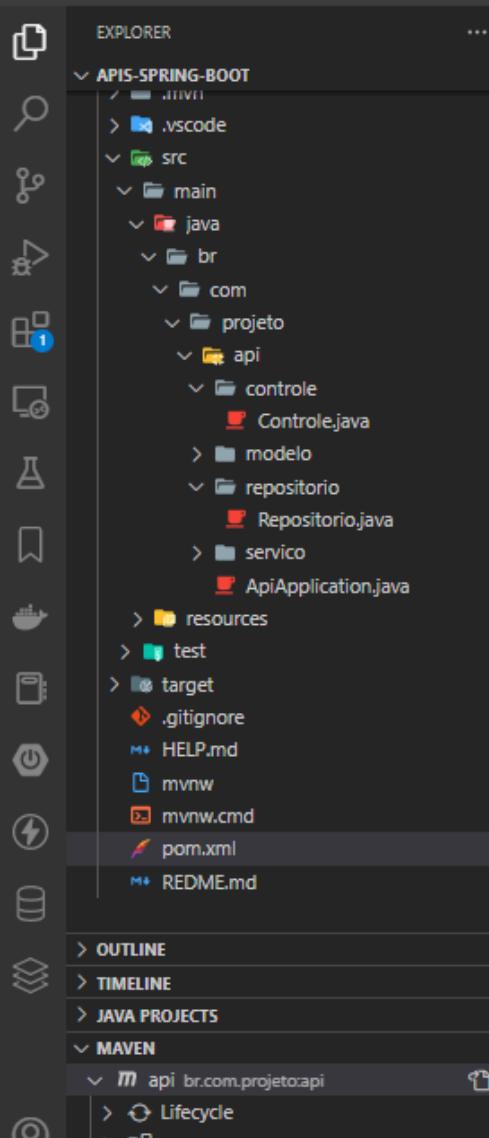
PROBLEMS OUTPUT TERMINAL

TERMINAL

```
: Initializing Servlet 'dispatcherServlet'
2023-01-25T15:16:39.542-03:00  INFO 8972 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet
: Completed initialization in 2 ms
PS E:\APIS-SPRING-BOOT\api>
```

Se tiver que  
confirmar pra  
instalar algo  
confirme e depois  
va no pom.xml ver  
se a dependencia foi  
instalada

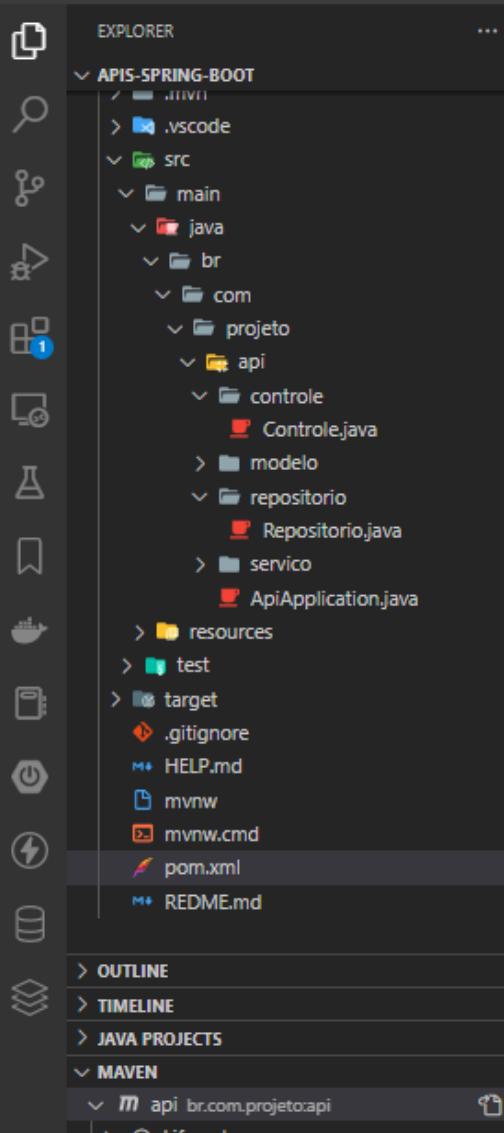
Debug: ApiApplication  
Run: ApiApplication



```
<?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>3.0.2</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>br.com.projeto</groupId>
    <artifactId>api</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>api</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-validation</artifactId>
            <version>3.0.2</version>
        </dependency>
    </dependencies>
</project>
```

Salve antes de fechar mais fique de olho nesse erro

Salve antes de fechar mais fique de olho nesse erro

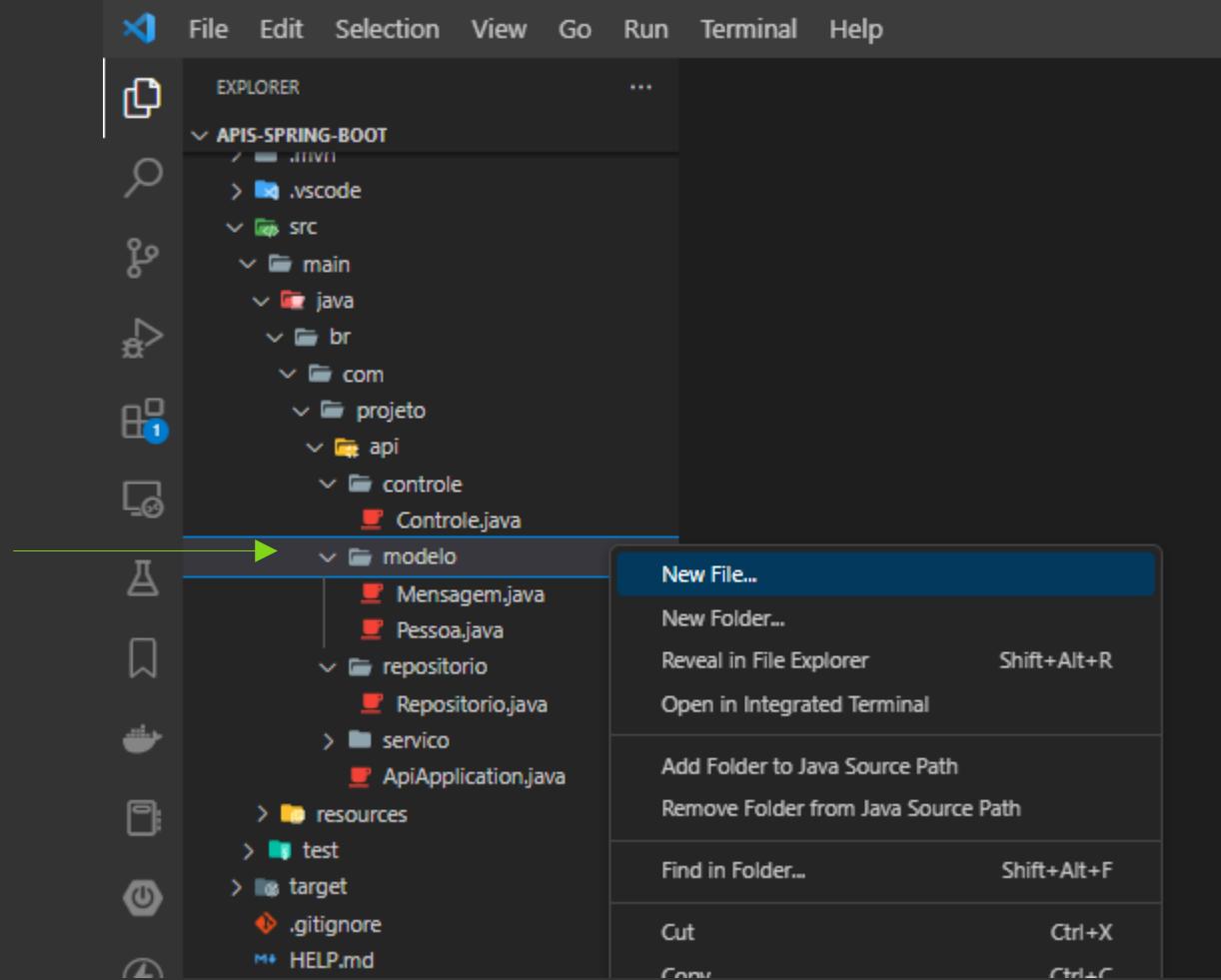


```
api > pom.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/
4      <modelVersion>4.0.0</modelVersion>
5      <parent>
6          <groupId>org.springframework.boot</groupId>
7          <artifactId>spring-boot-starter-parent</artifactId>
8          <version>3.0.2</version>
9          <relativePath/> <!-- lookup parent from repository -->
10     </parent>
11     <groupId>br.com.projeto</groupId>
12     <artifactId>api</artifactId>
13     <version>0.0.1-SNAPSHOT</version>
14     <name>api</name>
15     <description>Demo project for Spring Boot</description>
16     <properties>
17         <java.version>17</java.version>
18     </properties>
19     <dependencies>
20
21         <dependency>
22             <groupId>org.springframework.boot</groupId>
23             <artifactId>spring-boot-starter-validation</artifactId>
24             <version>3.0.2</version>
25         </dependency>
26
27     </dependencies>
```

PROBLEMS

No problems have been detected in the workspace.

