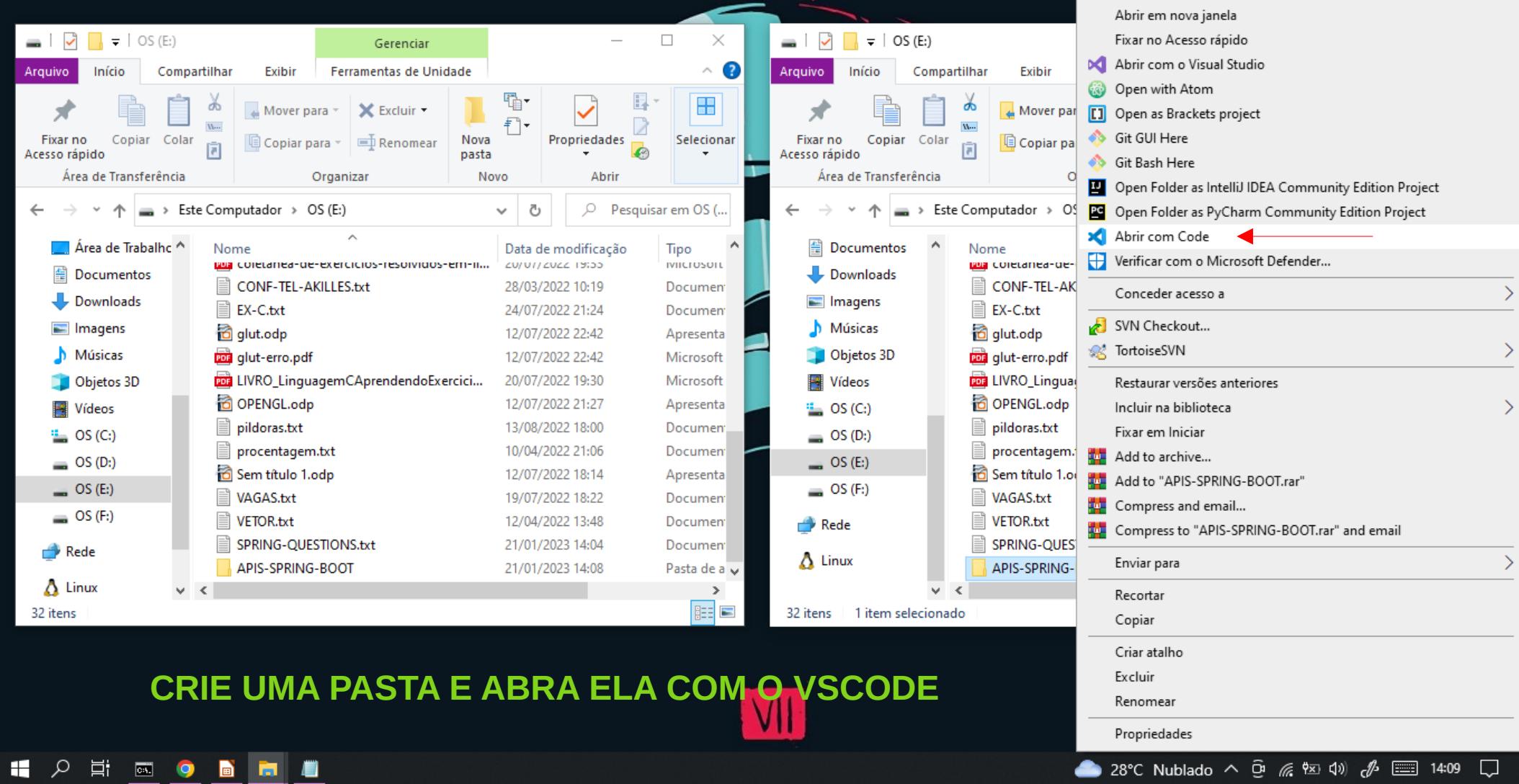
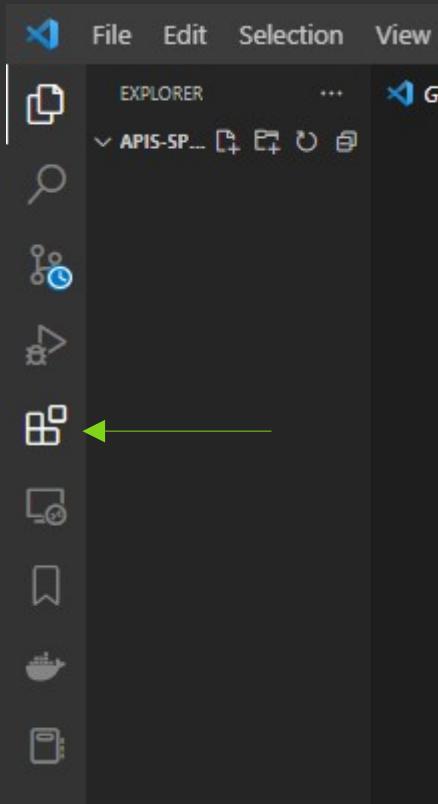


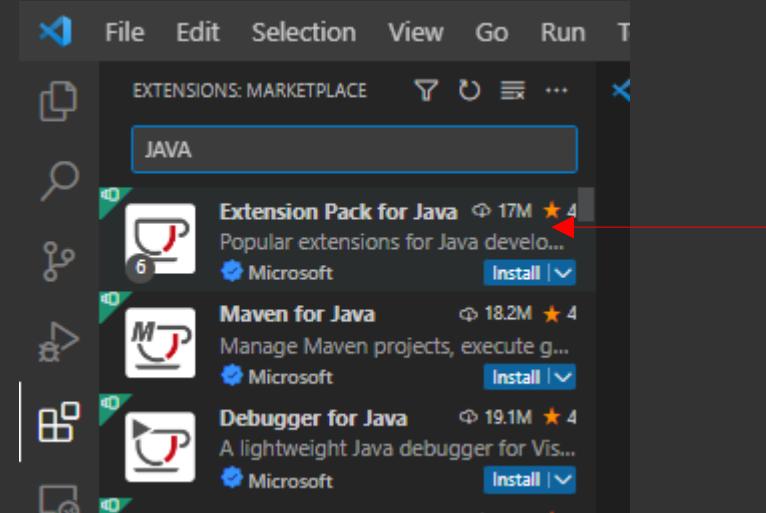
**JAVA SPRING BOOT
2023
NANO**

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
Red Hat [Install](#) | [View details](#)
- Spring Initializr Java** ⚡ 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 out)

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
Red Hat

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



Debugger for Java

A lightweight Java debugger for Visual Studio Code
Microsoft



Maven for Java

Manage Maven projects, execute goals, generate reports
Microsoft

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

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 | Format

Extension Packs

Extension Resources

Marketplace

Repository

License

Microsoft

More Info

Published 9/27/2017, 06:30 AM

Last released 11/29/2022, 02:20 PM

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**

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`)

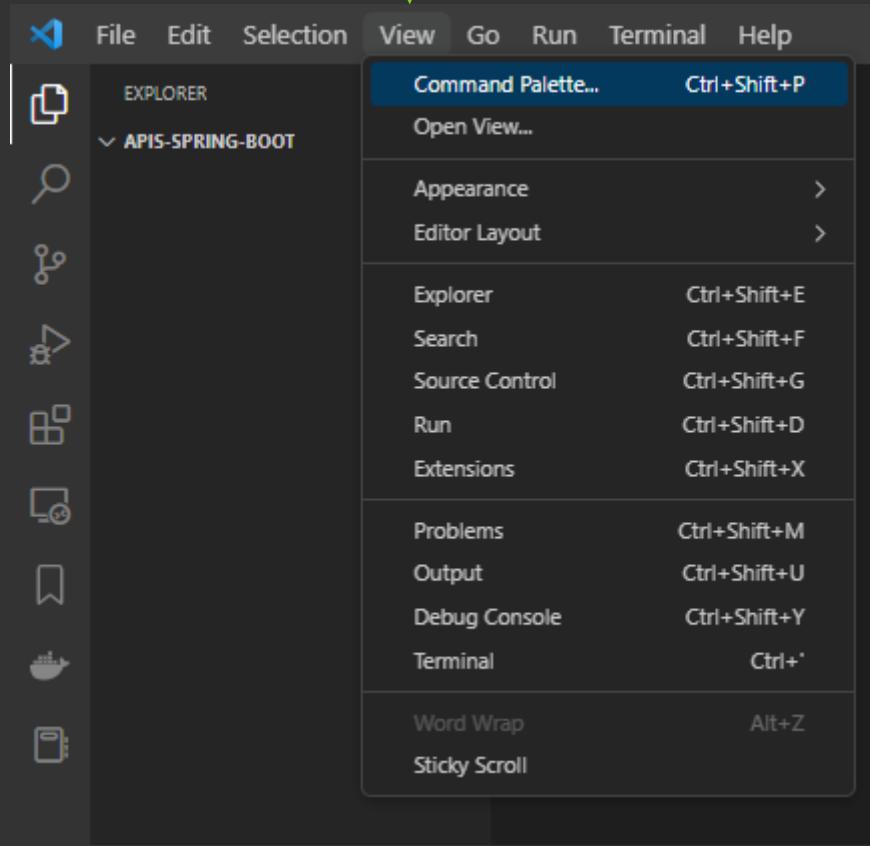
Spring Initializr Java

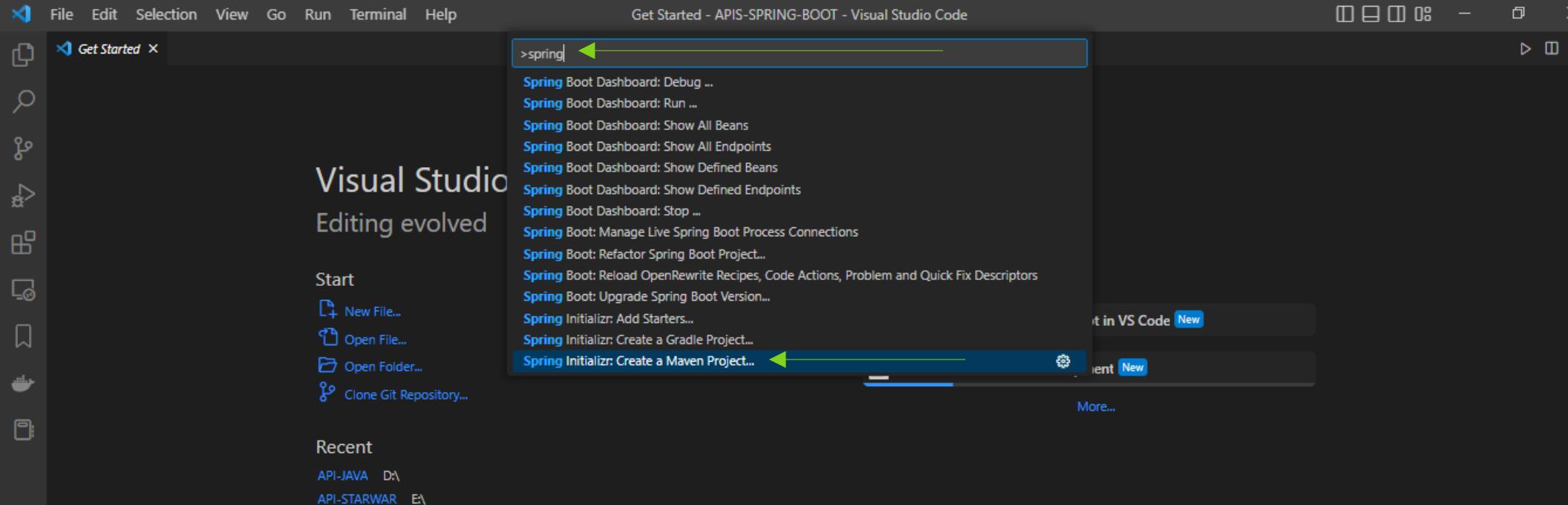
The [Spring Initializr Java Support](#) extension provides support for generating quickstart Spring Boot Java projects with Spring Initializr API.

Spring Boot Dashboard

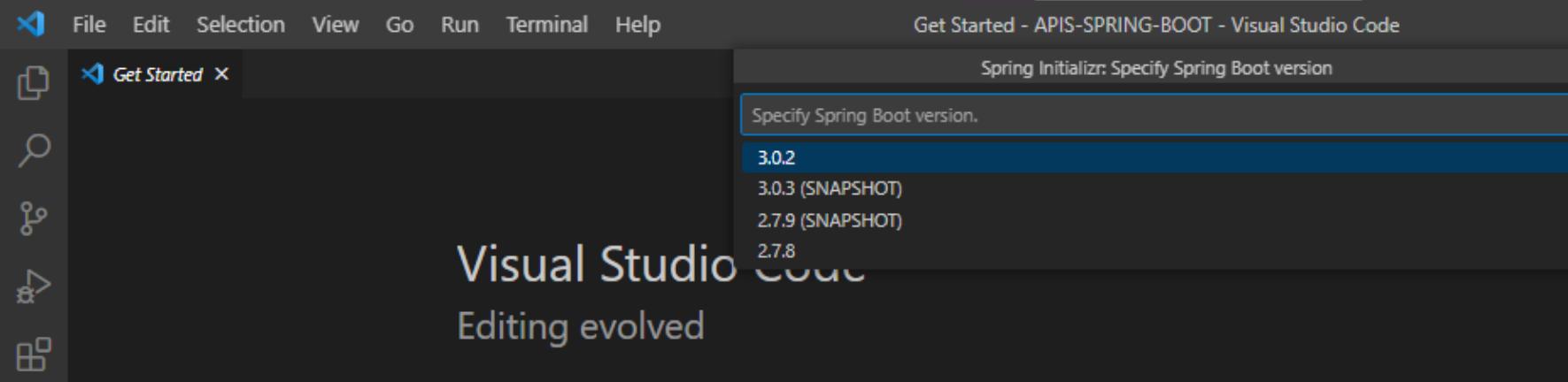
25°C Nublado Go Live 14:22

CRIANDO O PROJETO

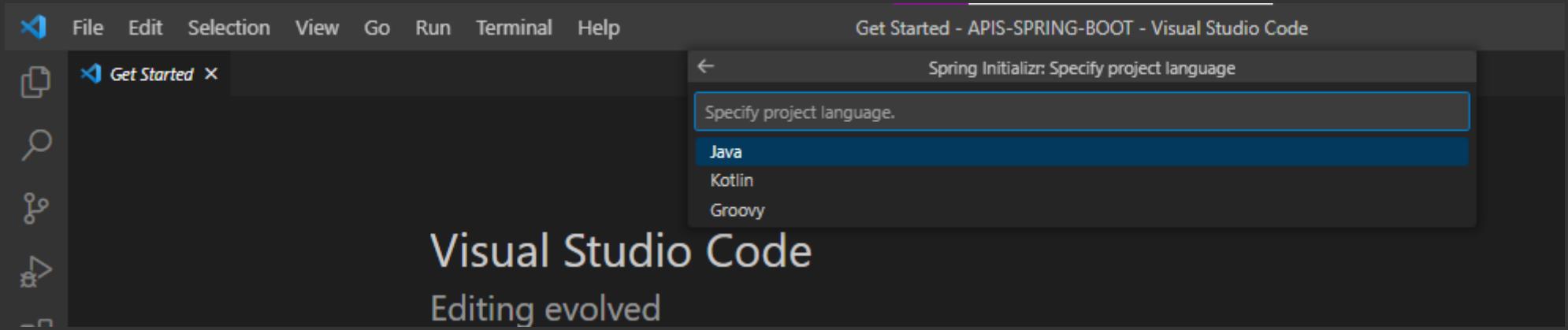




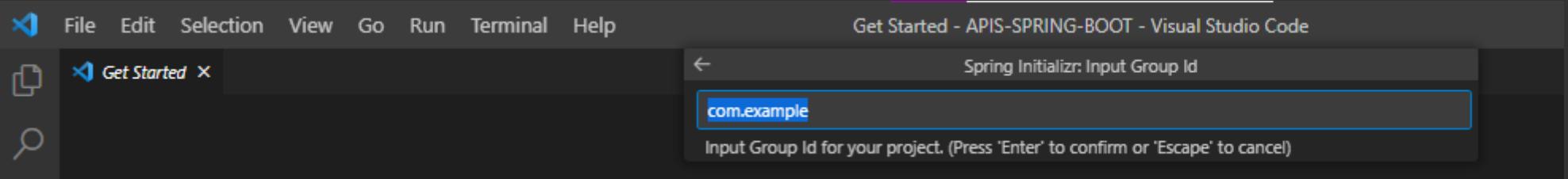
Digite spring e adicione spring initializr do maven
Essa que esta marcada ai



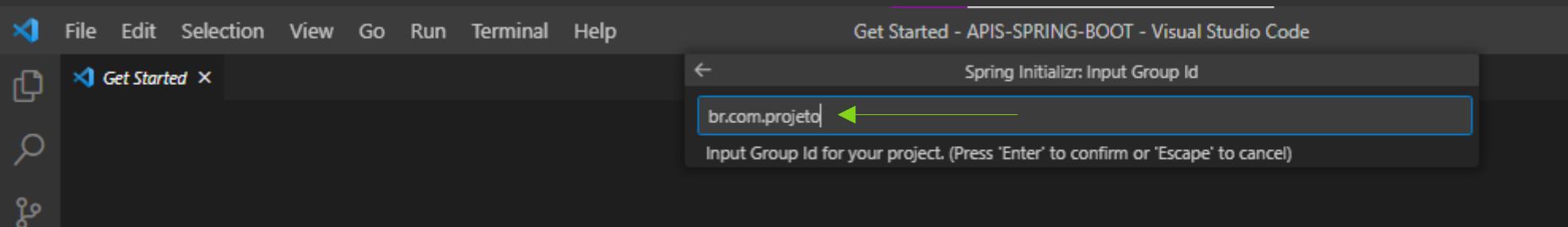
Escolha sua versão nada impede de troca-la mais tarde