Depois que você  
salva ele some



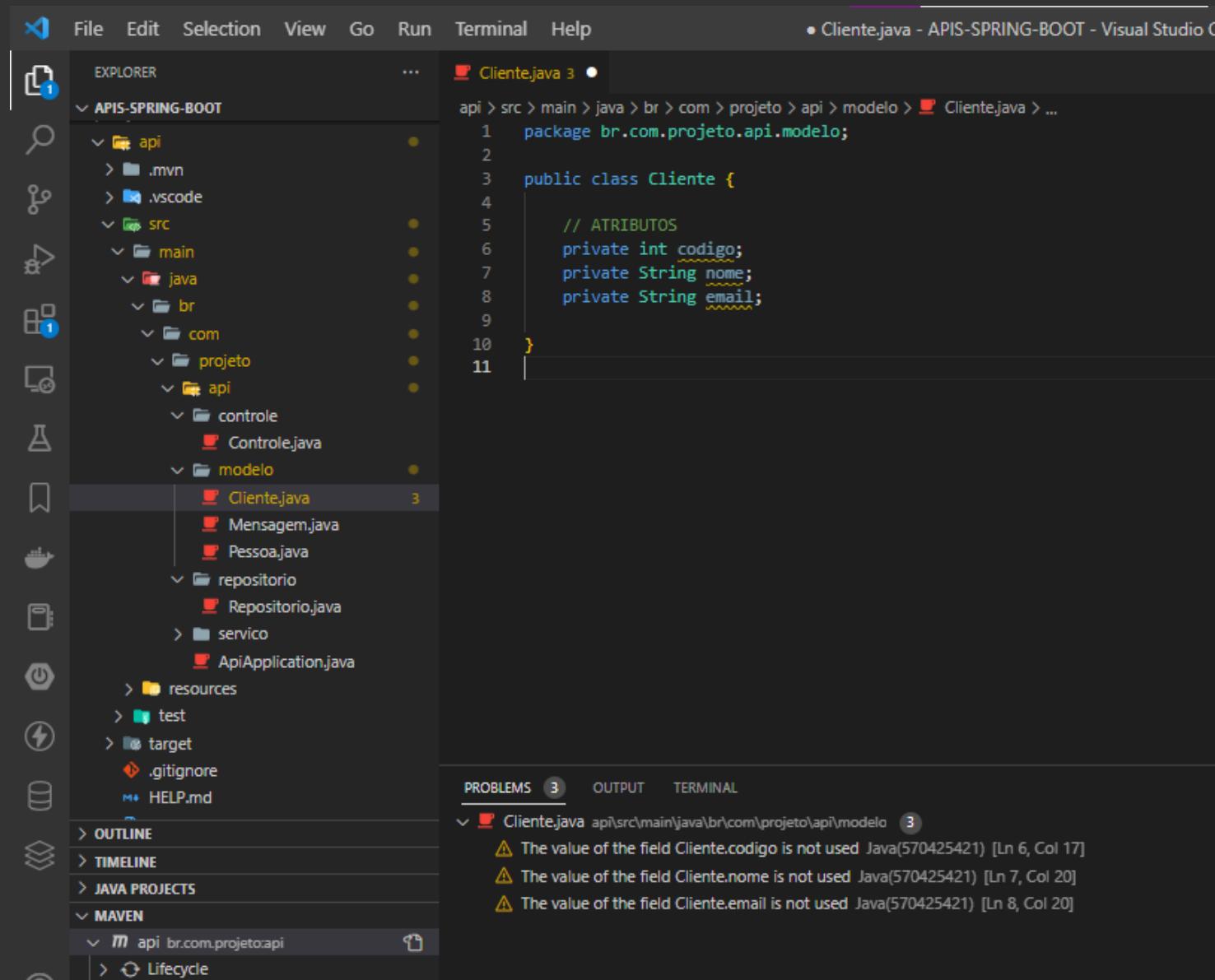
The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Cliente.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT:
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Cliente.java
                  - Mensagem.java
                  - Pessoa.java
                - repositorio
                  - Repositorio.java
                - servico
                  - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
  - Editor View:** The file Cliente.java is open, showing the following code:

```
1 package br.com.projeto.api.modelo;
2
3 public class Cliente {
4
5 }
```

A code completion dropdown is visible on the right, listing options related to the word "Cliente".
  - Bottom Status Bar:** PROBLEMS, OUTPUT, TERMINAL. The PROBLEMS tab is selected, displaying the message: "No problems have been detected in the workspace."

**Criamos o arquivo Cliente.java dentro da pasta modelo**



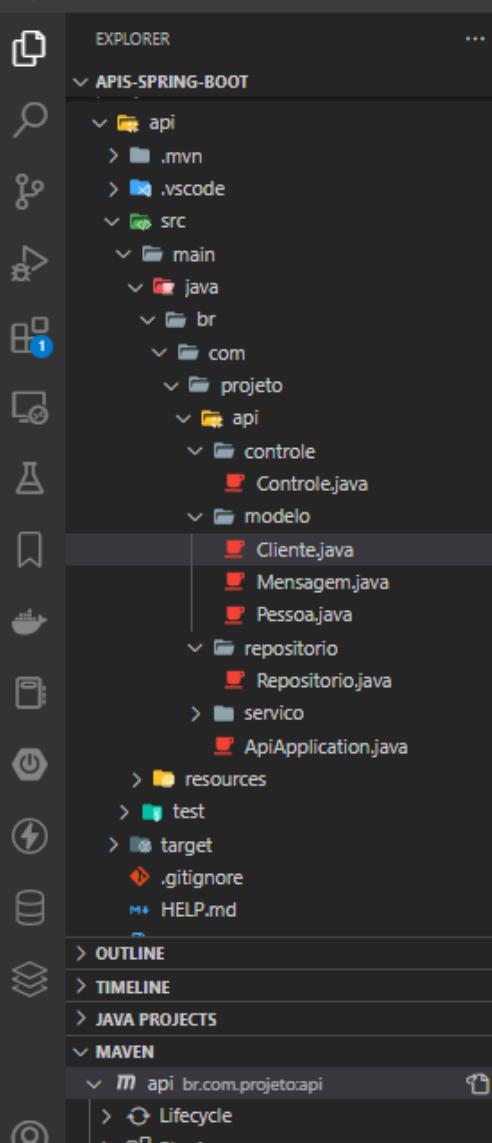
The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Cliente.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under 'APIS-SPRING-BOOT'. The 'src' folder contains 'main', 'java', 'br', 'com', 'projeto', 'api', 'controle', 'modelo', 'repositorio', 'servico', and 'resources' subfolders. 'Cliente.java' is selected in the 'modelo' folder.
- Code Editor:** Displays the 'Cliente.java' code:

```
1 package br.com.projeto.api.modelo;
2
3 public class Cliente {
4     // ATRIBUTOS
5     private int codigo;
```

A yellow lightbulb icon is shown next to the line 'private int codigo;'. A context menu is open over this line, titled 'Quick Fix...', listing several options:
  - Remove 'codigo', keep assignments with side effects
  - Add Javadoc for 'codigo'
  - Generate Getter and Setter for 'codigo'** (This option is highlighted in blue)
  - Generate Getter for 'codigo'
  - Generate Setter for 'codigo'
  - Generate Constructors...
  - Add final modifier for 'codigo'
- Problems View:** Shows three errors related to 'Cliente.java':
  - The value of the field Cliente.codigo is not used [Ln 6, Col 17]
  - The value of the field Cliente.nome is not used [Ln 7, Col 20]
  - The value of the field Cliente.email is not used [Ln 8, Col 20]
- Bottom Navigation:** PROBLEMS (3), OUTPUT, TERMINAL.

**Text Overlay:** SELECIONE A LINHA CLICK NA LAMPADA E CRIE OS METODOS GET E SET



```
1 package br.com.projeto.api.modelo;
2
3 public class Cliente {
4
5     // ATRIBUTOS
6     private int codigo;
7     private String nome;
8     private String email;
9
10    public int getCodigo() {
11        return codigo;
12    }
13    public void setCodigo(int codigo) {
14        this.codigo = codigo;
15    }
16    public String getNome() {
17        return nome;
18    }
19    public void setNome(String nome) {
20        this.nome = nome;
21    }
22    public String getEmail() {
23        return email;
24    }
25    public void setEmail(String email) {
26        this.email = email;
27    }
28}
29}
30}
```

CRIE GET E SET PRA  
TODOS

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

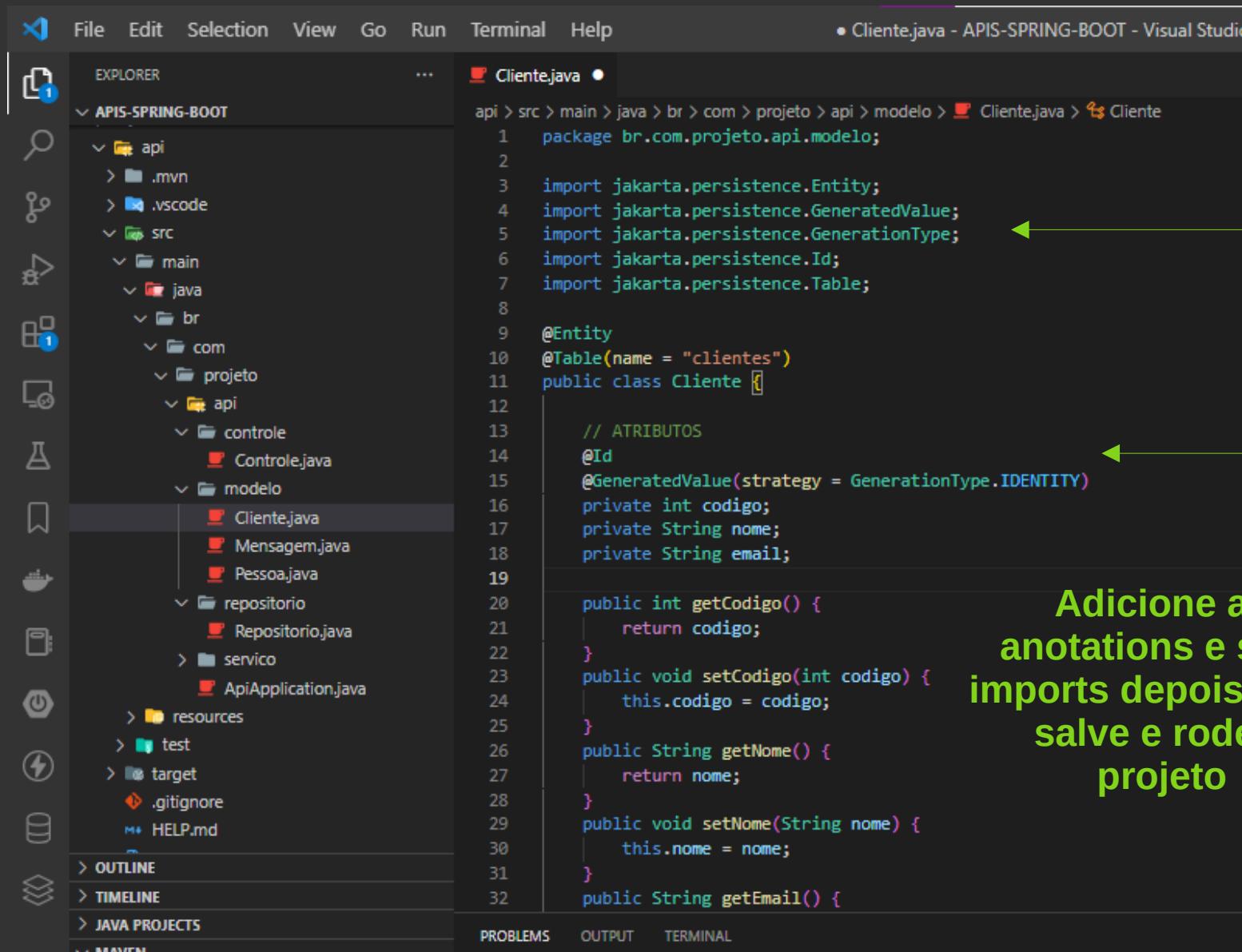
**DEPOIS DE CRIAR OS GET E SET ADICIONAMOS AS ANOTATIONS**

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Cliente.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer:** Shows the project structure for "APIS-SPRING-BOOT".
  - api
  - .mvn
  - .vscode
  - src
    - main
      - java
        - br
          - com
            - projeto
              - api
                - controle
                  - Controle.java
                - modelo
                  - Cliente.java
                  - Mensagem.java
                  - Pessoa.java
                - repositorio
                  - Repositorio.java
                - servico
                  - ApiApplication.java
    - resources
    - test
    - target
    - .gitignore
    - HELP.md
  - OUTLINE
  - TIMELINE
  - JAVA PROJECTS
  - MAVEN
- Code Editor:** Displays the "Cliente.java" file content.

```
package br.com.projeto.api.modelo;
import jakarta.persistence.Entity;
import jakarta.persistence.Table;
@Entity
@Table(name = "clientes")
public class Cliente {
    // ATRIBUTOS
    private int codigo;
    private String nome;
    private String email;
    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```
- Bottom Status Bar:** PROBLEMS, OUTPUT, TERMINAL. No problems have been detected in the workspace.

**Text Overlay:** Adicionando as anotações e seus imports



**Adicione as  
anotações e seus  
imports depois disso  
salve e rode o  
projeto**



SPRING BOOT DASHBOARD



APPS



Cliente.java X

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; modelo &gt; Cliente.java

```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8
9 @Entity
10 @Table(name = "clientes")
11 public class Cliente {
12
13     // ATRIBUTOS
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int codigo;
17     private String nome;
18     private String email;
19
20     public int getCodigo() {
21         return codigo;
22     }
23     public void setCodigo(int codigo) {
24         this.codigo = codigo;
25     }
26     public String getNome() {
27         return nome;
28     }
29     public void setNome(String nome) {
30         this.nome = nome;
31     }
32     public String getEmail() {
```

PROBLEMS

OUTPUT

TERMINAL

No problems have been detected in the workspace.

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** ApiApplication.java - APIS-SPRING-BOOT - Visual Studio Code
- Database Explorer:** Connected to 127.0.0.1@3306 (8.0.25). The schema "api\_spring\_16k" is selected, showing tables: clientes, pessoas (18 rows), columns (with fields: codigo int, idade varchar(255), nome varchar(255)), index, views, procedures, functions, dbagenda, sakila, world.
- Code Editor:** File named ApiApplication.java is open. The code is as follows:

```
1 package br.com.projeto.api;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ApiApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(primarySource: ApiApplication.class, args);
11     }
12 }
13
14 }
```

**Bottom Center Text:** Se for preciso atualize tá ai a tabela criada clientes

**Bottom Navigation:** PROBLEMS, OUTPUT, TERMINAL. The TERMINAL tab is selected.

**Bottom Terminal Output:**

```
2023-01-25T15:47:39.442-03:00 INFO 12912 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-01-25T15:47:39.515-03:00 INFO 12912 --- [ restartedMain] br.com.projeto.api.ApiApplication
```

#37  
Annotations pra validar os dados

File Edit Selection View Go Run Terminal Help

• Cliente.java - APIS-SPRING-BOOT - Visual Studio C

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
- main
- java
- br
- com
- projeto
- api
- controle
- Controle.java
- modelo
- Cliente.java
- Mensagem.java
- Pessoa.java
- repositorio
- Repositorio.java
- servico
- ApiApplication.java
- resources
- test
- target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

... Cliente.java ApiApplication.java

api > src > main > java > br > com > projeto > api > modelo > Cliente.java > Cliente

```
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8 import jakarta.validation.constraints.NotEmpty;
9
10 @Entity
11 @Table(name = "clientes")
12 public class Cliente {
13
14     // ATRIBUTOS
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private int codigo;
18
19     @NotEmpty(message = "Informe um nome")
20     private String nome;
21     private String email;
22
23     public int getCodigo() {
24         return codigo;
25     }
26     public void setCodigo(int codigo) {
27         this.codigo = codigo;
28     }
29     public String getNome() {
30         return nome;
31     }
32 }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Adicione a annotation e seu  
importe ela garante que  
pelo menos um caractere  
seja inserido