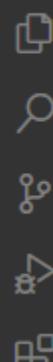
Escolha a linguagem do projeto



Group é o lugar onde vai ficar o projeto



Depois de dar o nome tecle enter



Get Started X



Spring Initializr: Input Artifact Id

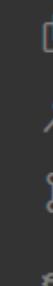
demo

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



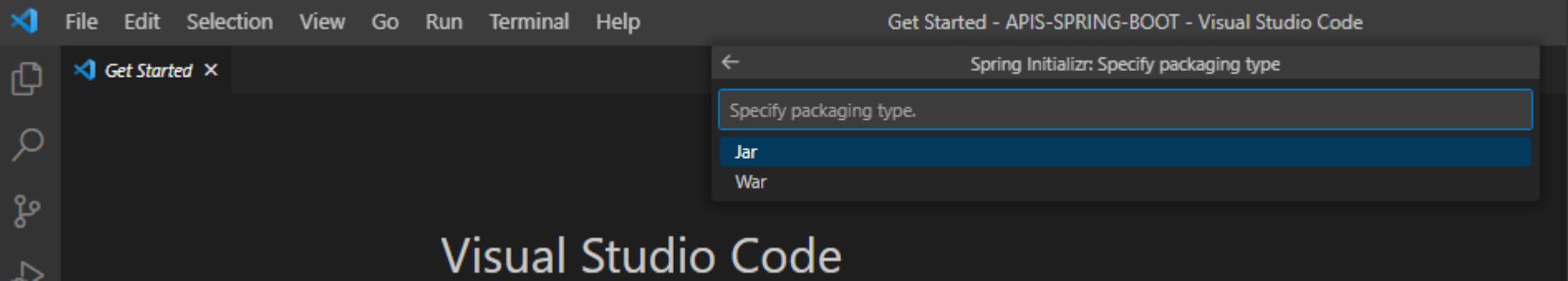
Spring Initializr: Input Artifact Id

api

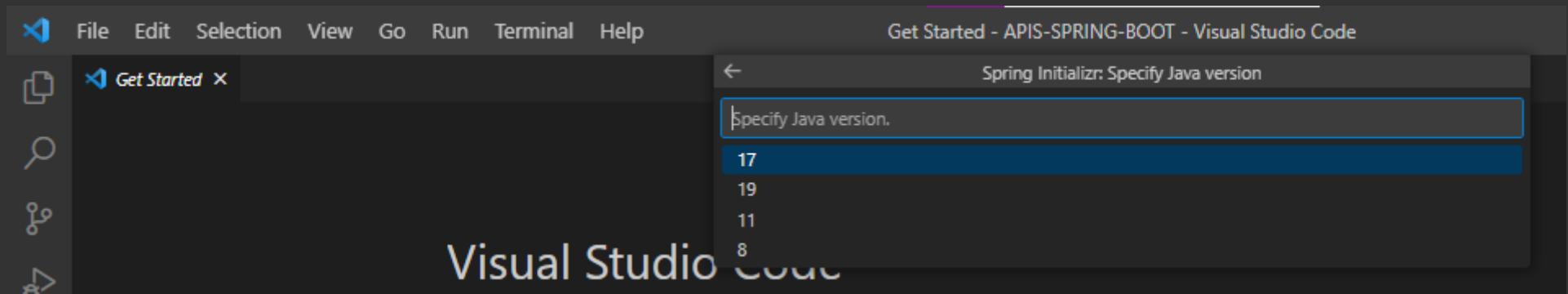
Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

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



Get Started X



Visual Studio

Editing evolved

Start

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

Recent

- API-JAVA D:\
- API-STARWARS E:\



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

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the def...

Spring Reactive Web Web

Build reactive web applications with Spring WebFlux and Netty.

Spring for GraphQL Web

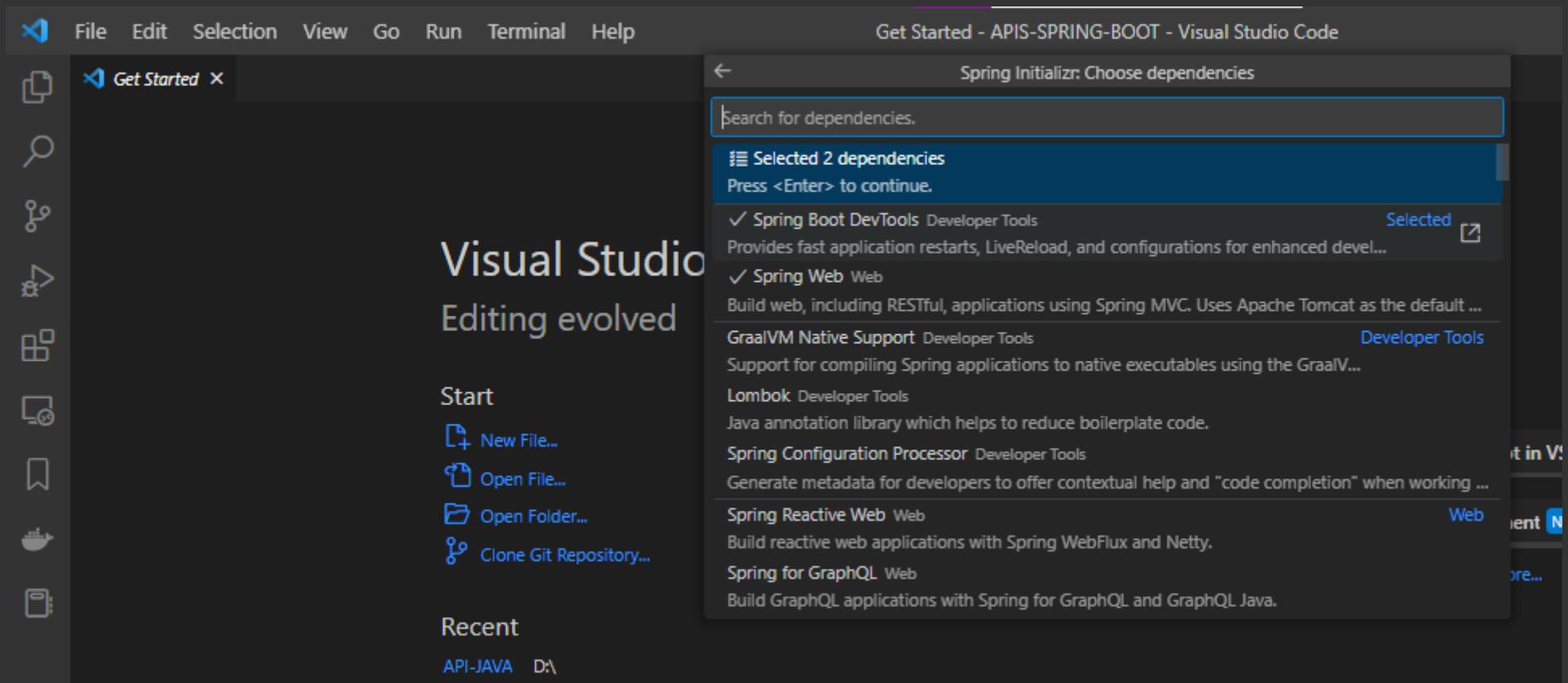
Build GraphQL applications with Spring for GraphQL and GraphQL Java.

it in VS Code New

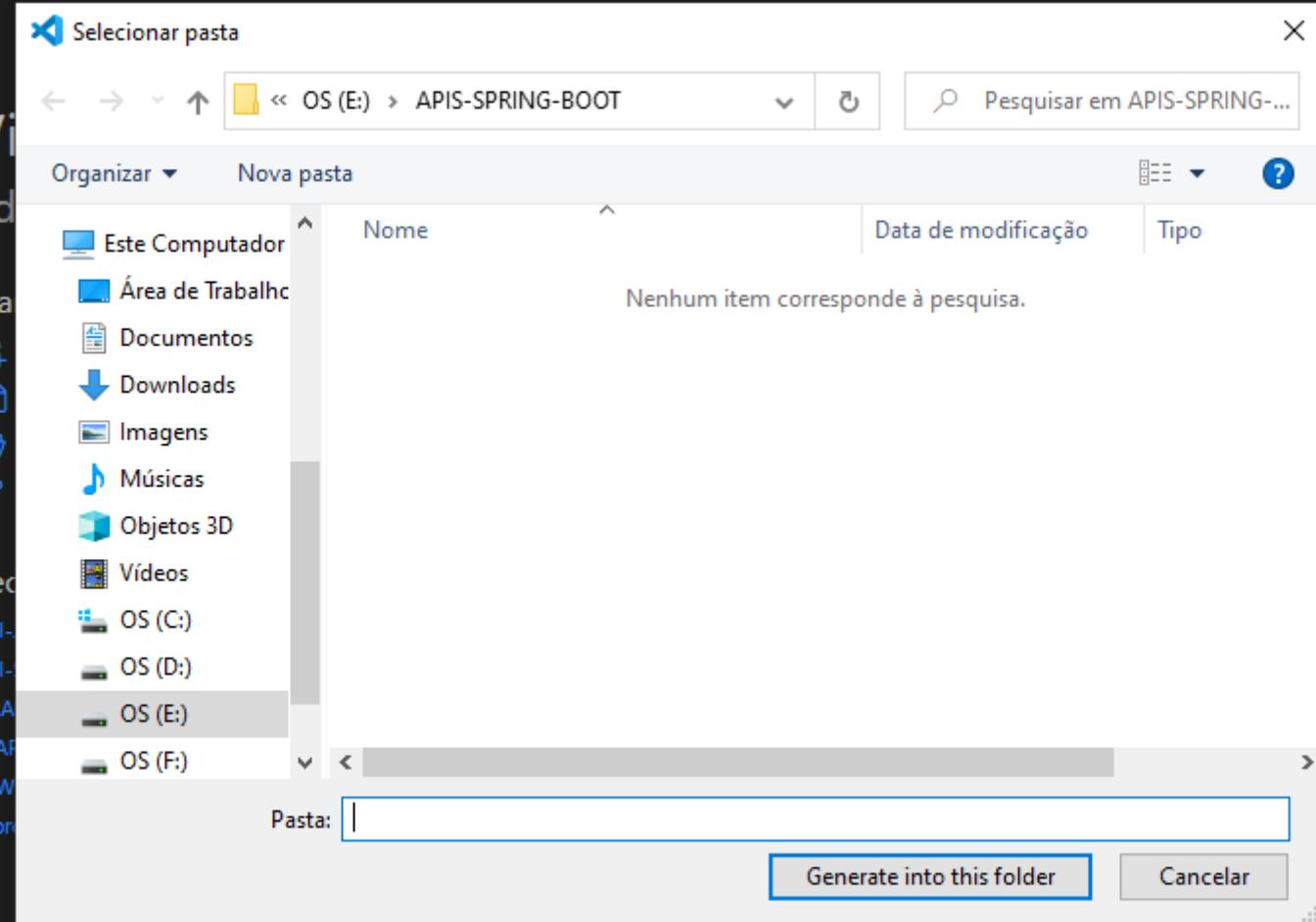
ent New

ore...

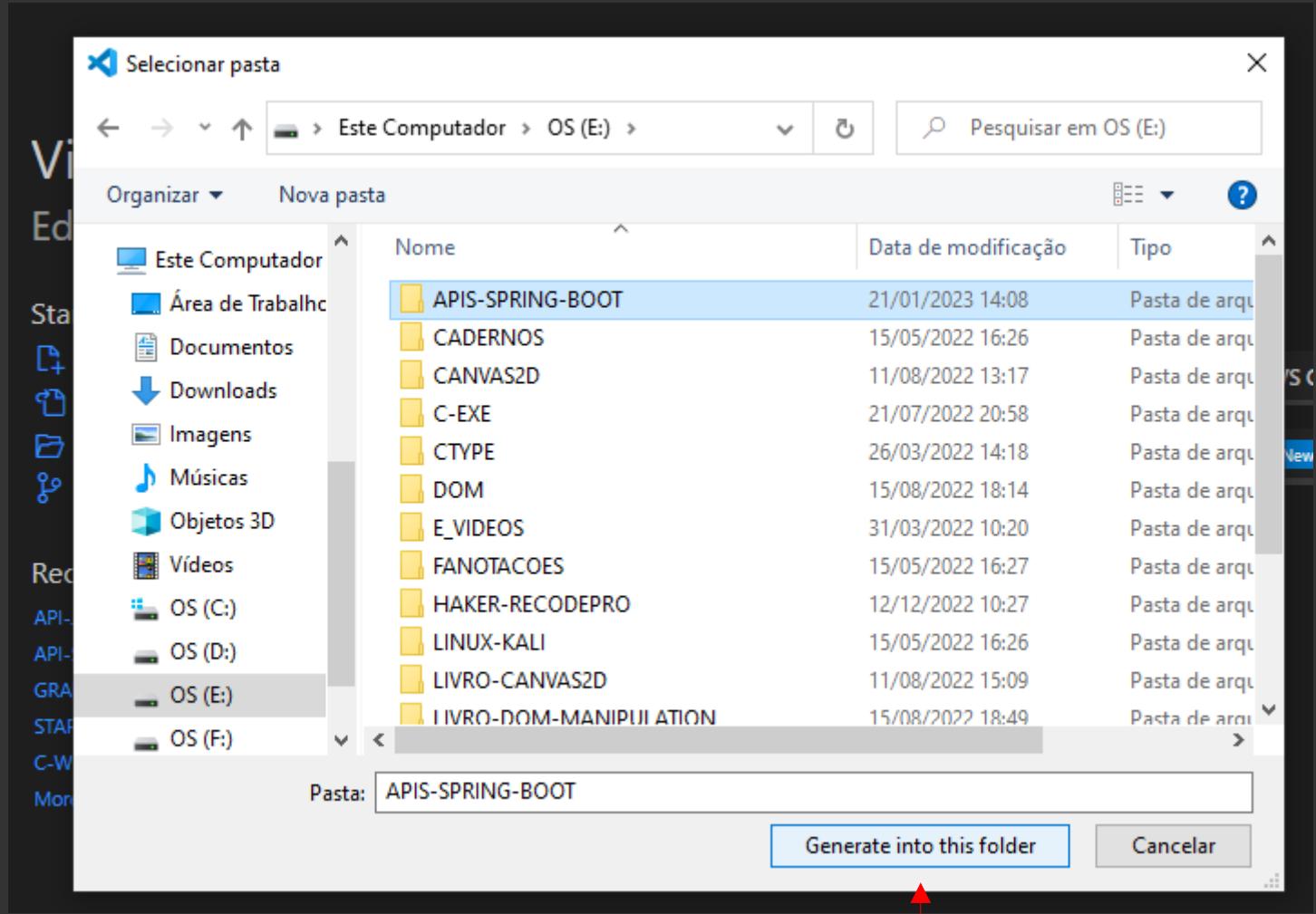
Adicione agora as dependencias do projeto
O que são dependencias? São funções que o spring usa