File Edit Selection View Go Run Terminal Help

• Cliente.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- < controle
- < Controle.java
- < modelo
- < Cliente.java
- < Mensagem.java
- < Pessoa.java
- < repositorio
- < Repositorio.java
- < servico
- < ApiApplication.java
- > resources
- > test
- > target
- < .gitignore
- < HELP.md
- < mvnw
- < mvnw.cmd
- < pom.xml
- < README.md

... Cliente.java ApiApplication.java

api > src > main > java > br > com > projeto > api > modelo > Cliente.java > Cliente

```
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.Table;
8 import jakarta.validation.constraints.Email;
9 import jakarta.validation.constraints.NotEmpty;
10
11 @Entity
12 @Table(name = "clientes")
13 public class Cliente {
14
15     // ATRIBUTOS
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private int codigo;
19
20     @NotEmpty(message = "Informe um nome")
21     private String nome;
22
23     @Email(message = "Informe um email valido") ←
24     private String email;
25
26     public int getCodigo() {
27         return codigo;
28     }
29     public void setCodigo(int codigo) {
30         this.codigo = codigo;
31     }
}
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Sempre lembre  
dos imports junto  
com as  
annotations

Essa annotation valida se o e-  
mail tem @ e . se tiver ok se  
não ela envia a mensagem  
que colocamos ai nos  
parentesis()



EXPLORER

Cliente.java

Controle.java

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; controle &gt; Controle.java &gt; Language Support for Java(TM) by Red Hat

```
97 }
98 }
99 @GetMapping("/boasVindas")
100 public String boasVindas(){
101     return "Seja bem vindo(a) ";
102 }
103 }
104 @GetMapping("/boasVindas/{nome}")
105 public String boasVindas(@PathVariable String nome){
106     return "Seja bem vindo(a) " + nome;
107 }
108 }
109 @PostMapping("/pessoa")
110 public Pessoa pessoa(@RequestBody Pessoa p){
111     return p;
112 }
113 }
114 }
115 @GetMapping("/status")
116 public ResponseEntity<?> status(){
117     return new ResponseEntity<?>(HttpStatus.CREATED);
118 }
119 }
120 @PostMapping("/cliente")
121 public void cliente(){}
122 }
123 }
124 }
125 }
```

Criamos o metodo e sua rota

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

```
116     public ResponseEntity<?> status(){
117         return new ResponseEntity<>(HttpStatus.CREATED);
118     }
119
120     @PostMapping("/cliente")
121     public void cliente(@RequestBody){?
122     }
123
124 }
125 }
```

The code editor shows a Java controller class with annotations. A yellow question mark icon is placed over the `@RequestBody` annotation in the `cliente` method. A green arrow points from this icon up towards the code, indicating a tooltip or hover information.

PROBLEMS 3 OUTPUT TERMINAL

▼  Controle.java ao\src\main\java\br\com\projeto\api\controle 3

**@RequestBody porque vamos manipular json**

A screenshot of an IDE showing Java code. The code is as follows:

```
17     |         return new ResponseEntity<>(HttpStatus.CREATED);
18     |
19
20     @PostMapping("/cliente")
21     public void cliente(@RequestBody Cliente){}
22
23     }
24
25 }
```

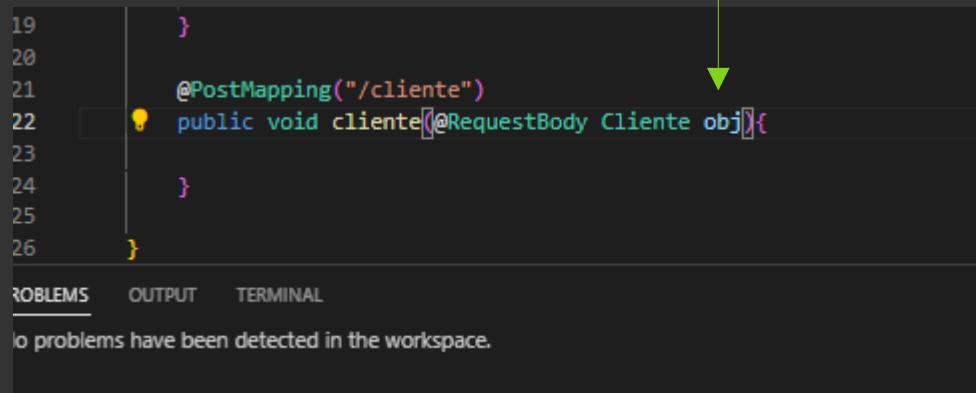
The word `Cliente` is underlined with a red squiggle, indicating a syntax error. A yellow lightbulb icon is positioned above the line `public void cliente(@RequestBody Cliente){}`. A green arrow points from the text "Esse objeto json vai ser do tipo cliente" at the bottom to the `Cliente` parameter in the code. A tooltip is displayed over the `Cliente` parameter, listing various `Cliente` classes from different packages:

- Cliente - br.com.projeto.api.modelo
- ClientEndpoint - jakarta.websocket
- ClientEndpointConfig - jakarta.websocket
- ClientEndpointHolder - org.apache.tomcat.websocket
- ClientAbortException - org.apache.catalina.connector
- ClientDefaultCodecs - org.springframework.http.client
- ClientMessages - com.mysql.cj.x.protobuf.Mysqlx
- ClientMessagesOrBuilder - com.mysql.cj.x.protobuf...
- ClientRequest - org.springframework.boot.actuate...
- ClientRequestObservationContext - org.springframe...
- ClientRequestObservationConvention - org.springfr...
- ClientRequests - org.springframework.boot.actuate...

Below the code editor, there is a "PROBLEMS" tab with three errors:

- Controlador.java api\src\main\java\br\com\projeto\api\controlador: Syntax error, insert ")" to complete ClassBody Java(1610612976)
- Syntax error, insert "... VariableDeclaratorId" to complete FormalParameterList Java(1610612968) [L]
- Syntax error on token ")", delete this token Java(1610612968) [L]

Esse objeto json vai ser do tipo cliente



```
19     }
20
21     @PostMapping("/cliente")
22     public void cliente(@RequestBody Cliente obj){}
23
24     }
25
26 }
```

PROBLEMS    OUTPUT    TERMINAL

No problems have been detected in the workspace.

E o nome dele vai ser obj  
Parametro feito

A screenshot of a Java code editor showing a validation error. The code is as follows:

```
L20     }
L21
L22     @PostMapping("/cliente")
L23     public void cliente(@Valid @RequestBody Cliente obj){
L24
L25     }
L26
L27 }
```

The line L23 contains a validation annotation: `@Valid @RequestBody Cliente obj`. A yellow lightbulb icon is positioned next to the `@Valid` annotation, indicating a problem. A green arrow points upwards from the text "Antes de toda anotação de validação temos que adicionar essa anotação ai @Valid" to this error icon.

PROBLEMS    OUTPUT    TERMINAL

No problems have been detected in the workspace.

**Antes de toda anotação de validação temos que adicionar  
essa anotação ai `@Valid`**



EXPLORER



&gt; .vscode



&lt; API-SPRING-BOOT

&gt; api

&gt; .mvn

&gt; .vscode



&lt; src

&gt; main

&gt; java

&gt; br

&gt; com

&gt; projeto

&gt; api

&gt; controle

Controle.java



&gt; modelo

Cliente.java

Mensagem.java

Pessoa.java

&gt; repositorio

Repository.java

&gt; servico

ApiApplication.java

&gt; resources

&gt; test

&gt; target

.gitignore



HELP.md



mvnw



mvnw.cmd

Cliente.java Controle.java

api &gt; src &gt; main &gt; java &gt; br &gt; com &gt; projeto &gt; api &gt; controle &gt; Controle.java &gt; Language Support for .

```
1 package br.com.projeto.api.controle;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.DeleteMapping;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.PostMapping;
12 import org.springframework.web.bind.annotation.PutMapping;
13 import org.springframework.web.bind.annotation.RequestBody;
14 import org.springframework.web.bind.annotation.RestController;
15
16 import br.com.projeto.api.modelo.Cliente;
17 import br.com.projeto.api.modelo.Pessoa;
18 import br.com.projeto.api.repositorio.Repositorio;
19 import br.com.projeto.api.servico.Servico;
20 import jakarta.validation.Valid;
21
22 @RestController
23 public class Controle {
24
25     @Autowired
26     private Repositorio acao;
27
28     @Autowired
29     private Servico servico;
```

Só pra mostrar os imports até aqui

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

File Edit Selection View Go Run Terminal Help Controle.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER ... Cliente.java Controle.java X

APIS-SPRING-BOOT

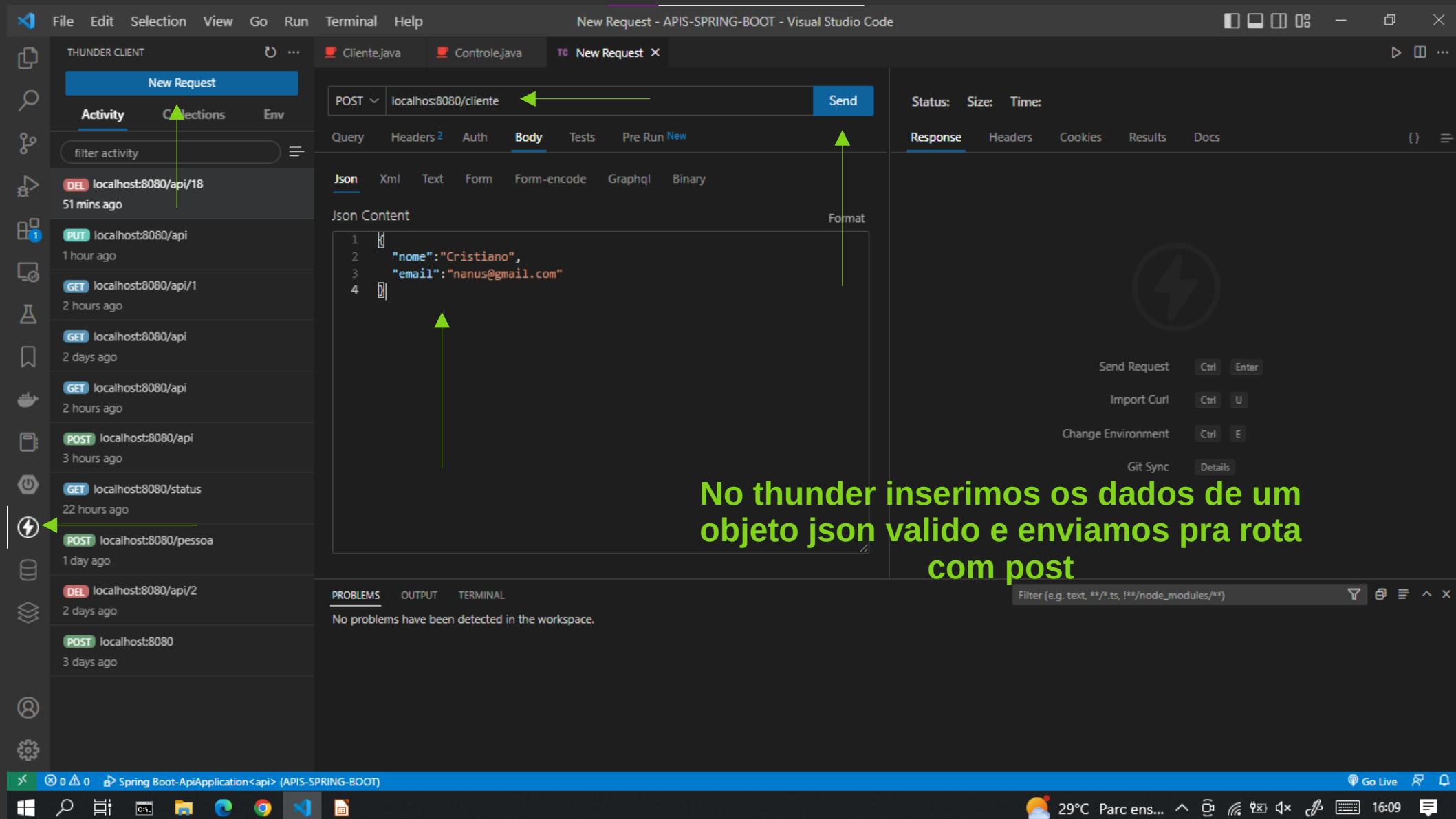
- .vscode
- api
- .mvn
- .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
          - controle
            - Controle.java
  - modelo
    - Cliente.java
    - Mensagem.java
    - Pessoa.java
  - repositorio
    - Repositorio.java
  - servico
  - ApiApplication.java
- resources
- test
- target
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md

```
101     @GetMapping("/boasVindas")
102     public String boasVindas(){
103         return "Seja bem vindo(a) ";
104     }
105
106     @GetMapping("/boasVindas/{nome}")
107     public String boasVindas(@PathVariable String nome){
108         return "Seja bem vindo(a) " + nome;
109     }
110
111     @PostMapping("/pessoa")
112     public Pessoa pessoa(@RequestBody Pessoa p){
113         return p;
114     }
115
116
117     @GetMapping("/status")
118     public ResponseEntity<?> status(){
119         return new ResponseEntity<?>(HttpStatus.CREATED);
120     }
121
122     @PostMapping("/cliente")
123     public void cliente(@Valid @RequestBody Cliente obj){
124
125     }
126
127 }
128 }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Salve e vamos testar



File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request

New Request Activity Collections Env filter activity

POST http://localhost:8080/cliente just now

POST localhost:8080/cliente just now

DEL localhost:8080/api/18 51 mins ago

PUT localhost:8080/api 1 hour ago

GET localhost:8080/api/1 2 hours ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 2 hours ago

POST localhost:8080/api 3 hours ago

GET localhost:8080/status 22 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 days ago

POST localhost:8080 3 days ago

0 ▲ 0 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

POST http://localhost:8080/cliente

Status: ERROR Size: 0 B Time: 0 ms

Response Headers Cookies Results Docs

Json Content Format

```
1 {
2   "nome": "Cristiano",
3   "email": "nanus@gmail.com"
4 }
```

Couldn't resolve the hostname to an IP address, Verify Url:  
<http://localhost:8080/cliente>

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

29°C Parc ens... 16:13

# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Jan 25 16:11:48 BRT 2023

There was an unexpected error (type=Method Not Allowed, status=405).

Method 'GET' is not supported.

```
org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported  
    at  
org.springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.handleNoMatch(RequestMappingInfoHandlerMapping.java:191)  
    at  
org.springframework.web.servlet.handler.AbstractHandlerMethodMapping.lookupHandlerMethod(AbstractHandlerMethodMapping.java:105)  
    at  
org.springframework.web.servlet.handler.AbstractHandlerMethodMapping.getHandlerInternal(AbstractHandlerMethodMapping.java:87)  
    at  
org.springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.getHandlerInternal(RequestMappingInfoHandlerMapping.java:125)  
    at  
org.springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.getHandlerInternal(RequestMappingInfoHandlerMapping.java:125)
```

**Resolvendo o erro só esperei e pronto**

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request X

New Request

Activity Collections Env

filter activity

POST localhost:8080/cliente just now

POST http://localhost:8080/cliente just now

POST localhost:8080/cliente just now

DEL localhost:8080/api/18 51 mins ago

PUT localhost:8080/api 1 hour ago

GET localhost:8080/api/1 2 hours ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 2 hours ago

POST localhost:8080/api 3 hours ago

GET localhost:8080/status 22 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 days ago

0 0 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

POST localhost:8080/cliente Send

Status: 200 OK Size: 0 Bytes Time: 293 ms

Response Headers 3 Cookies Results Docs

1

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "email": "nanus@gmail.com"  
4 }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

29°C Parc ens... 16:17

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request

New Request Activity Collections Env filter activity

POST localhost:8080/cliente Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary Format

Json Content

```
1 {  
2   "nome": "", ←  
3   "email": "nanus@gmail.com"  
4 }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Status: 400 Bad Request Size: 6.19 KB Time: 71 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "timestamp": "2023-01-25T19:19:37.217+00:00",  
3   "status": 400,  
4   "error": "Bad Request",  
5   "trace": "org.springframework.web.bind.MethodArgumentNotValidException:  
Validation failed for argument [0] in public void br.com.projeto.api.  
.controle.Controle.cliente(br.com.projeto.api.modelo.Cliente):  
[Field error in object 'cliente' on field 'nome': rejected value [];  
codes [NotEmpty.cliente.nome,NotEmpty.nome,NotEmpty.java.lang.String  
,NotEmpty]; arguments [org.springframework.context.support.  
.DefaultMessageSourceResolvable: codes [cliente.nome,nome];  
arguments []]; default message [none]; default message [Informe um  
nome]] \r\n\tat org.springframework.web.servlet.mvc.method  
.annotation.RequestMappingMethodProcessor.resolveArgument  
(RequestMappingMethodProcessor.java:144)\r\n\tat org  
.springframework.web.method.support.  
.HandlerMethodArgumentResolverComposite.resolveArgument  
(HandlerMethodArgumentResolverComposite.java:122)\r\n\tat org  
.springframework.web.method.support.InvocableHandlerMethod  
.getMethodArgumentValues(InvocableHandlerMethod.java:181)\r\n\tat org  
.springframework.web.method.support.InvocableHandlerMethod  
.invokeForRequest(InvocableHandlerMethod.java:148)\r\n\tat org  
.springframework.web.servlet.mvc.method.annotation  
.ServletInvocableHandlerMethod.invokeAndHandle  
(ServletInvocableHandlerMethod.java:117)\r\n\tat org.springframework
```