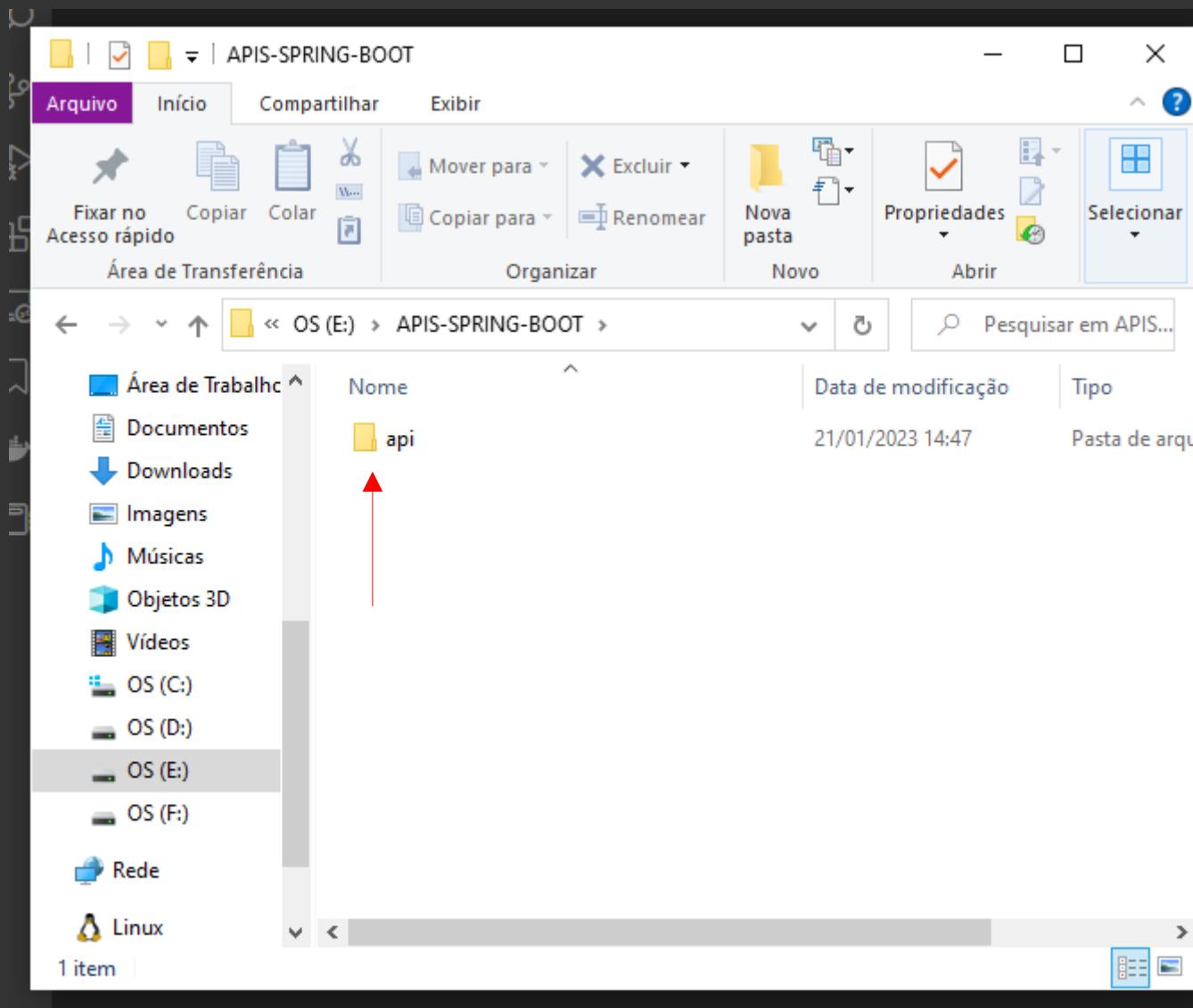


**Click para selecionar spring boot dev tools e spring web
Por fim de enter para selecionar**

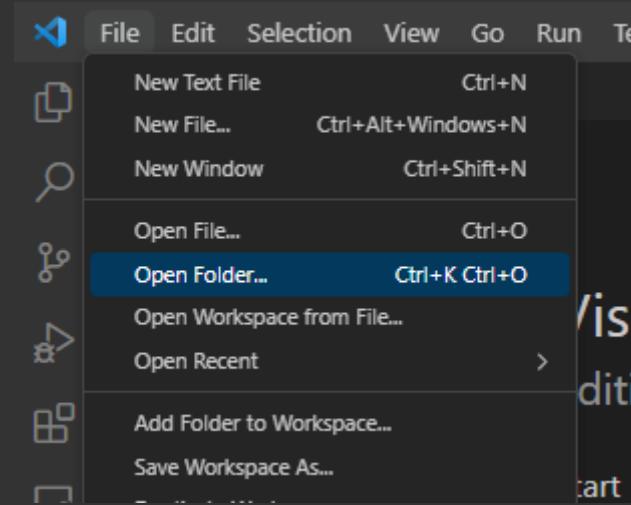


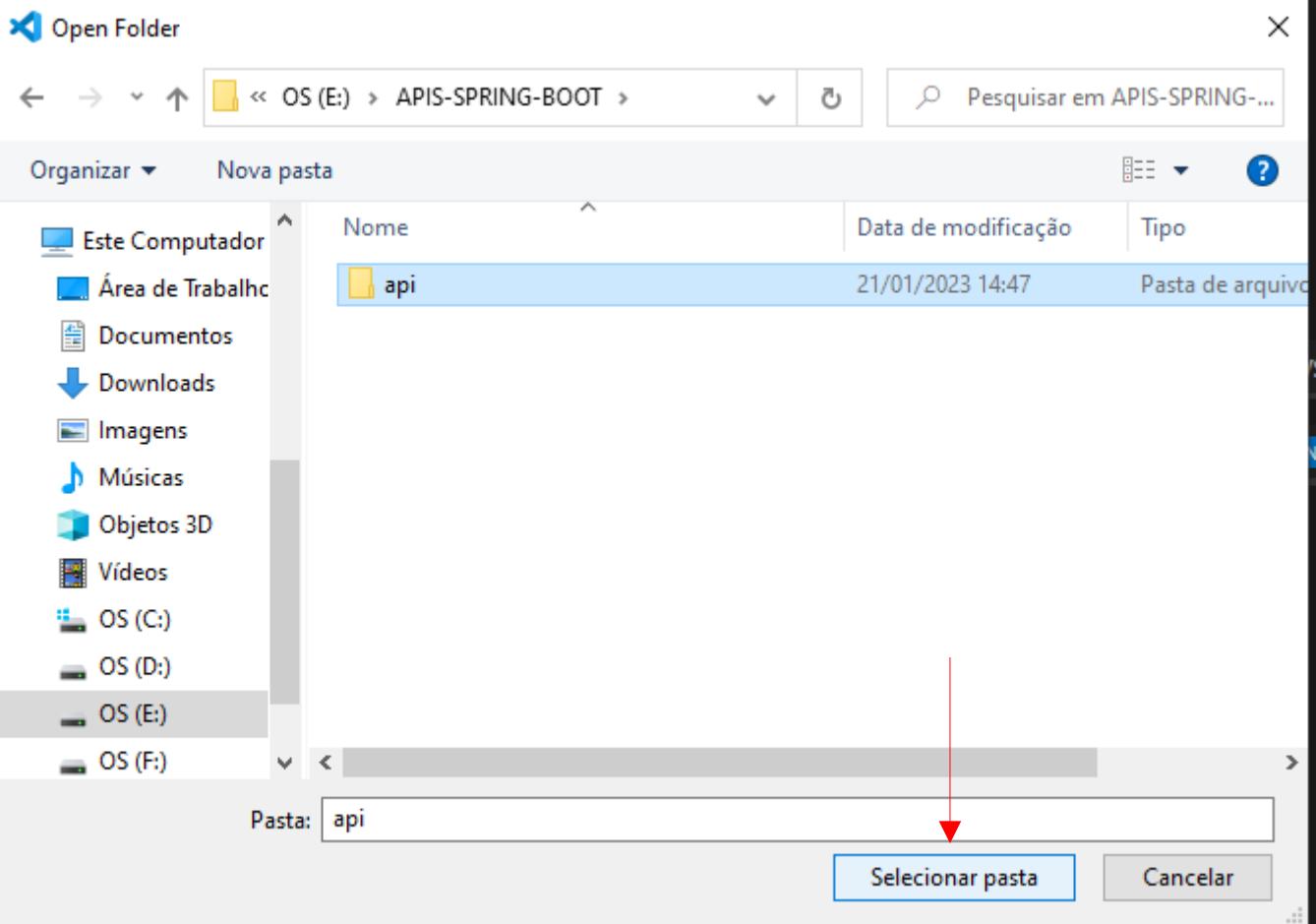
**Ele abre para você escolher onde vai salvar o projeto
Você pode escolher qualquer pasta para salvar seu projeto**





Abrindo a pasta que criamos







EXPLORER

- ... **Get Started**
- 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

- Get Started with Java Development

[More...](#)

Espere todos os arquivos serem baixados

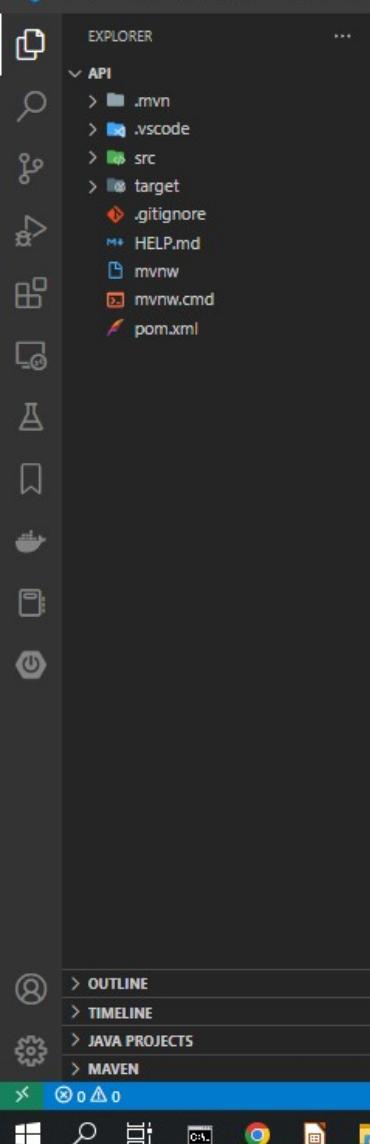
Show All Commands **Ctrl + Shift + /**

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 }
```



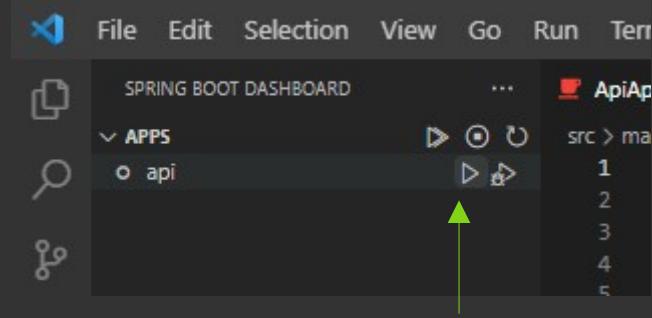
Nessa opção spring boot dashboard

The screenshot shows a code editor interface with a dark theme. At the top is a menu bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Find. The left sidebar displays a project structure under 'SPRING BOOT DASHBOARD' with 'APPS' expanded, showing a single 'api' folder. The main editor area shows the 'ApiApplication.java' file:

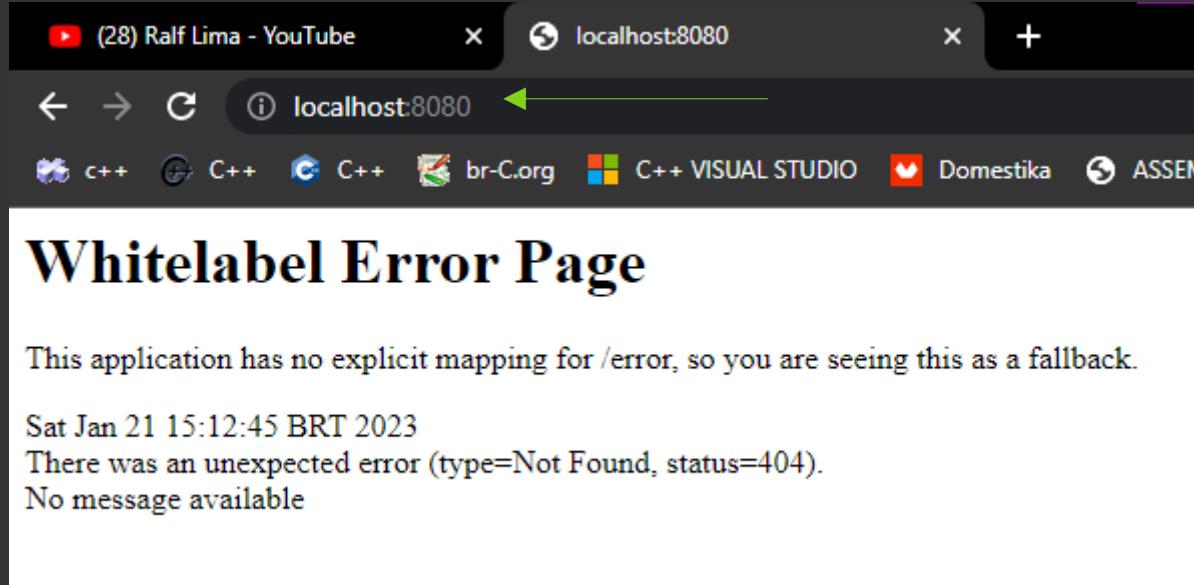
```
src > main > java > br > com  
1 package br.com.  
2  
3 import org.spring  
4 import org.spring  
5  
6 @SpringBootApp  
7 public class Ap  
8  
9     Run | Debug  
10    public stat  
11        SpringA  
12    }  
13 }  
14 }
```

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

E rodamos o projeto
Tanto faz qualquer uma das escolhas

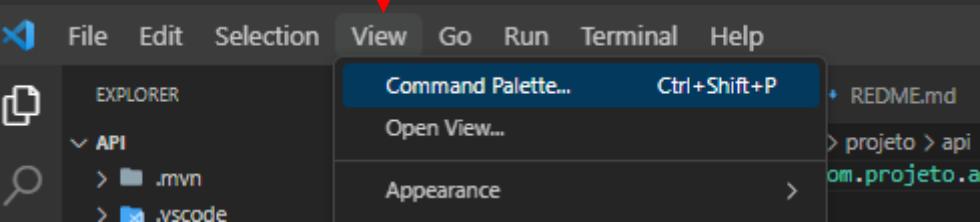


EXECUTE O PROJETO PARA FAZER UM TESTE



ERRO BOM SIGNIFICA QUE ESTA RODANDO

IMPLEMENTANDO O CONTROLLER PARA CORRIGIR O ERRO



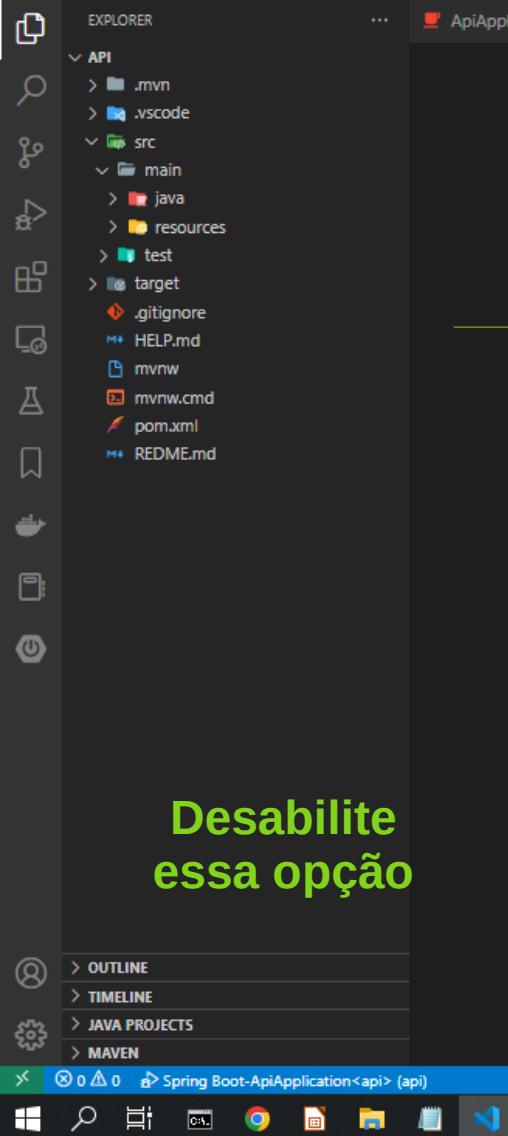
ApiApplication.java - api - Visual Studio Code

>pref

Preferences: Configure Runtime Arguments
Preferences: File Icon Theme
Preferences: Keymaps
Preferences: Language Extensions
Preferences: Open Accessibility Settings
Preferences: Open Default Keyboard Shortcuts (JSON)
Preferences: Open Default Settings (JSON)
Preferences: Open Keyboard Shortcuts
Preferences: Open Keyboard Shortcuts (JSON)
Preferences: Open Settings (UI)
Preferences: Open User Settings
Preferences: Open User Settings (JSON)
Preferences: Open Workspace Settings
Preferences: Open Workspace Settings (JSON)
Preferences: Product Icon Theme
Preferences: Toggle between Light/Dark Themes

Ctrl + K Ctrl + S

Quick Open Previous Editor From History



Desabilite essa opção

- > Text Editor
- > Workbench
- > Window
- ⌄ Features
- ▶ Explorer
 - Search
 - Debug
 - Testing
 - Source Control
 - Extensions
 - Terminal
 - Task
 - Problems
 - Output
 - Comments
 - Remote
 - Timeline
 - Notebook
 - Audio Cues
 - Merge Editor
- > Application
- > Security
- > Extensions

Exp

Auto Reveal

Controls whether the Explorer should automatically reveal and select files when opening them.

tru

Auto Reveal Exclud

Configure glob pattern

****/node_modules**

**/bower_component

Add Data

| Compact Folder

Controls whether the Explorer should render folders in a compact form. In such a form, single child folders will be compressed in a combined tree element. Useful for Java package structures, for example.

[Confirm Delete](#)

Controls whether the Explorer should ask for confirmation when deleting a file via the trash

Confirm Drag And Drop

Controls whether the Explorer should ask for confirmation to move files and folders via drag and drop.

Confirm Undo

Controls whether the Explorer should ask for confirmation when undoing

default

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 an "Ap" file. 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" (Shift+Alt+R), "Open in Integrated Terminal", "Add Folder to Java Source Path", "Remove Folder from Java Source Path", and "Find in Folder..." (Shift+Alt+F).

```
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 }
```

**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:** Shows the command "ApiApplication.java" being typed.
- Explorer:** 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
 - REDME.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

File Edit Selection View Go Run Terminal Help

• Controle.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 > controle > Controle.java > Controle

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

class Controle

interface Controle

enum Controle

record Controle()

abstract class Controle

@interface Controle

ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

REDME.md

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" (expanded), "main" (expanded), "java" (expanded), "br" (expanded), "com" (expanded), and "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 }
```