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Se não passamos o nome o status é um  
**BAD\_REQUEST**

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live 29°C Parc ens... 16:19

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request

New Request Activity Collections Env filter activity

POST localhost:8080/cliente Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "email": "nanus"  
4 }
```

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Status: 400 Bad Request Size: 6.32 KB Time: 24 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "timestamp": "2023-01-25T19:21:30.779+00:00",  
3   "status": 400,  
4   "error": "Bad Request",  
5   "trace": "org.springframework.web.bind.MethodArgumentNotValidException:  
Validation failed for argument [0] in public void br.com.projeto.api.  
.controle.Controle.cliente(br.com.projeto.api.modelo.Cliente):  
[Field error in object 'cliente' on field 'email': rejected value  
[nanus]; codes [Email.cliente.email,Email.email,Email.java.lang.  
.String,Email]; arguments [org.springframework.context.support.  
.DefaultMessageSourceResolvable: codes [cliente.email,email];  
arguments []; default message [email],[Ljakarta.validation.  
.constraints.Pattern$Flag:@bc8c1b7,.*]; default message [Informe um  
email valido]] \r\n\tat org.springframework.web.servlet.mvc.method.  
.annotation.RequestResponseBodyMethodProcessor.resolveArgument  
(RequestResponseBodyMethodProcessor.java:144)\r\n\tat org.  
.springframework.web.method.support.  
.HandlerMethodArgumentResolverComposite.resolveArgument  
(HandlerMethodArgumentResolverComposite.java:122)\r\n\tat org.  
.springframework.web.method.support.InvocableHandlerMethod.  
.getMethodArgumentValues(InvocableHandlerMethod.java:181)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.  
.invokeForRequest(InvocableHandlerMethod.java:148)\r\n\tat org.  
.springframework.web.servlet.mvc.method.annotation.  
.ServletInvocableHandlerMethod.invokeAndHandle
```

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

AGORA PASSAMOS UM EMAIL INVALIDO NOVAMENTE UM BAD\_REQUEST

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request

New Request Activity Collections Env filter activity

POST localhost:8080/cliente Send

Status: 200 OK Size: 0 Bytes Time: 11 ms

Response Headers 3 Cookies Results Docs

1

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "email": "nanus@gmail.com"  
4 }
```

PROBLEMS OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

No problems have been detected in the workspace.

# VOLTANDO OS DADOS TEMOS A MENSAGEM CERTA

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

#38  
**ResponseEntityExceptionHandler**

7 - MODELO

8 - VINCULANDO CONTROLE E MODELO

9 - TESTANDO API

10 - CONFIGURANDO BANCO DE DADOS

11 - DEPENDÊNCIAS

12 - GERANDO TABELAS

13 - CONEXÃO COM MYSQL

14 - REPOSITÓRIO

15 - @AUTOWIRED

16 - CADASTRAR

17 - SELECIONAR

18 - FILTRAR

```
import java.util.HashMap;
import java.util.Map;

// Importar http, validation, bind, context e servlet
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.FieldError;
import org.springframework.web.bind.MethodArgumentNotValidException;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.context.request.WebRequest;
import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;

@ControllerAdvice
public class Validacao extends ResponseEntityExceptionHandler {

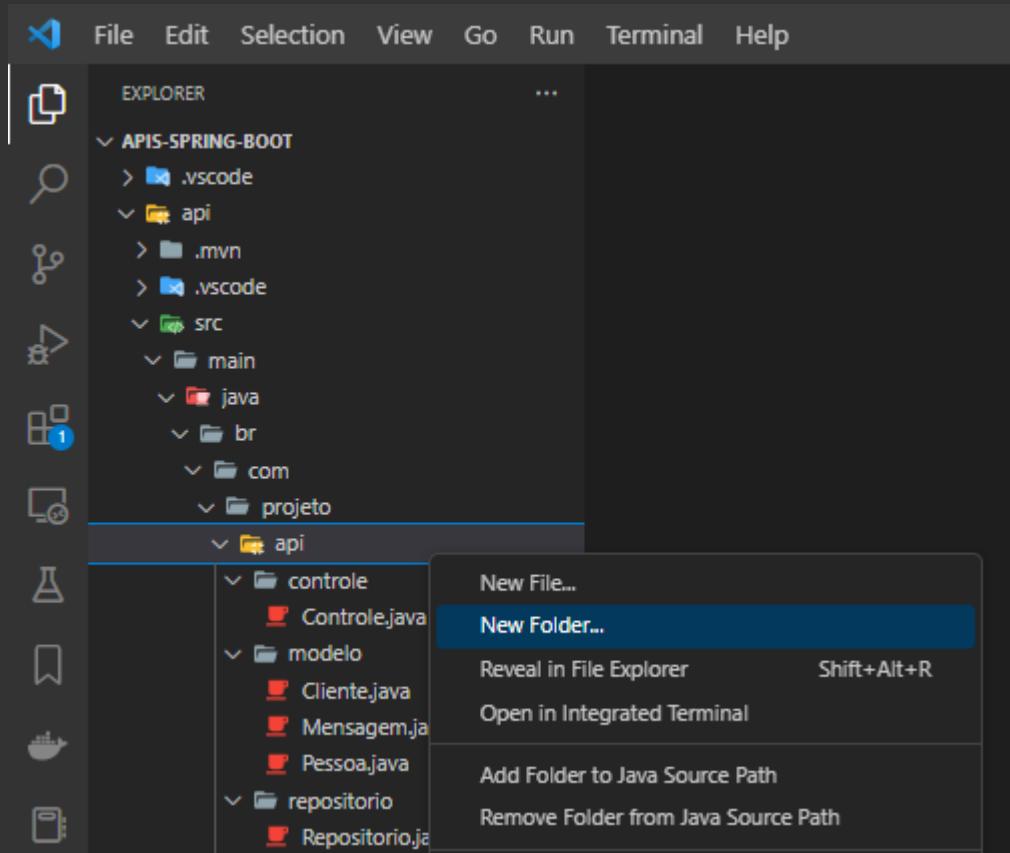
    @Override
    protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex,
            HttpHeaders headers, HttpStatus status, WebRequest request) {

        Map<String, String> errors = new HashMap<>();
        ex.getBindingResult().getAllErrors().forEach((error) -> {