@Rest

- @RestController - org.springframework.web.bind.annotation.RestController
- @RestControllerAdvice - org.springframework.web.bind.annotation.RestControllerAdvice
- @RestControllerEndpoint - org.springframework.boot.actuate.endpoint.web.annotation.RestControllerEndpoint
- @ResponseStatus - org.springframework.web.bind.annotation.ResponseStatus
- @RegisterReflectionForBinding - org.springframework.web.bind.annotation.RegisterReflectionForBinding
- @RequestScope - org.springframework.web.context.annotation.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" (selected), "main", "java", "br", "com", "projeto", "api", "controle", "Controle.java" (highlighted). 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 for the "@Get" annotation is open, 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
 - ApiApplication.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 }
15
```
- 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

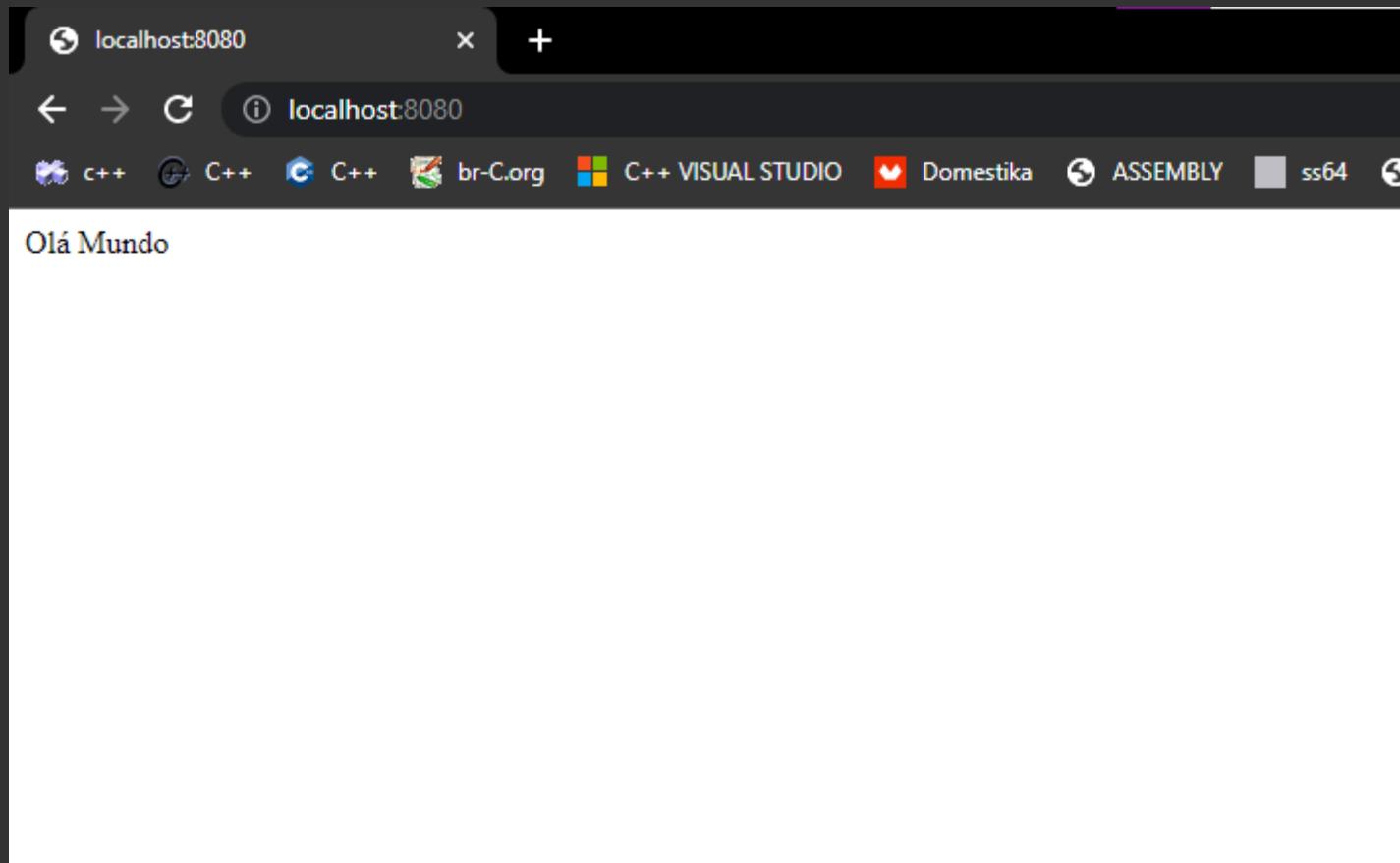
mvnw

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

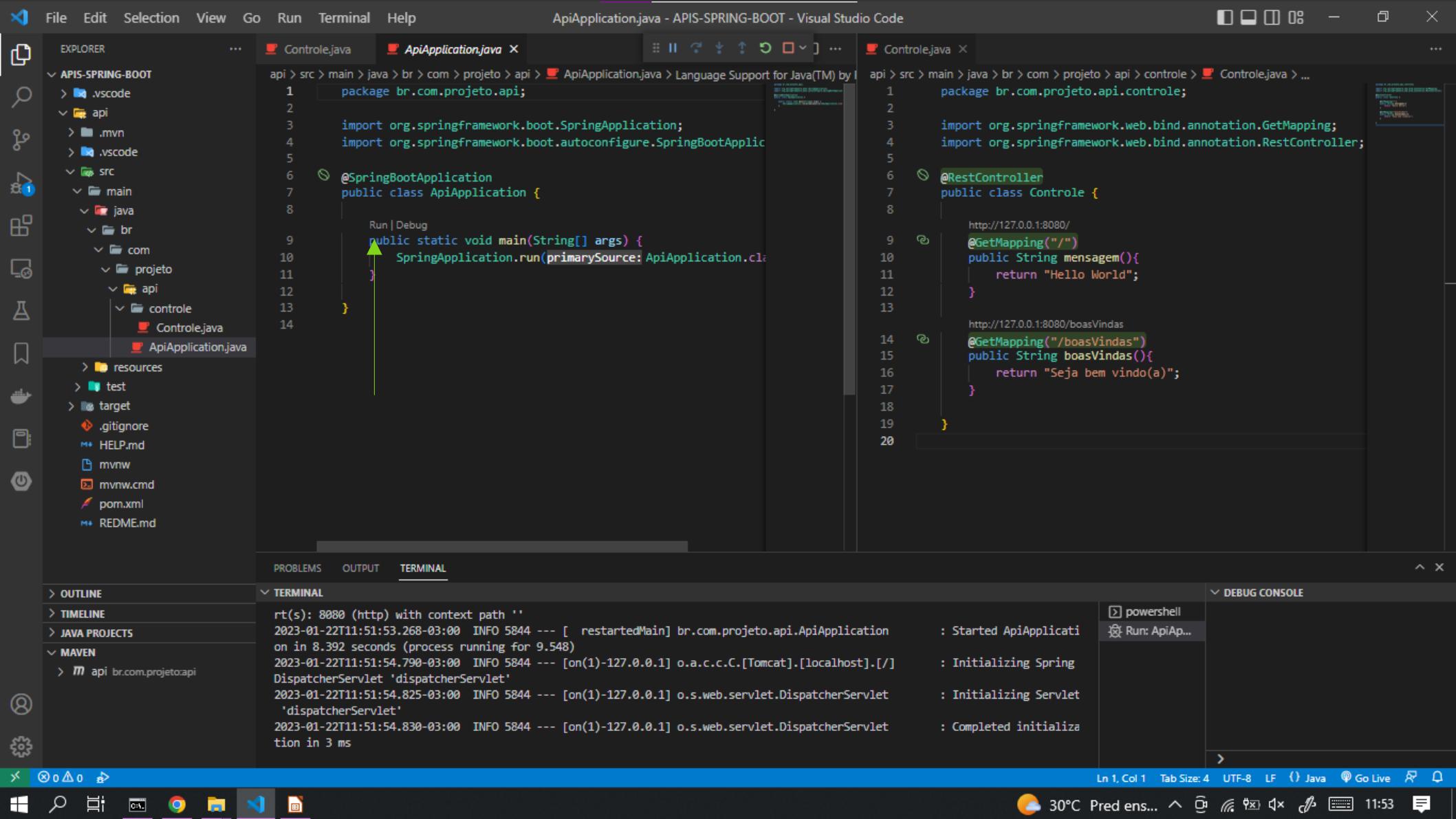
#6 **@PathVariable**

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

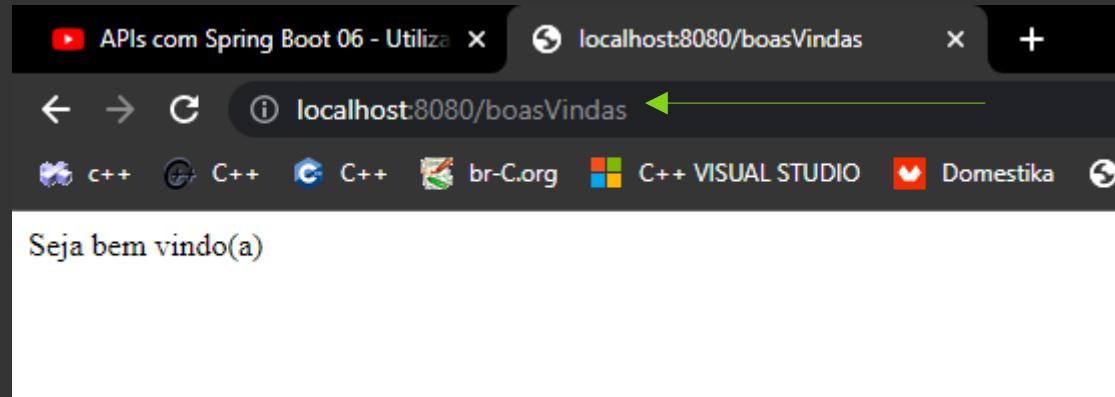
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "src/main/java" folder contains "br.com.projeto.api" which further contains "controle" and "Controle.java". Other files like "ApiApplication.java", ".gitignore", "mvnw", "mvnw.cmd", "pom.xml", and "REDME.md" are also listed.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

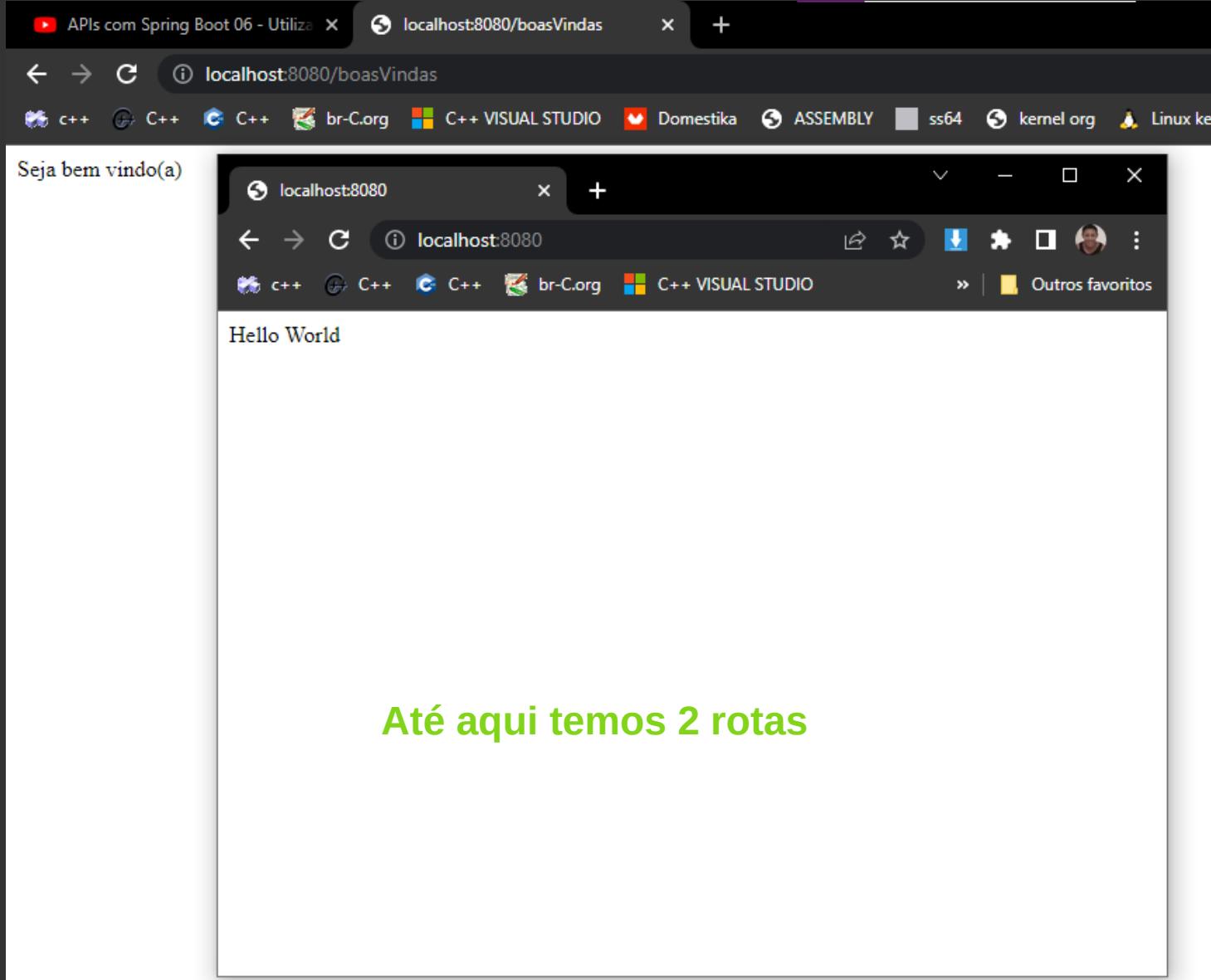
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping("/")
10    public String mensagem(){
11        return "Hello World";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

> .vscode

< api

> .mvn

> .vscode

< src

< main

< java

< br

< com

< projeto

< api

< controle

Controle.java 3

ApiApplication.java

> resources

> test

> target

< .gitignore

< HELP.md

< mvnw

< mvnw.cmd

< pom.xml

< 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.java

Controle.java

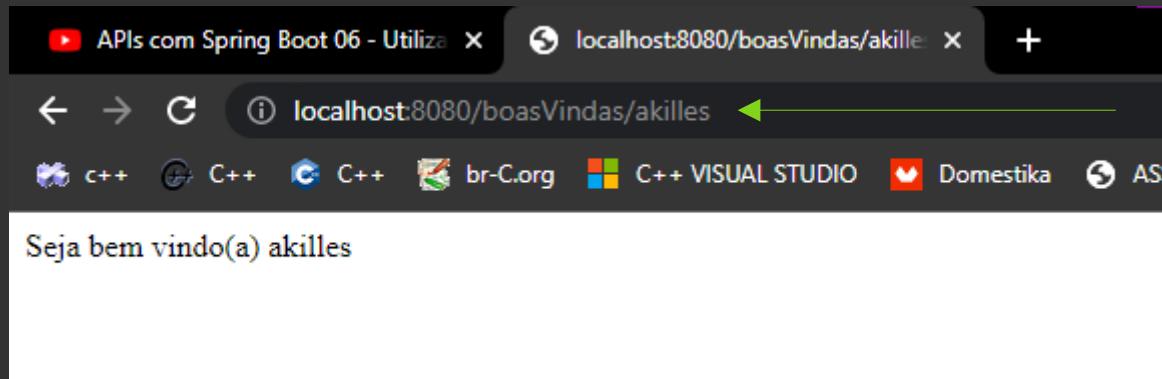
api > src > main > java > br > com > projeto > api > controle > Controle.java > ...

```
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    }
21
```

{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

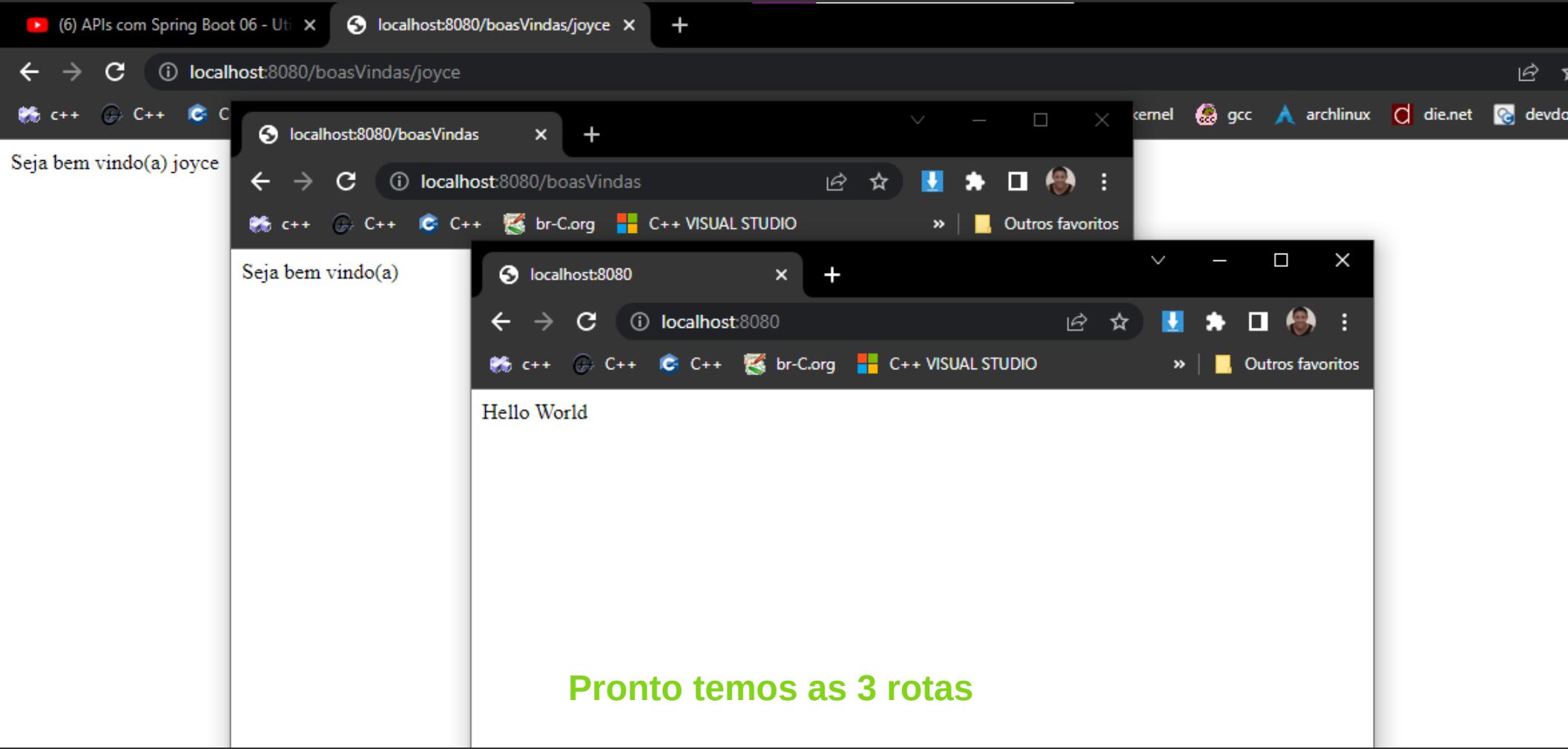
APIS-SPRING-BOOT

- .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



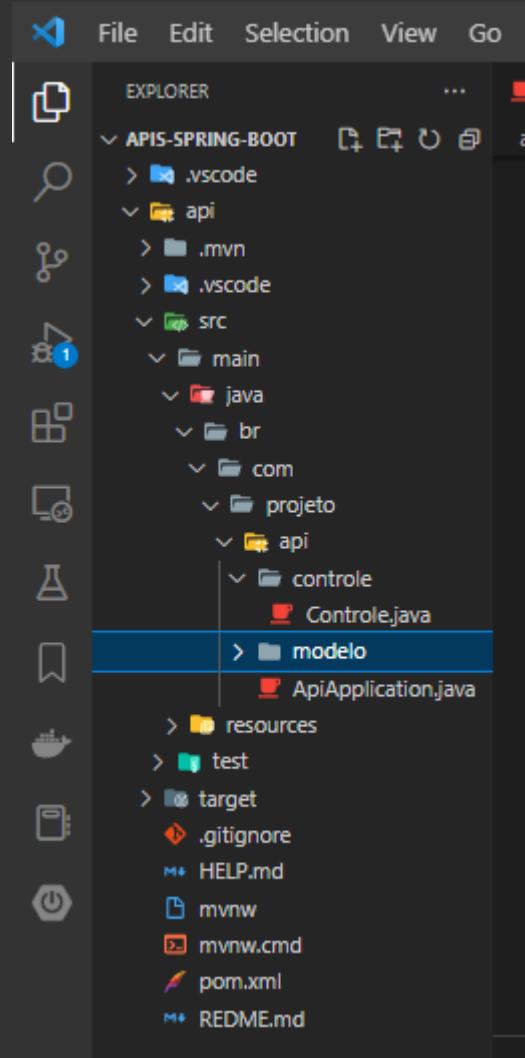
#7
MODELOS

**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...". The code editor on the right displays a simple "Controle.java" file with the following content:

```
2
3     import org.springframework.web.bind.annotation.*;
4     import org.springframework.web.bind.annotation.RequestMapping;
5     import org.springframework.web.bind.annotation.RestController;
6
7     @RestController
8     public class Controle {
9
10        @RequestMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14    }
```

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



Pessoa.java
Controle.java



PROBLEMS
OUTPUT
TERMINAL

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

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**

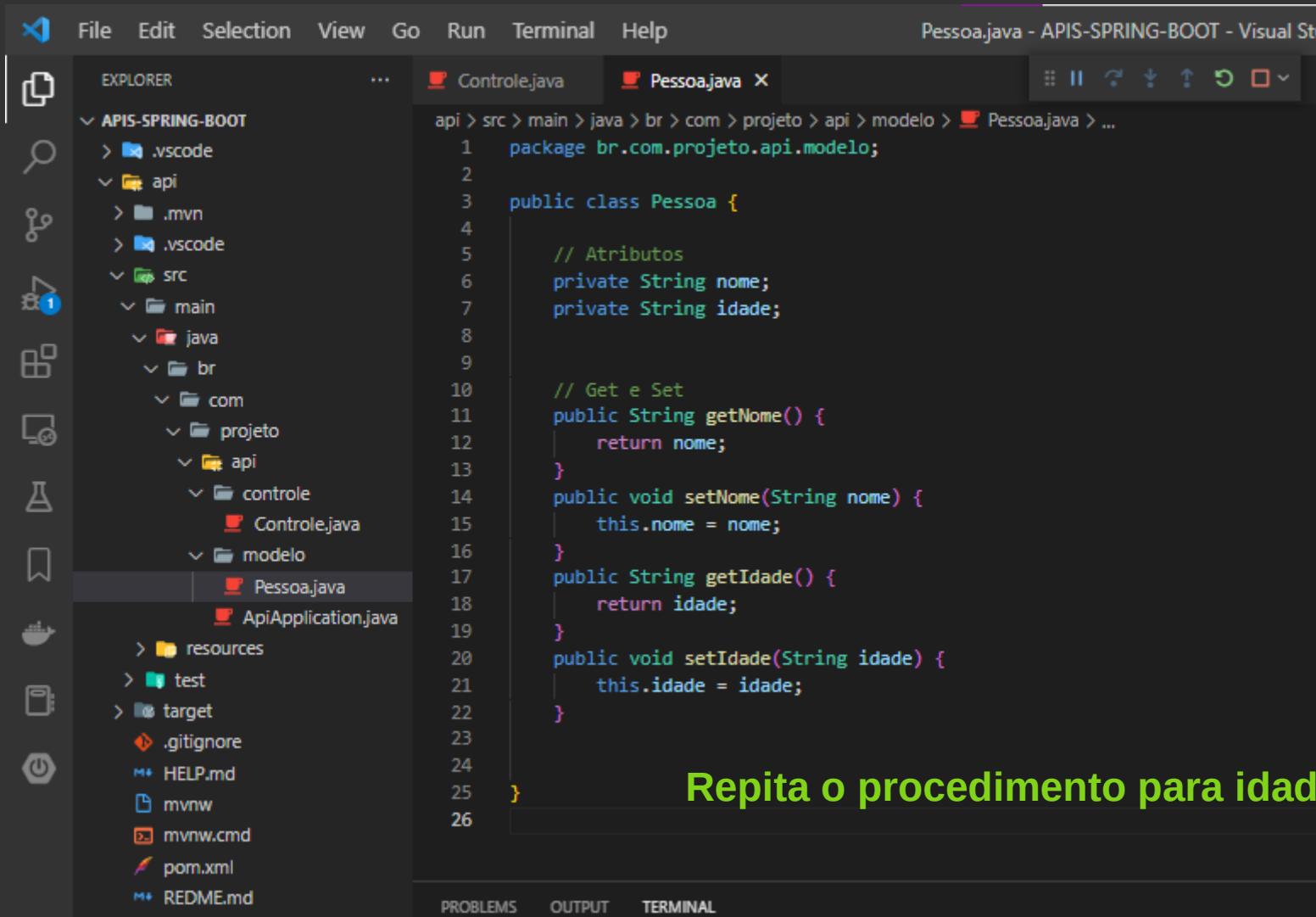
**Lembre-se de deixar os atributos sempre privados pois podemos manipula-
los atravez dos metodos get e set**

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', 'jar', and 'dependency' subfolders.
- Code Editor:** 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 'Remove 'nome'', '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'
- 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



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
- > resources
- > test
- > target
- > .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- REDME.md

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     // Atributos
5     private String nome;
6     private String idade;
7
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
```

Colocados na ordem manualmente

PROBLEMS OUTPUT TERMINAL

#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    }
```

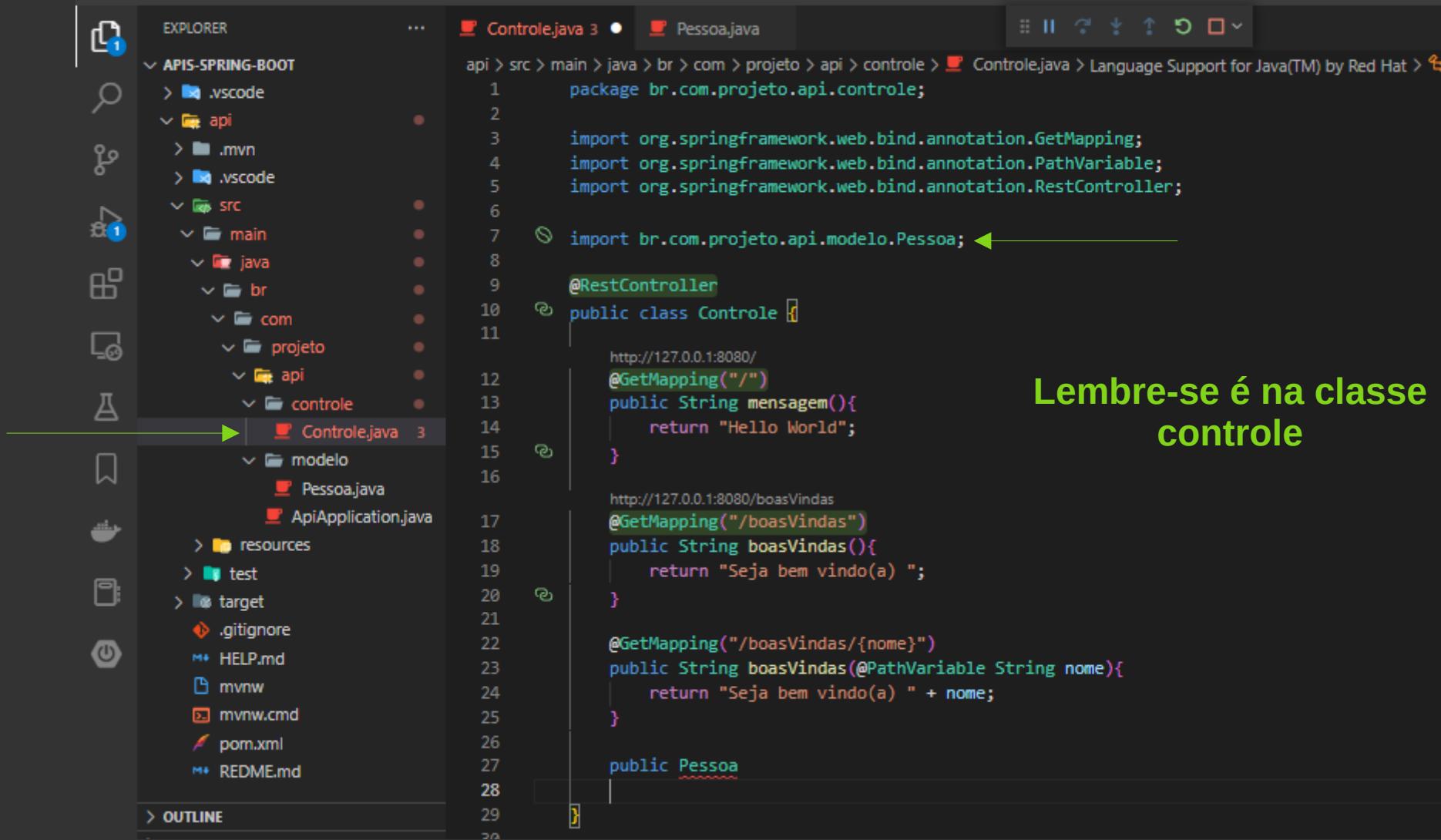
PROBLEMS 3 OUTPUT TERMINAL

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

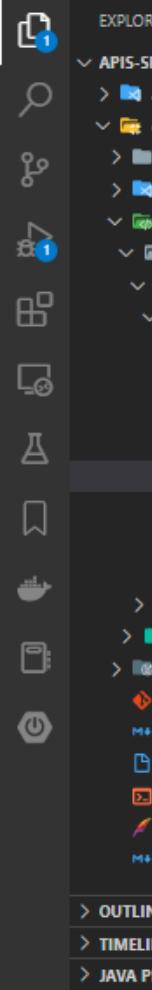
Pessoa.java

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



Controle.java 3 ● Pessoajava

```
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

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
- < modelo
- Pessoa.java
- ApiApplication.java
- > resources
- > test
- > target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- README.md

Controle.java 3 • Pessoa.java

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
- < README.md

Controle.java

Pessoa.java

```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for J
```

```
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        @GetMapping("/boasVindas/{nome}")
25        public String boasVindas(@PathVariable String nome){
26            return "Seja bem vindo(a) " + nome;
27        }
28
29        public Pessoa pessoa(@RequestBody Pessoa p){
30            return p;
31        }
32    }
33
34 }
```

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:

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

Controle.java • Pessoa.java

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

```
2
3      import org.springframework.web.bind.annotation.GetMapping;
4      import org.springframework.web.bind.annotation.PathVariable;
5      import org.springframework.web.bind.annotation.PostMapping;
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         http://127.0.0.1:8080/
15         @GetMapping("/")
16         public String mensagem(){
17             return "Hello World";
18
19         http://127.0.0.1:8080/boasVindas
20         @GetMapping("/boasVindas")
21         public String boasVindas(){
22             return "Seja bem vindo(a) ";
23
24         @GetMapping("/boasVindas/{nome}")
25         public String boasVindas(@PathVariable String nome){
26             return "Seja bem vindo(a) " + nome;
27
28
29         @PostMapping
30         public Pessoa pessoa(@RequestBody Pessoa p){
31             return p;
32
33 }
```

PROBLEMS OUTPUT TERMINAL

EXPLORER ... Controle.java X Pessoa.java

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

OUTLINE

TIMELINE

JAVA PROJECTS

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

```
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    }
```

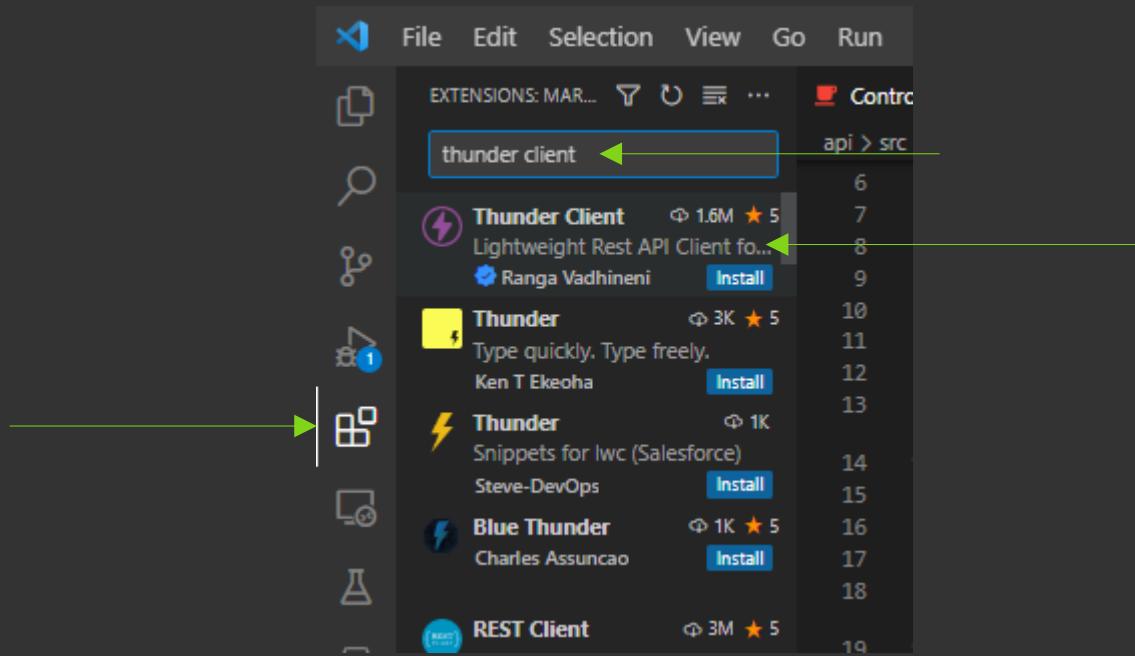
PROBLEMS OUTPUT TERMINAL

Por fim criamos a rota /pessoa

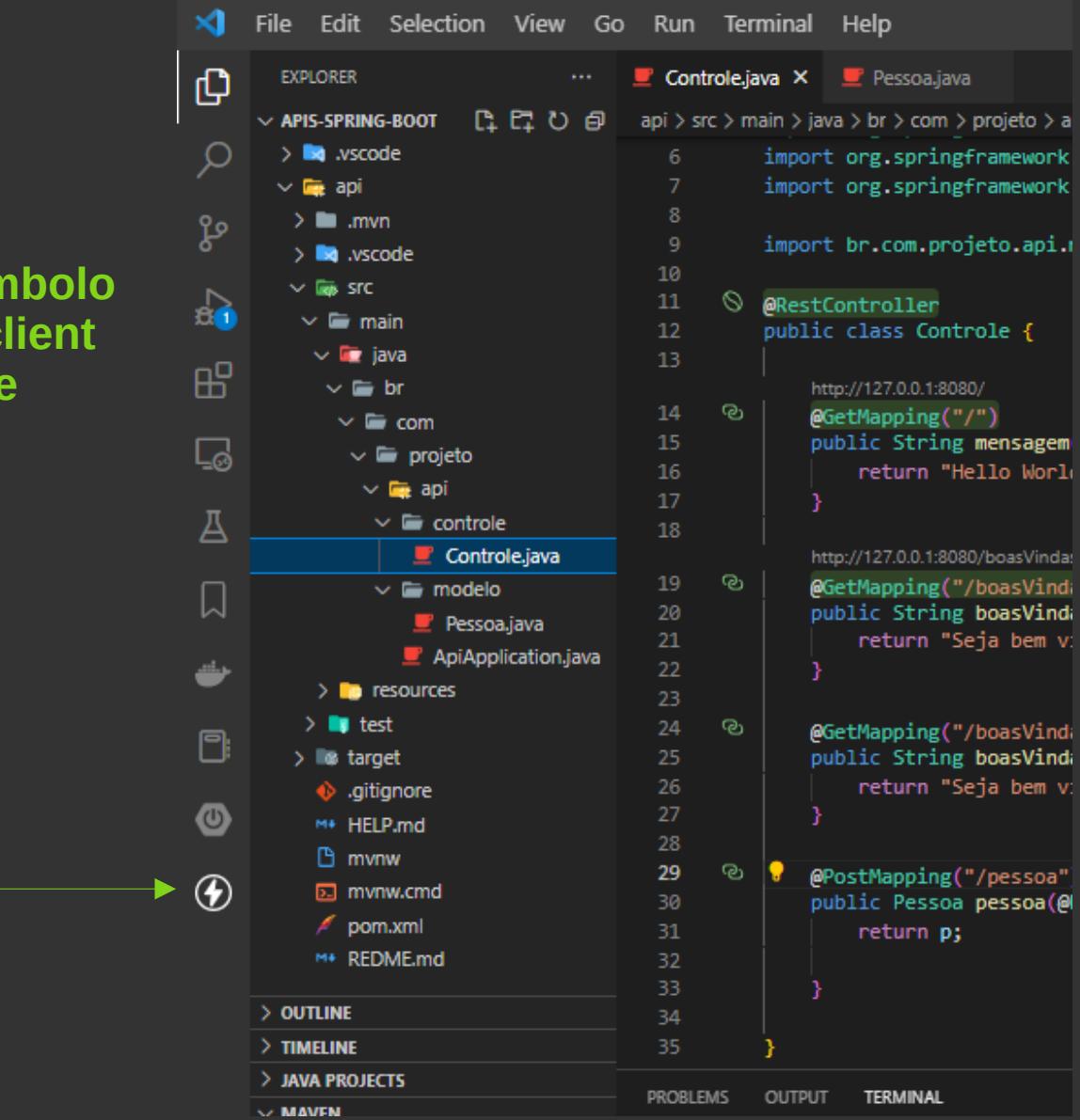
#9
Testando o projeto com thunder

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



Repare no simbolo
do thunder client
Click nele

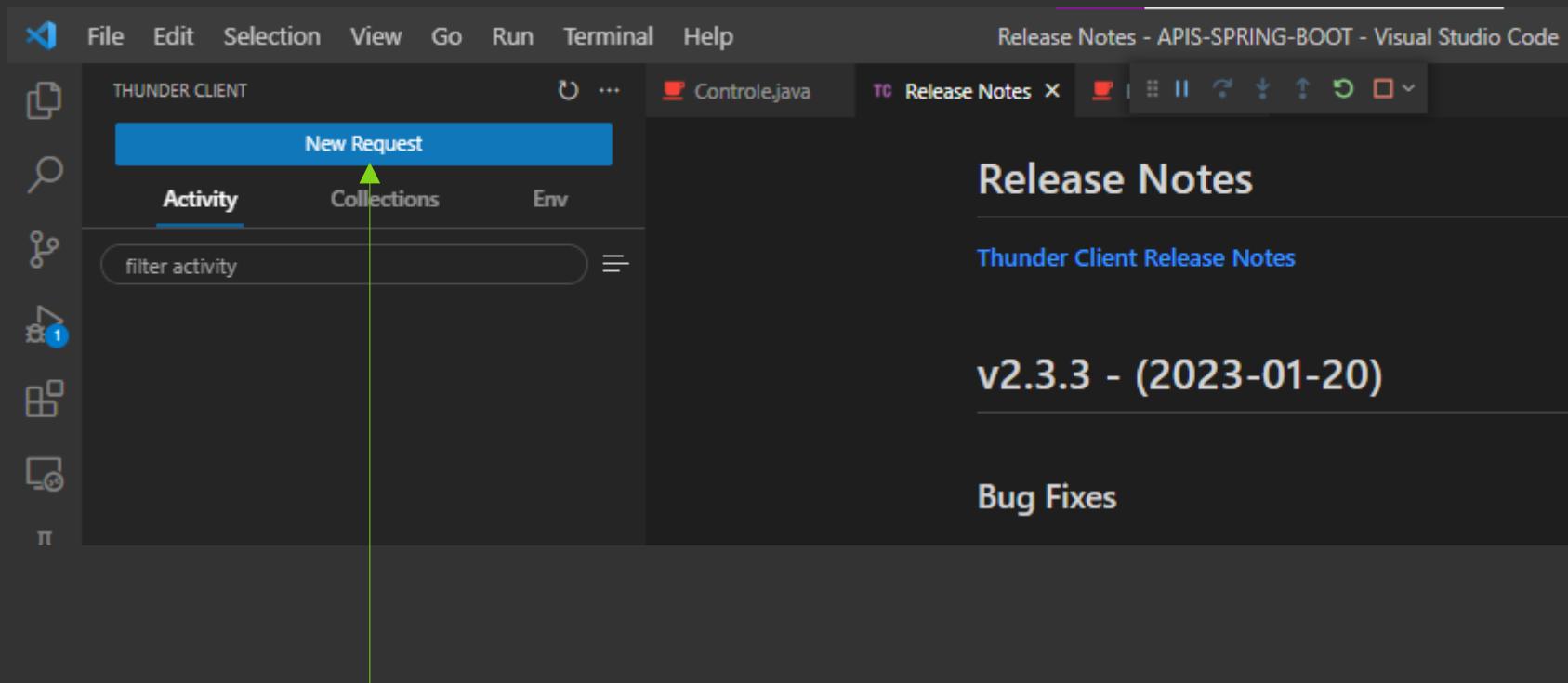


The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Bar:** 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

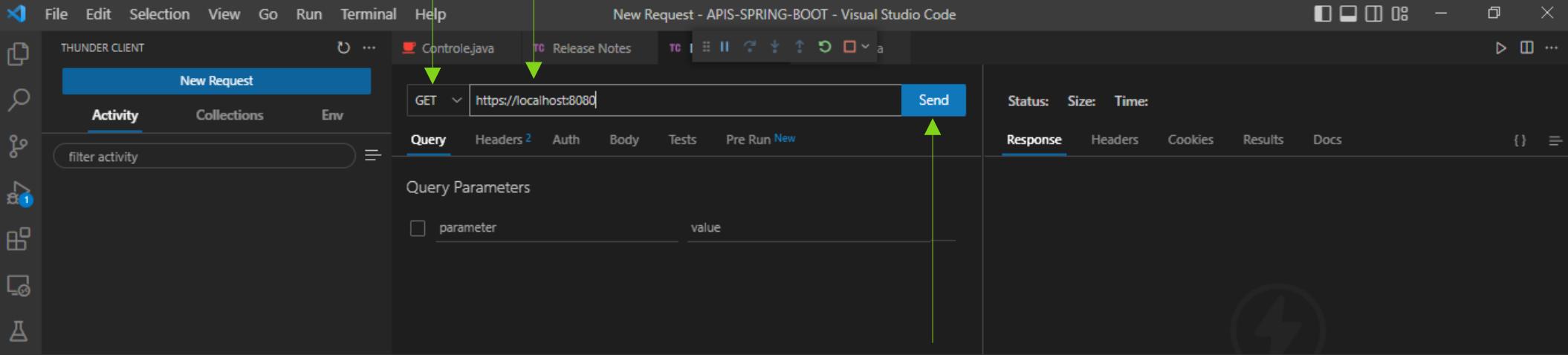
2023-01-22T12:59:05.123-03:00 INFO 5844 --- [restartedMain] br.com.projeto.api.ApiApplication : Started ApiApplication in 0.798 seconds (process running for 4041.403)

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

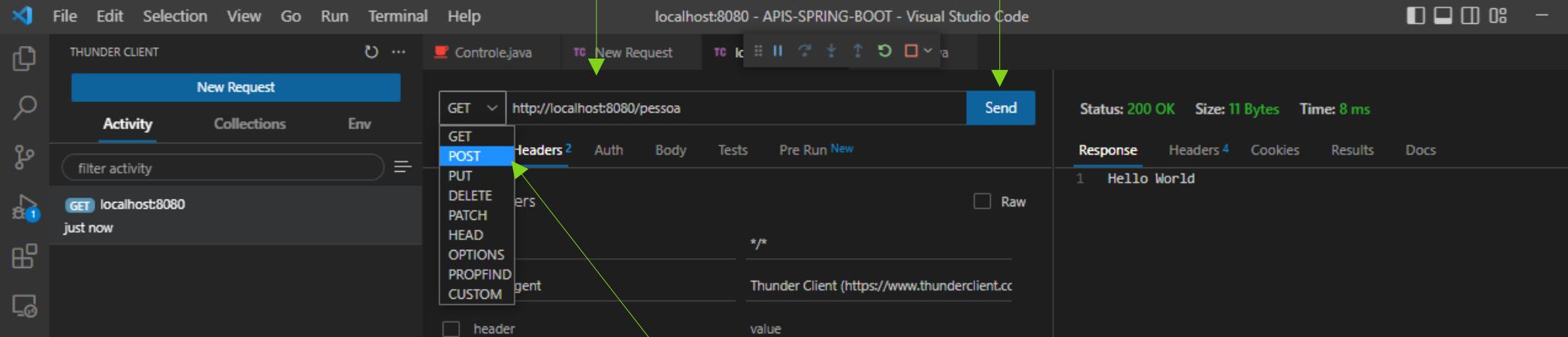


Fazemos um teste na rota hello world digite a url com o metodo get e click em send

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 to 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](#) [⚙️](#)

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](#).

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

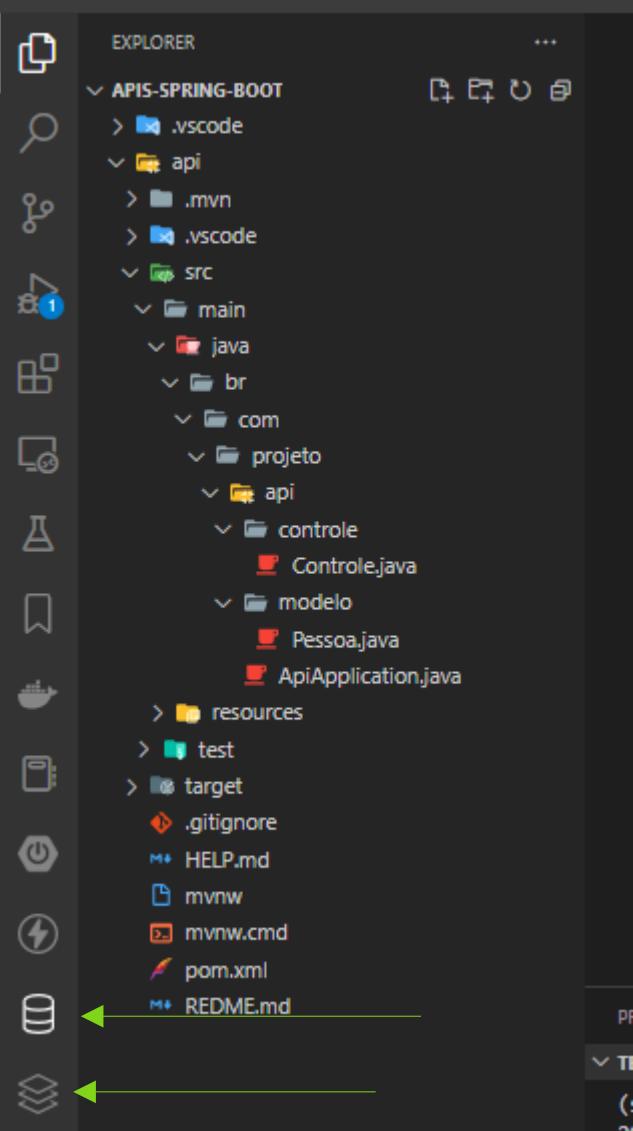
```
(s): 8080 (http) with context path ""
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApp...
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sp...
2023-01-22T13:41:52.193-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Se...
2023-01-22T13:41:52.194-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initi...
2023-01-22T13:41:52.200-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initialization in 4 ms
```

DEBUG CONSOLE

- [powershell](#)
- [Run: ApiAp...](#)

31°C Parc ens... 13:53

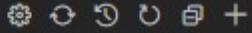
File Edit Selection View Go Run Terminal



Ela cria 2 atalhos click no database



DATABASE



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



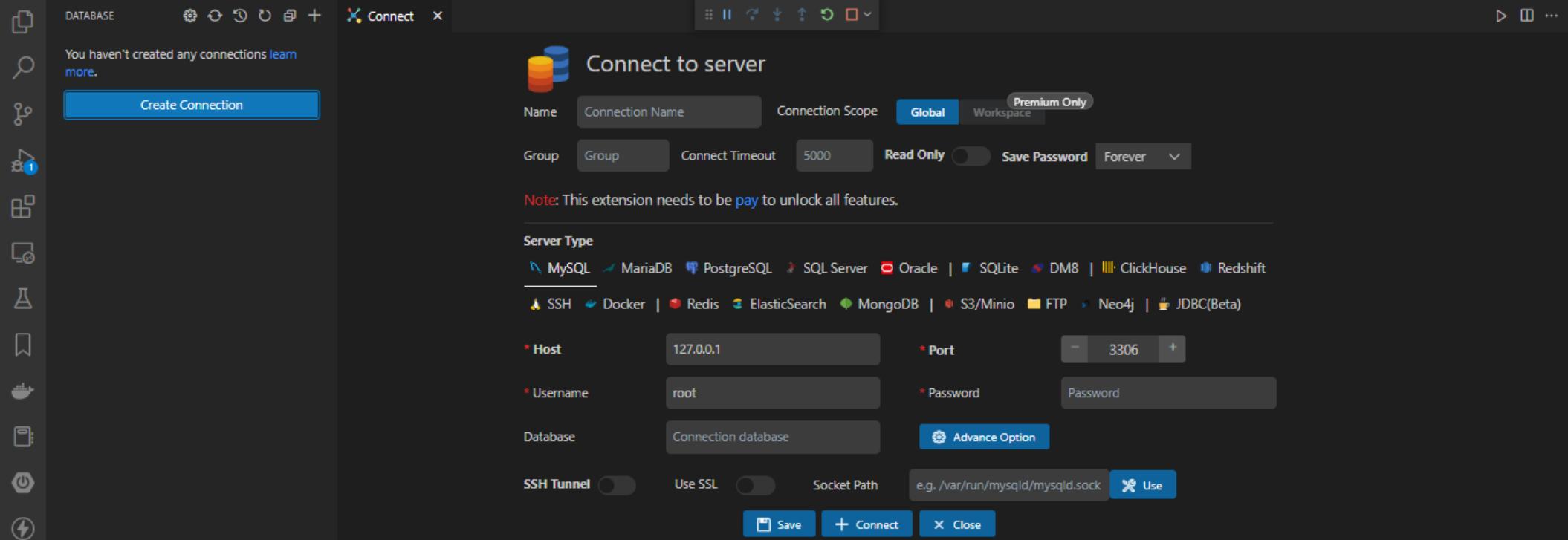
Create Connection



PROBLE

TERMIN

(s): 8



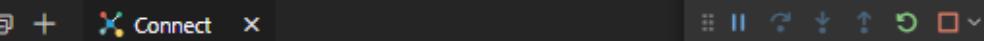
PROBLEMS OUTPUT TERMINAL

```
(s): 8080 (http) with context path ''
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication      : Started ApiAp
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
DispatcherServlet 'dispatcherServlet'
2023-01-22T13:41:52.194-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      : Initializing
servlet 'dispatcherServlet'
2023-01-22T13:41:52.200-03:00 INFO 10316 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      : Completed ini
alization in 4 ms
[]
```

◀ DEBUG CONSOLE

powershell

PowerShell



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

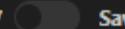
Group

Group

Connect Timeout

5000

Read Only



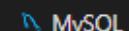
Save Password

Forever



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

Server Type



MySQL



MariaDB



PostgreSQL



SQL Server



Oracle



SQLite



DM8



ClickHouse



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

X Close

Akilles101

A senha é a verdadeira do mysql

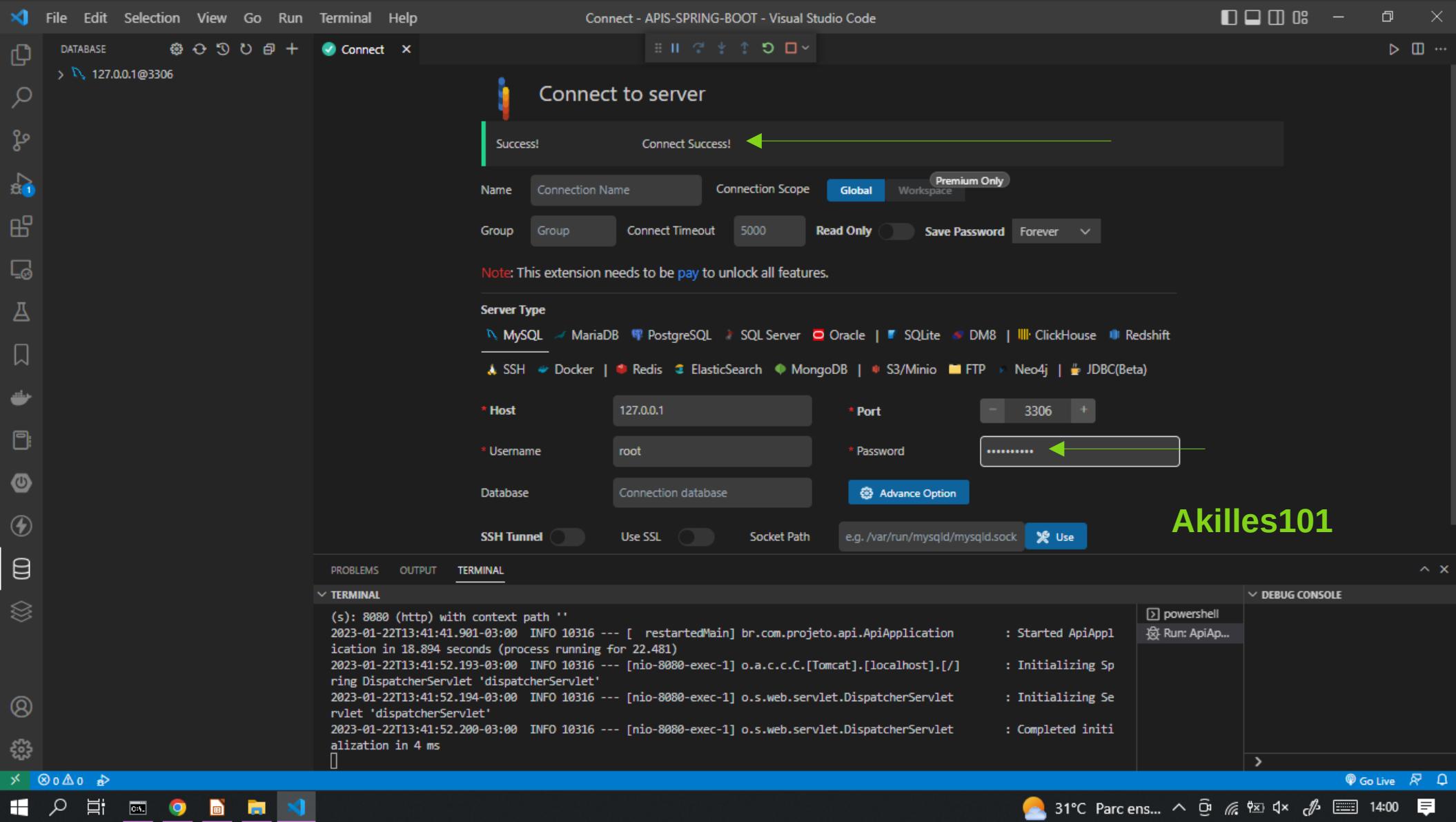
PROBLEMS OUTPUT TERMINAL

TERMINAL

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

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

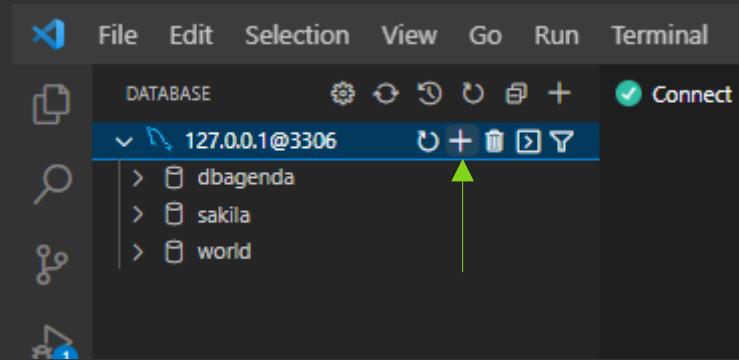
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

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 displays "create-db-template.sql - APIS-SPRING-BOOT - Visual Studio". The left sidebar has icons for Database, Schema, User, and Session, with "DATABASE" selected. Below it, a connection tree shows "127.0.0.1@3306" expanded, listing databases "dbagenda", "sakila", and "world". The main workspace contains a terminal window with the following content:

```
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 has a terminal tab titled 'create-db-template.sql' which contains the following SQL script:

```
CREATE DATABASE api_spring
    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
 - ApiApplication.java
 - resources
 - static
 - templates
 - application.properties
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - README.md

ApiApplication.java X

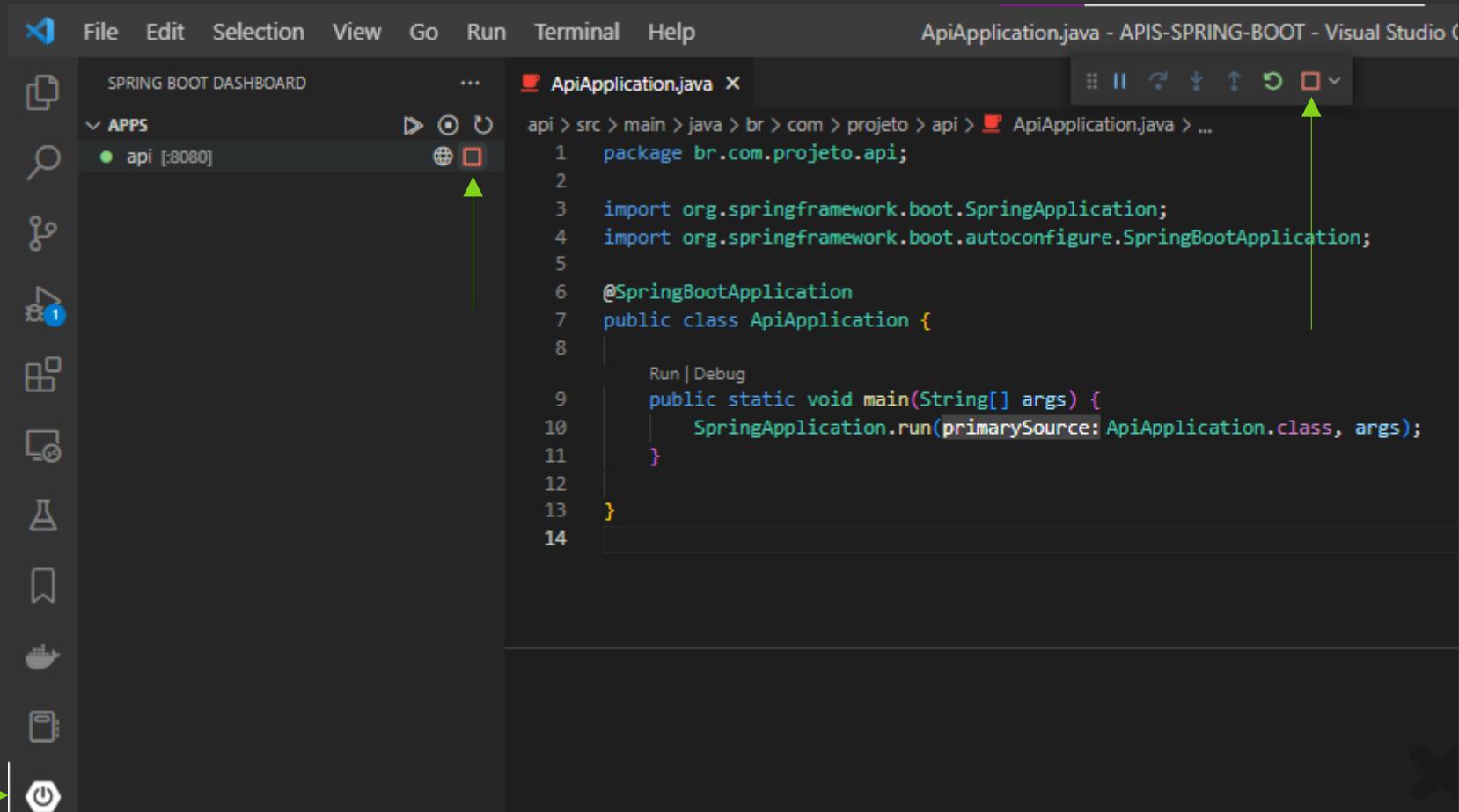
api > src > main > java > br > com > projeto > api > ApiApplication.java > ...

```
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)



File Edit Selection View Go Run Terminal Help

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

SPRING BOOT DASHBOARD

APPS

api [:8080]

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 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>

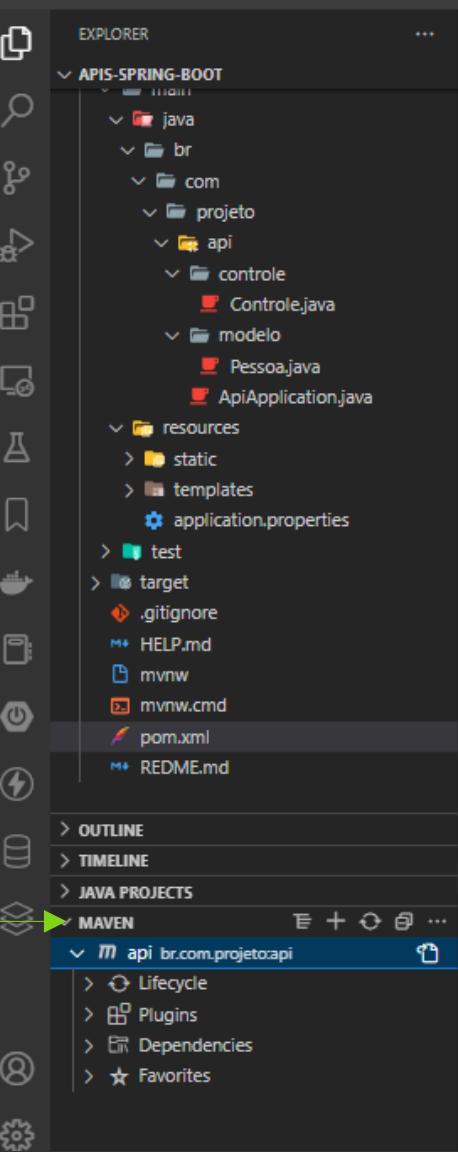
DEBUG CONSOLE

Run: ApiAp...

Run: ApiAp...

ADICIONANDO AS DEPENDÊNCIAS

File Edit Selection View Go Run

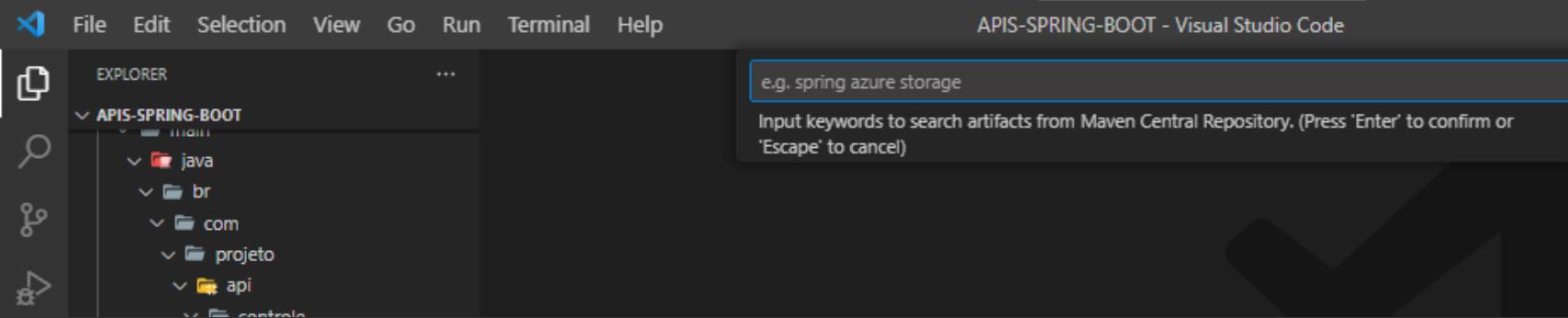


O QUE É O MAVEN?

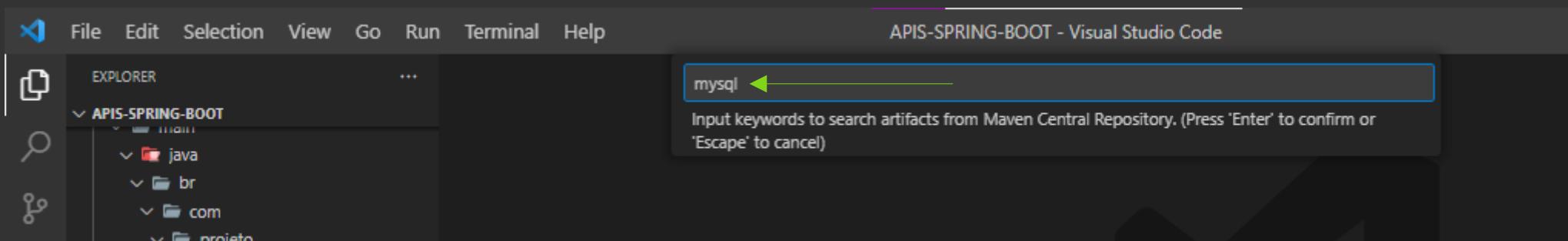
Responsável por gerenciar os pacotes do spring

Click no sinal de +

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

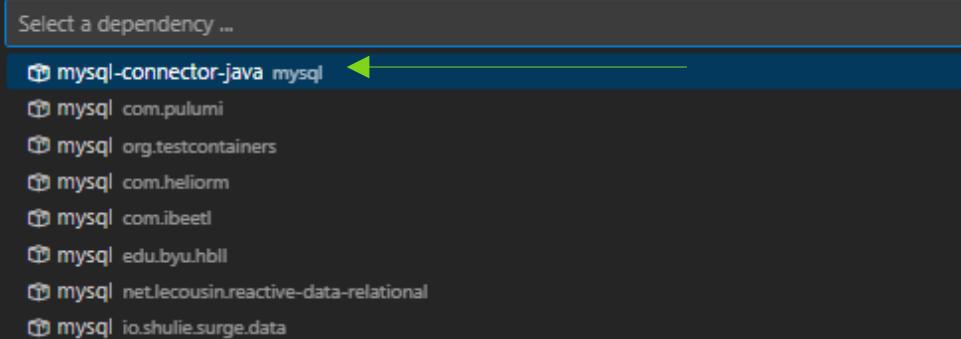
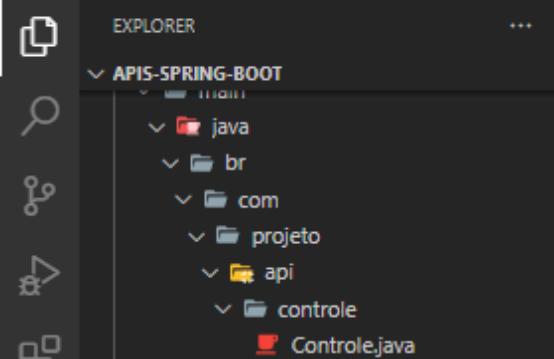


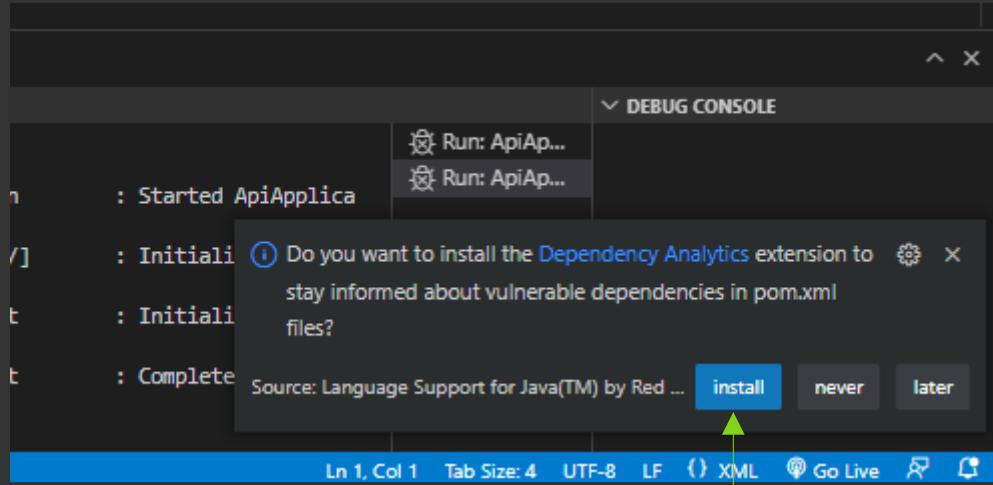
Digite mysql e tecle enter



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

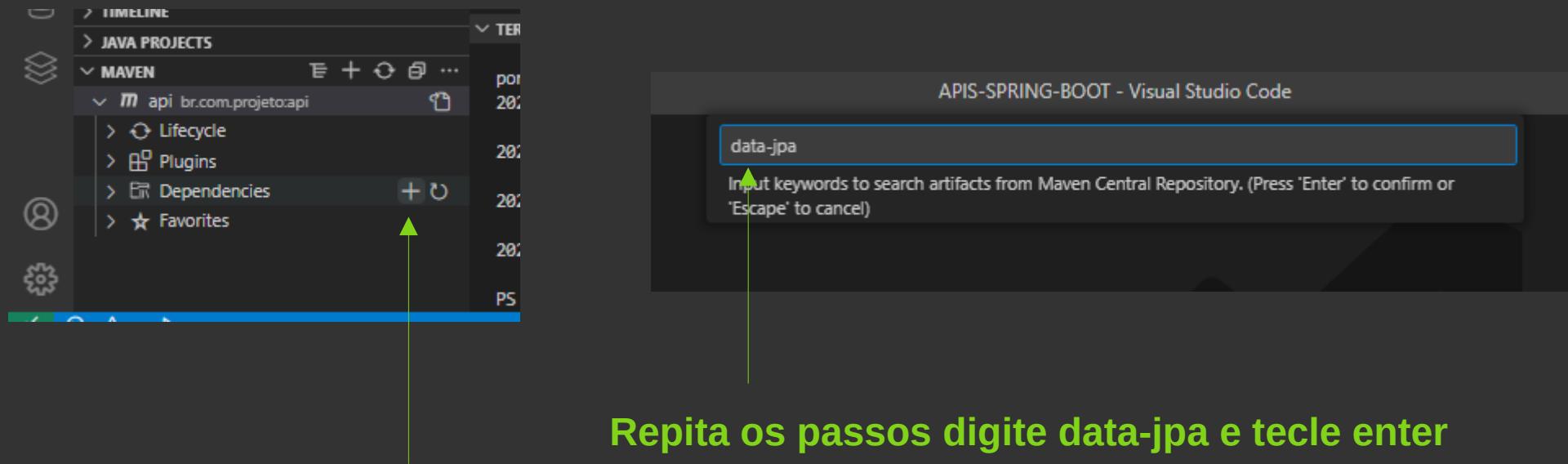
```
port(s): 8080 (http) with context path ''
2023-01-23T11:23:14.611-03:00  INFO 6540 --- [ restartedMain] br.com.projeto.api.ApiApplication
: Started ApiApplication in 11.71 seconds (process running for 14.239)
2023-01-23T11:23:15.963-03:00  INFO 6540 --- [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/]
: 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...

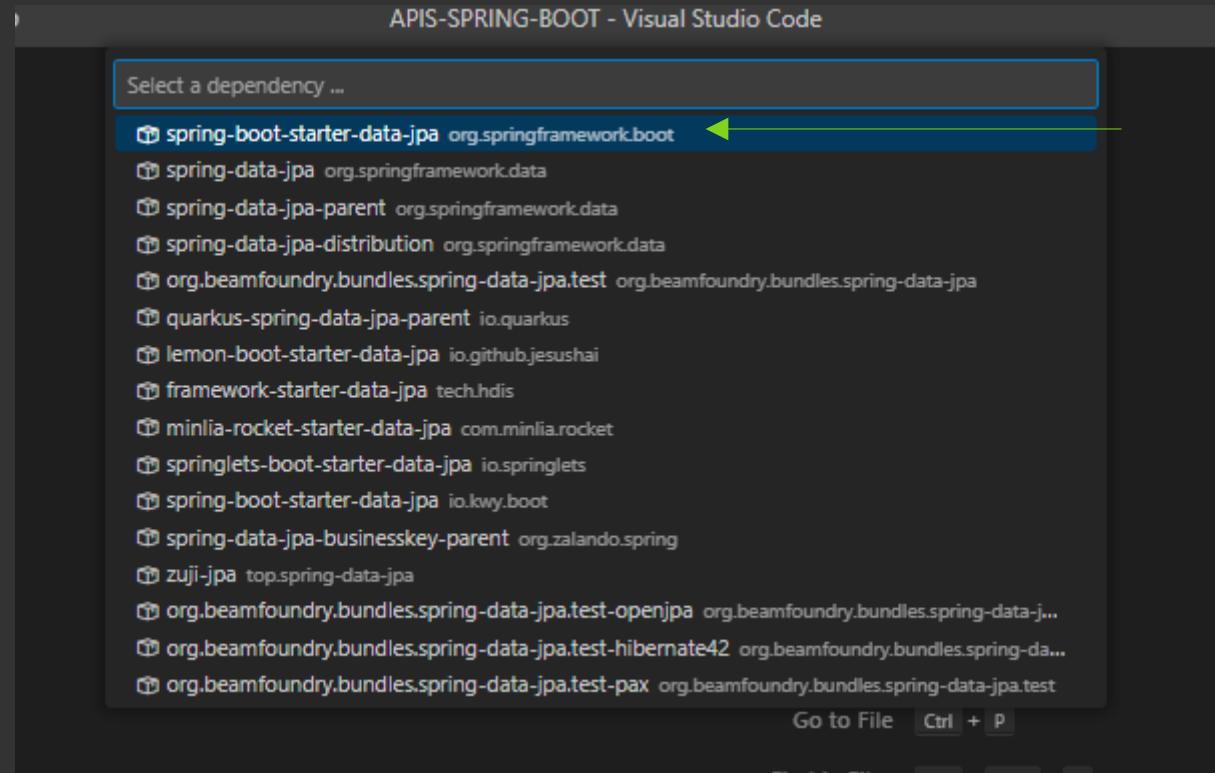
Run: ApiAp...