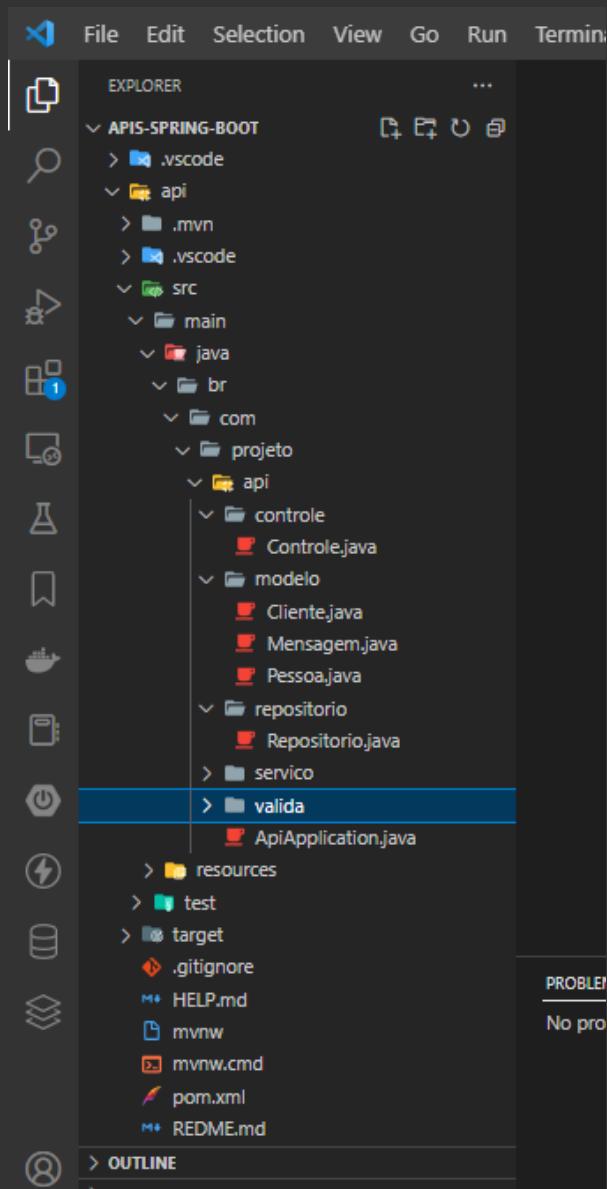
            String fieldName = ((FieldError) error).getField();
            String message = error.getDefaultMessage();
            errors.put(fieldName, message);
        });
    }
}
```

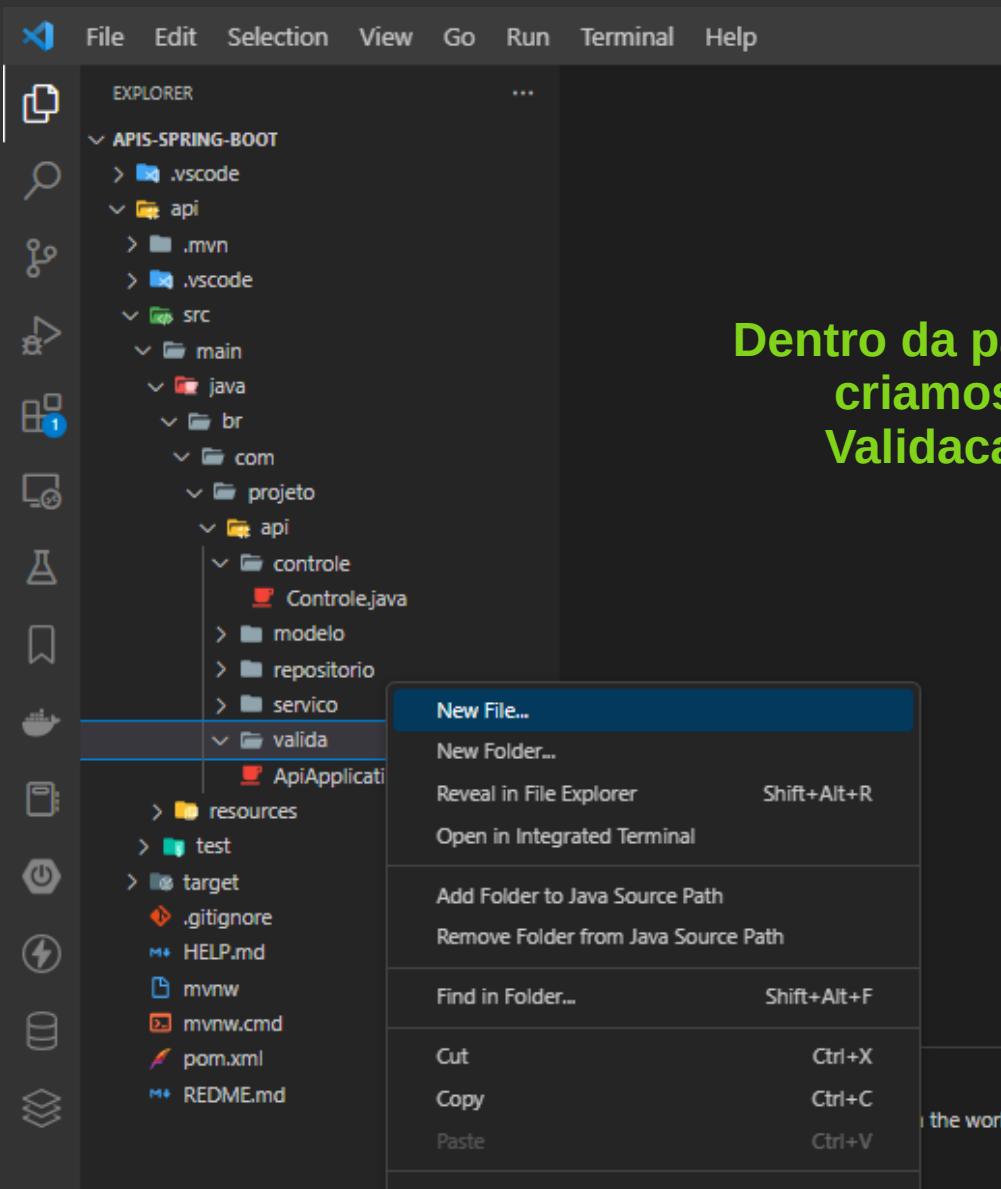


Vamos até esse link  
e copiamos esse  
código



**Criamos uma nova pasta dentro de api com o nome de valida**





Dentro da pasta valida  
criamos o file  
**Validacao.java**

The screenshot shows the Visual Studio Code interface with a Java project named "APIS-SPRING-BOOT". The "EXPLORER" sidebar on the left displays the project structure, including ".vscode", ".mvn", ".vscode", "src" (with "main", "java", "br", "com", "projeto", "api", "controle", "modelo", "repositorio", "servico", "valida"), "resources", "test", "target", ".gitignore", "HELP.md", "mvnw", "mvnw.cmd", "pom.xml", and "README.md". The "VALIDACAO.java" file is open in the main editor area. The code completion dropdown is visible, showing suggestions for the class "Validacao":

```
1 package br.com.projeto.api.valida;
2
3 public class Validacao {
4     |
5 }
6
```

Suggestions in the dropdown:

- class Validacao
- interface Validacao
- enum Validacao
- record Validacao()
- abstract class Validacao
- @interface Validacao

The status bar at the bottom shows "PROBLEMS", "OUTPUT", and "TERMINAL", with the message "No problems have been detected in the workspace."

File Edit Selection View Go Run Terminal Help

• Validacao.java - APIS-SPRING-BOOT

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
  - .mvn
  - .vscode
- src
  - main
    - java
      - br
        - com
          - projeto
            - api
              - controle
                - Controle.java
              - modelo
              - repositorio
              - servico
            - valida
              - Validacao.java
              - ApiApplication.java
  - resources
  - test
  - target
  - .gitignore
  - HELP.md
  - mvnw

Validacao.java

```
1 package br.com.projeto.api.valida;
2
3 public class Validacao {
4
5 }
```

Remova isso só deixe o pacote e  
cole o que estava na pagina

File Edit Selection View Go Run Terminal Help Validacao.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER API-SPRING-BOOT ... Validacao.java 3 localhost:8080/cliente

api > src > main > java > br > com > projeto > api > valida > Validacao.java > ...

```
1 package br.com.projeto.api.valida;
2 // Importar Map e HashMap
3 import java.util.HashMap;
4 import java.util.Map;
5
6 // Importar http, validation, bind, context e servlet
7 import org.springframework.http.HttpHeaders;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.validation.FieldError;
11 import org.springframework.web.bind.MethodArgumentNotValidException;
12 import org.springframework.web.bind.annotation.ControllerAdvice;
13 import org.springframework.web.context.request.WebRequest;
14 import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;
15
16 @ControllerAdvice
17 public class Validacao extends ResponseEntityExceptionHandler{
18
19     @Override
20     protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex, org.springframework.web.bind.support.WebInputBinderError error) {
21         Map<String, String> errors = new HashMap<>();
22         ex.getBindingResult().getAllErrors().forEach(error) ->{
23
24             String fieldName = ((FieldError) error).getField();
25             String message = error.getDefaultMessage();
26             errors.put(fieldName, message);
27         });
28         return new ResponseEntity<Object>(errors, HttpStatus.BAD_REQUEST);
29     }
30 }
```

PROBLEMS 3 OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Validacao.java api\src\main\java\br\com\projeto\api\valida 3

- The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
- The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
- The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

LN 7, COL 45 (36 selected) Spaces: 4 UTF-8 CRLF { Java Go Live

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Windows Search Task View Chrome VS Code File Explorer View Timeline Java Projects Maven

28°C Parc ens... 16:47

PROBLEMS 3 OUTPUT TERMINAL

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

- ⚠ The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
  - ⚠ The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
  - ⚠ The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

File Edit Selection View Go Run Terminal Help Validacao.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER API-SPRING-BOOT ... Validacao.java 3 localhost:8080/cliente

api > src > main > java > br > com > projeto > api > valida > Validacao.java > ...

```
3 import java.util.HashMap;
4 import java.util.Map;
5
6 // Importar http, validation, bind, context e servlet
7 import org.springframework.http.HttpHeaders;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.validation.FieldError;
11 import org.springframework.web.bind.MethodArgumentNotValidException;
12 import org.springframework.web.bind.annotation.ControllerAdvice;
13 import org.springframework.web.context.request.WebRequest;
14 import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;
15
16 @ControllerAdvice
17 public class Validacao extends ResponseEntityExceptionHandler{
18
19     @Override
20     protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex, org.springframework.http.HttpHeaders headers, HttpStatus status, WebRequest request) {
21         Map<String, String> errors = new HashMap<>();
22         ex.getBindingResult().getAllErrors().forEach((error) ->{
23
24             String fieldName = ((FieldError) error).getField();
25             String message = error.getDefaultMessage();
26             errors.put(fieldName, message);
27         });
28         return new ResponseEntity<Object>(errors, HttpStatus.BAD_REQUEST);
29     }
30 }
31 }
```

PROBLEMS 3 OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Validacao.java api\src\main\java\br\com\projeto\api\valida 3

- The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
- The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
- The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

LN 7, COL 45 (36 selected) Spaces: 4 UTF-8 CRLF {} Java Go Live

0 3 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

File Explorer Timeline Java Projects MAVEN

Windows Taskbar

Cloud 28°C Parc ens... 16:48

```
@Override  
protected ResponseEntity<Object>  
handleMethodArgumentNotValid(org.springframework.web.bind.MethodArgumentNotValidException ex,  
org.springframework.http.HttpHeaders headers,  
org.springframework.http.HttpStatusCode status,  
org.springframework.web.context.request.WebRequest request){
```

File Edit Selection View Go Run Terminal Help Validacao.java - APIS-SPRING-BOOT - Visual Studio Code

EXPLORER API-SPRING-BOOT

Validacao.java 3 localhost:8080/cliente

```
3 import java.util.HashMap;
4 import java.util.Map;
5
6 // Importar http, validation, bind, context e servlet
7 import org.springframework.http.HttpHeaders;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.validation.FieldError;
11 import org.springframework.web.bind.MethodArgumentNotValidException;
12 import org.springframework.web.bind.annotation.ControllerAdvice;
13 import org.springframework.web.context.request.WebRequest;
14 import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;
15
16 @ControllerAdvice
17 public class Validacao extends ResponseEntityExceptionHandler{
18
19     @Override
20     protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex, HttpStatus status) {
21         Map<String, String> errors = new HashMap<>();
22         ex.getBindingResult().getAllErrors().forEach((error) ->{
23
24             String fieldName = ((FieldError) error).getField();
25             String message = error.getDefaultMessage();
26             errors.put(fieldName, message);
27         });
28         return new ResponseEntity<Object>(errors, HttpStatus.BAD_REQUEST);
29     }
30 }
31 }
```

PROBLEMS 3 OUTPUT TERMINAL

Validacao.java api\src\main\java\br\com\projeto\api\valida 3

- The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
- The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
- The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Salvamos e vamos no thunder testar essas não são erros são avisos de warning

Ln 21, Col 8 (306 selected) Spaces: 4 UTF-8 CRLF { Java Go Live

0 3 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Windows Search Task View Chrome Microsoft Edge File Explorer

28°C Parc ens... 16:48

File Edit Selection View Go Run Terminal Help localhost:8080/cliente - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Validacao.java 3 localhost:8080/cliente

New Request

Activity Collections Env

filter activity

POST localhost:8080/cliente 2 mins ago

POST http://localhost:8080/cliente 37 mins ago

POST localhost:8080/cliente 39 mins ago

DEL localhost:8080/api/18 2 hours ago

PUT localhost:8080/api 2 hours ago

GET localhost:8080/api/1 3 hours ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 3 hours ago

POST localhost:8080/api 4 hours ago

GET localhost:8080/status 23 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 hours ago

File Edit Selection View Go Run Terminal Help localhost:8080/cliente - APIS-SPRING-BOOT - Visual Studio Code

localhost:8080/cliente - APIS-SPRING-BOOT - Visual Studio Code

POST localhost:8080/cliente Send

Status: 400 Bad Request Size: 26 Bytes Time: 42 ms

Response Headers 4 Cookies Results Docs

1 {  
2 "nome": "Informe um nome"  
3 }

Format

Json Content

1 {  
2 "nome": "",  
3 "email": "nanus@gmail.com"  
4 }

PROBLEMS 3 OUTPUT TERMINAL

Validacao.java api\src\main\java\br\com\projeto\api\valida 3

- The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
- The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
- The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

28°C Parc ens... 16:49

File Edit Selection View Go Run Terminal Help localhost:8080/cliente - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Validacao.java 3 localhost:8080/cliente

New Request

Activity Collections Env filter activity

POST localhost:8080/cliente just now

POST http://localhost:8080/cliente 37 mins ago

POST localhost:8080/cliente 39 mins ago

DEL localhost:8080/api/18 2 hours ago

PUT localhost:8080/api 2 hours ago

GET localhost:8080/api/1 3 hours ago

GET localhost:8080/api 2 days ago

GET localhost:8080/api 3 hours ago

POST localhost:8080/api 4 hours ago

GET localhost:8080/status 23 hours ago

POST localhost:8080/pessoa 1 day ago

DEL localhost:8080/api/2 2 hours ago

0 ▲ 3 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

POST localhost:8080/cliente Send

Status: 400 Bad Request Size: 60 Bytes Time: 16 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "nome": "Informe um nome",  
3   "email": "Informe um email valido"  
4 }
```

Format

Json Content

```
1 {  
2   "nome": "",  
3   "email": "nanu"  
4 }
```

PROBLEMS 3 OUTPUT TERMINAL Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Validacao.java api\src\main\java\br\com\projeto\api\valida 3

- The import org.springframework.http.HttpHeaders is never used Java(268435844) [Ln 7, Col 9]
- The import org.springframework.web.bind.MethodArgumentNotValidException is never used Java(268435844) [Ln 11, Col 9]
- The import org.springframework.web.context.request.WebRequest is never used Java(268435844) [Ln 13, Col 9]

28°C Parc ens... 16:51

File Edit Selection View Go Run Terminal Help New Request - APIS-SPRING-BOOT - Visual Studio Code

THUNDER CLIENT Cliente.java Controle.java New Request

New Request Activity Collections Env filter activity

POST localhost:8080/cliente Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode Graphql Binary

Json Content Format

```
1 {  
2   "nome": "Cristiano",  
3   "email": "nanus"  
4 }
```

O código anterior é só pra evitar que a mensagem aparece assim

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Status: 400 Bad Request Size: 6.32 KB Time: 24 ms

Response Headers 4 Cookies Results Docs

```
1 {  
2   "timestamp": "2023-01-25T19:21:30.779+00:00",  
3   "status": 400,  
4   "error": "Bad Request",  
5   "trace": "org.springframework.web.bind.MethodArgumentNotValidException:  
Validation failed for argument [0] in public void br.com.projeto.api.  
.controle.Controle.cliente(br.com.projeto.api.modelo.Cliente):  
[Field error in object 'cliente' on field 'email': rejected value  
[nanus]; codes [Email.cliente.email,Email.email,Email.java.lang.  
.String,Email]; arguments [org.springframework.context.support.  
.DefaultMessageSourceResolvable: codes [cliente.email,email];  
arguments []; default message [email],[Ljakarta.validation.  
.constraints.Pattern$Flag:@bc8c1b7,.*]; default message [Informe um  
email valido]] \r\n\tat org.springframework.web.servlet.mvc.method.  
.annotation.RequestResponseBodyMethodProcessor.resolveArgument  
(RequestResponseBodyMethodProcessor.java:144)\r\n\tat org.  
.springframework.web.method.support.  
.HandlerMethodArgumentResolverComposite.resolveArgument  
(HandlerMethodArgumentResolverComposite.java:122)\r\n\tat org.  
.springframework.web.method.support.InvocableHandlerMethod.  
.getMethodArgumentValues(InvocableHandlerMethod.java:181)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.  
.invokeForRequest(InvocableHandlerMethod.java:148)\r\n\tat org.  
.springframework.web.servlet.mvc.method.annotation.  
.ServletInvocableHandlerMethod.invokeAndHandle
```

Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT) Go Live

#####  
**FIM 2023**  
#####

**NANO 2023**