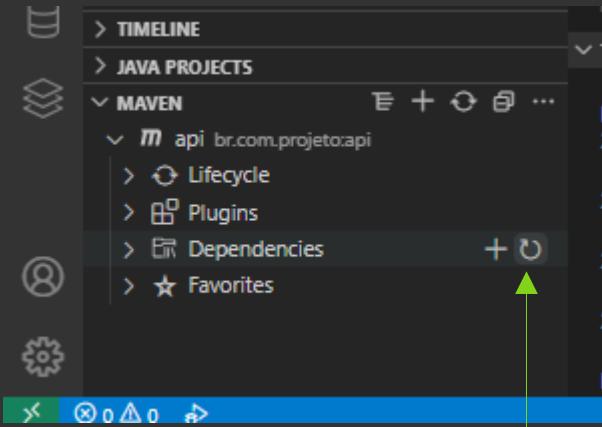
DEBUG CONSOLE

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



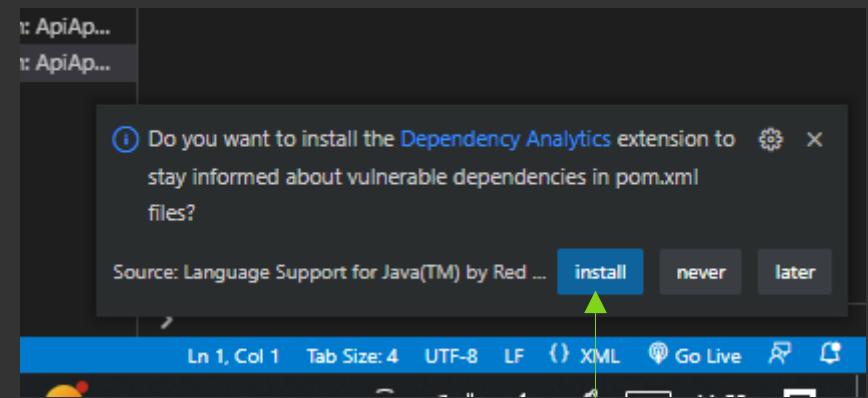
Repita os passos digite **data-jpa** e tecle enter





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
PS E:\APIS-SPRING-BOOT>

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>
```

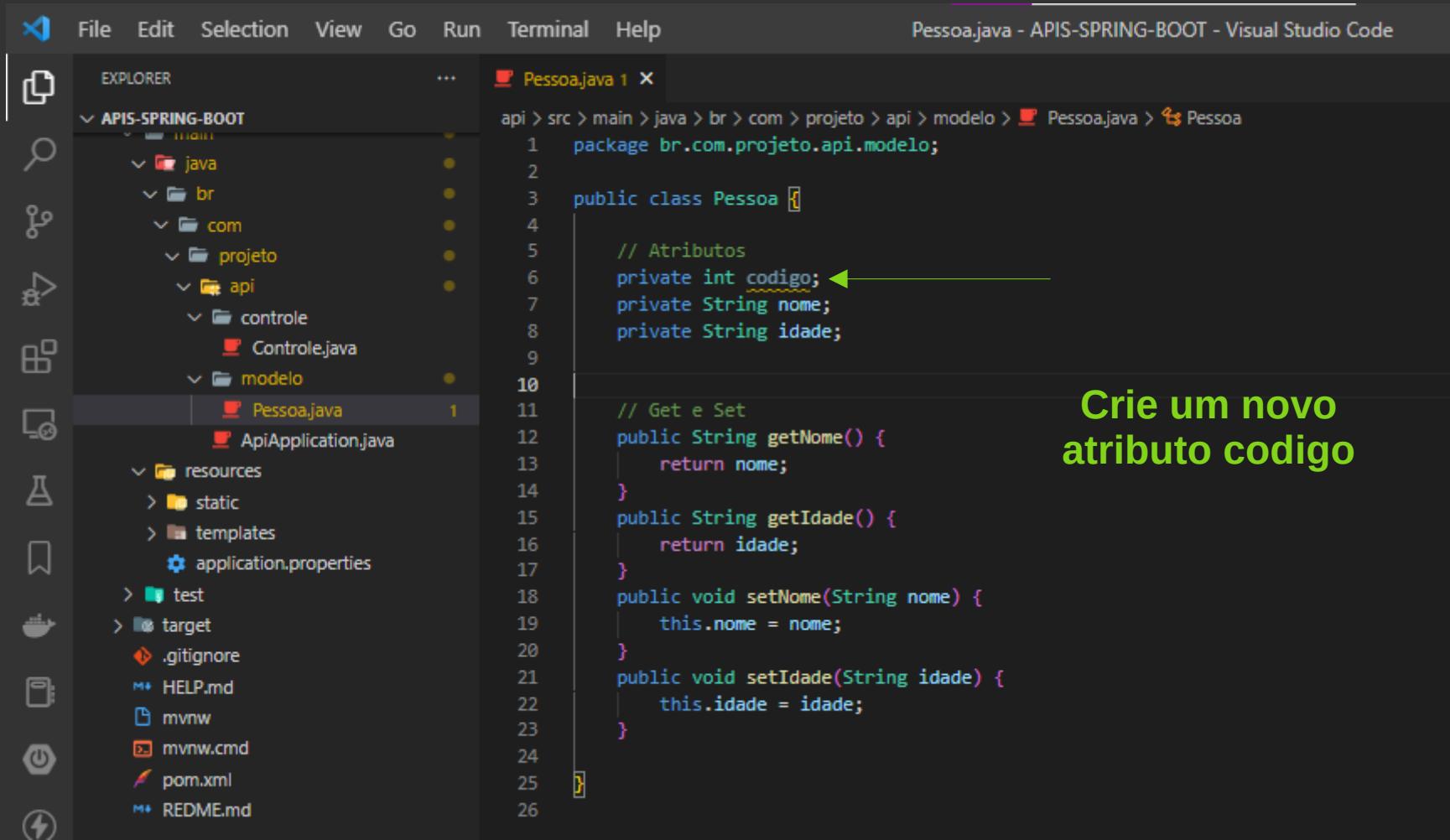
Adicionando assim temos ela instalada

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.
- Active Editor:** Pessoajava - APIS-SPRING
- Explorer:** 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:** The file Pessoajava.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:** A "Quick Fix..." menu is displayed 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' (highlighted)
 - 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:

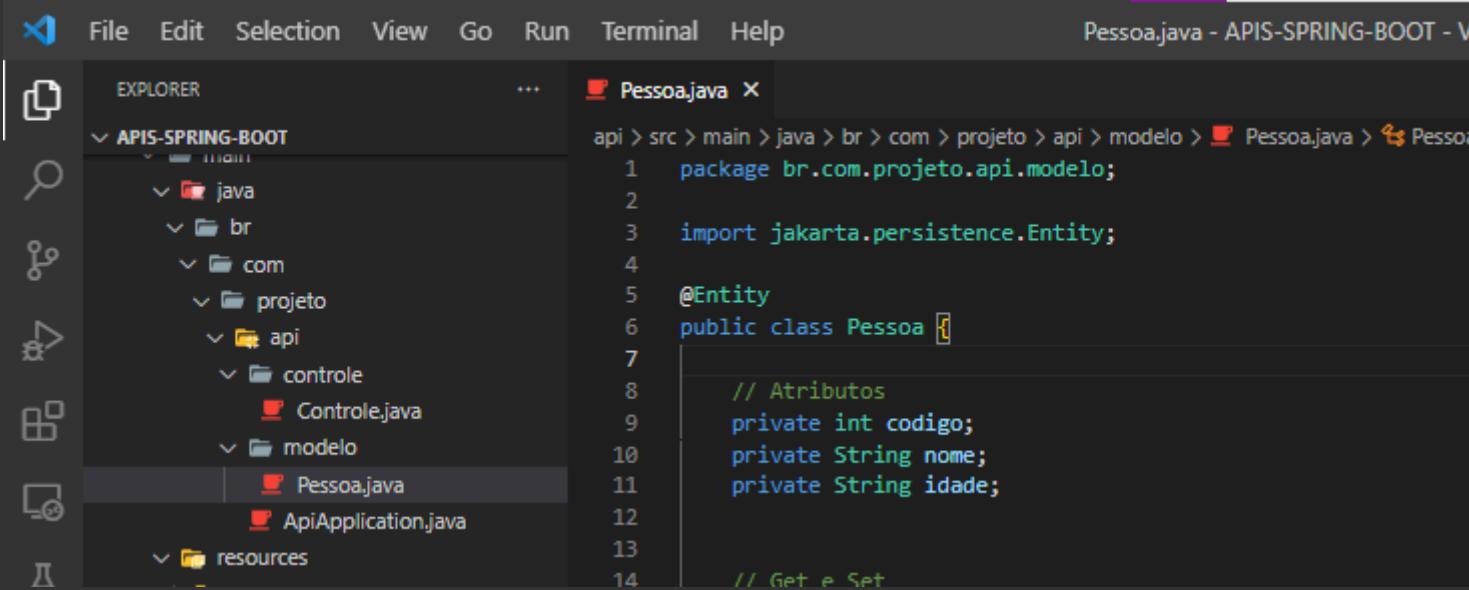
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `Pessoa.java` from the `APIS-SPRING-BOOT` project. The code is as follows:

```
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:** The file Pessoa.java is open, showing the following 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
13     private org.springframework.data.annotation.Id id - org.springframework.data.annotation
14     private jakarta.persistence.IdClass idClass - jakarta.persistence
15         .IdGeneratorType - org.hibernate.annotations
16         .RowId - org.hibernate.annotations
17         .Index - org.hibernate.annotations
18     public jakarta.persistence.Index index - jakarta.persistence
19     }
20     .Indexed - org.springframework.stereotype
21     public com.fasterxml.jackson.annotation.
```

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 (containing Controle.java)
 - modelo (containing Pessoa.java)
 - resources
 - static
 - templates
 - application.properties
- Code Editor:** The file "Pessoajava.java" is open, showing Java code for an Entity class named "Pessoa". The code includes annotations for Entity, Table, and Id, and defines attributes for codigo, nome, and idade.

```
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 @GeneratedValue
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 `GeneratedValue` 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`.

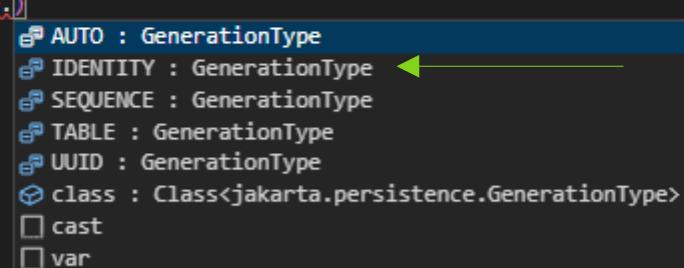
Pessoa.java ●

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 >  Pessoa.java >  Pessoa >  codigo

```
2 import jakarta.persistence.Entity;
3 import jakarta.persistence.GeneratedValue;
4 import jakarta.persistence.GenerationType; ◀
5 import jakarta.persistence.Id;
6 import jakarta.persistence.Table;
7
8
9 @Entity
10 @Table(name = "pessoas")
11 public class Pessoa {
12
13     // Atributos
14     @Id
15     @GeneratedValue(strategy = GenerationType.AUTO) ▾
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     public void setNome(String nome) {
```



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
 resources
 static
 templates
 application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
README.md

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**

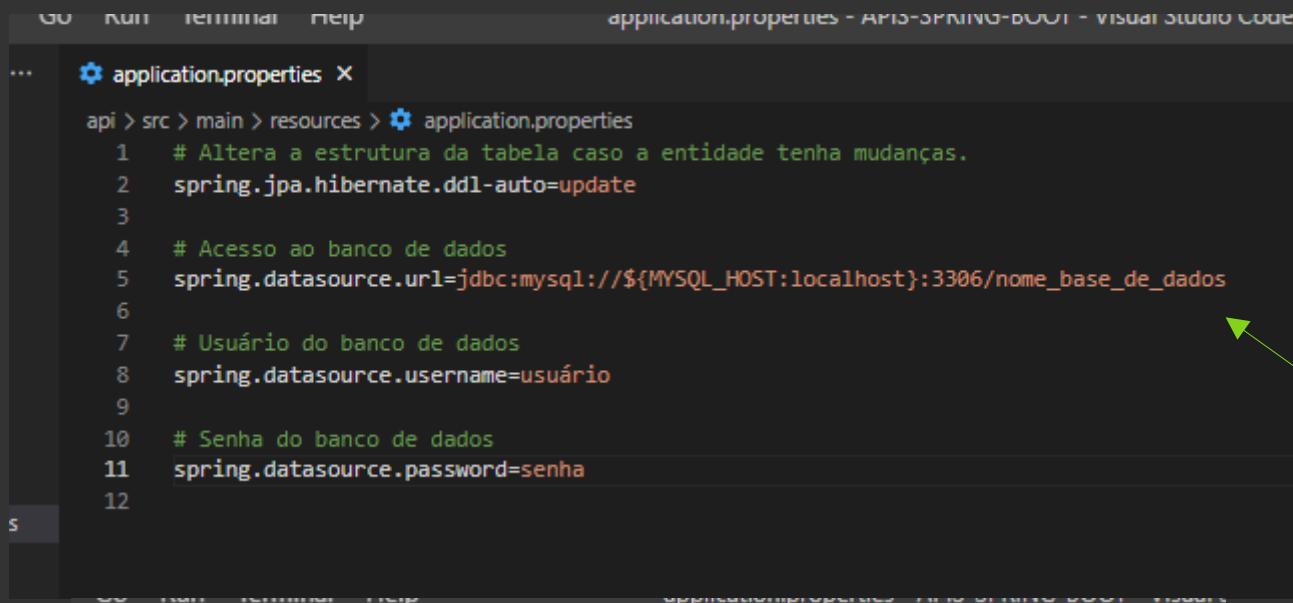
Se o arquivo pom.xml marcar um erro feche o vscode e abra novamente

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
```

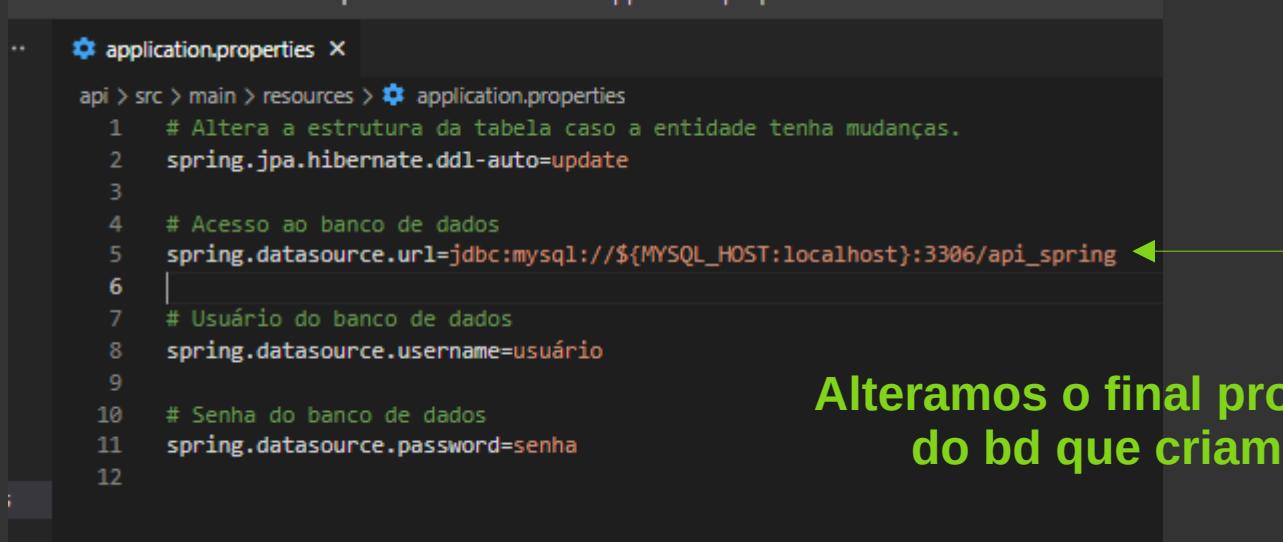
Annotations are present in the bottom right corner of the code editor:

- 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`

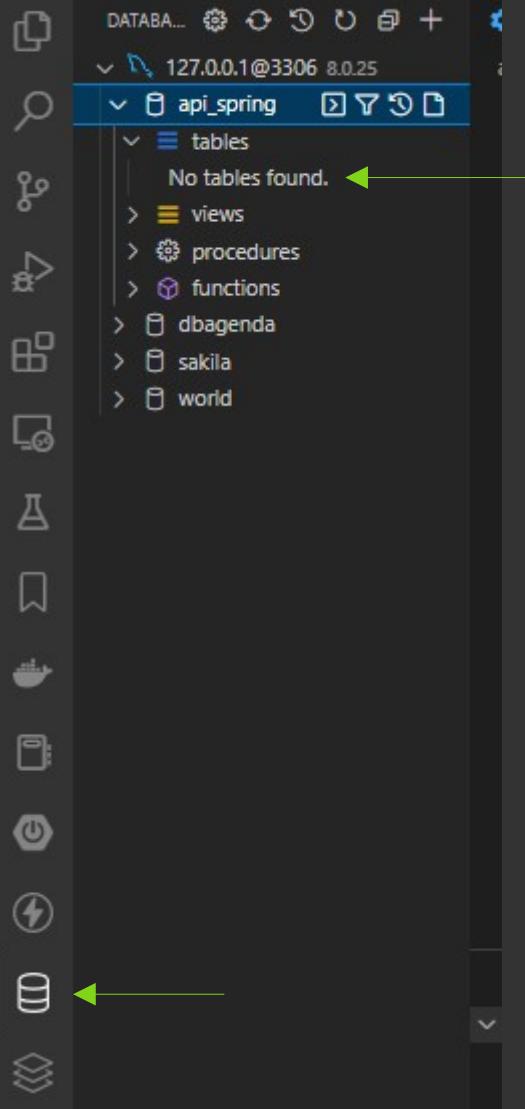
The screenshot shows the Visual Studio Code interface with the following details:

- 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
- Code Editor:** 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

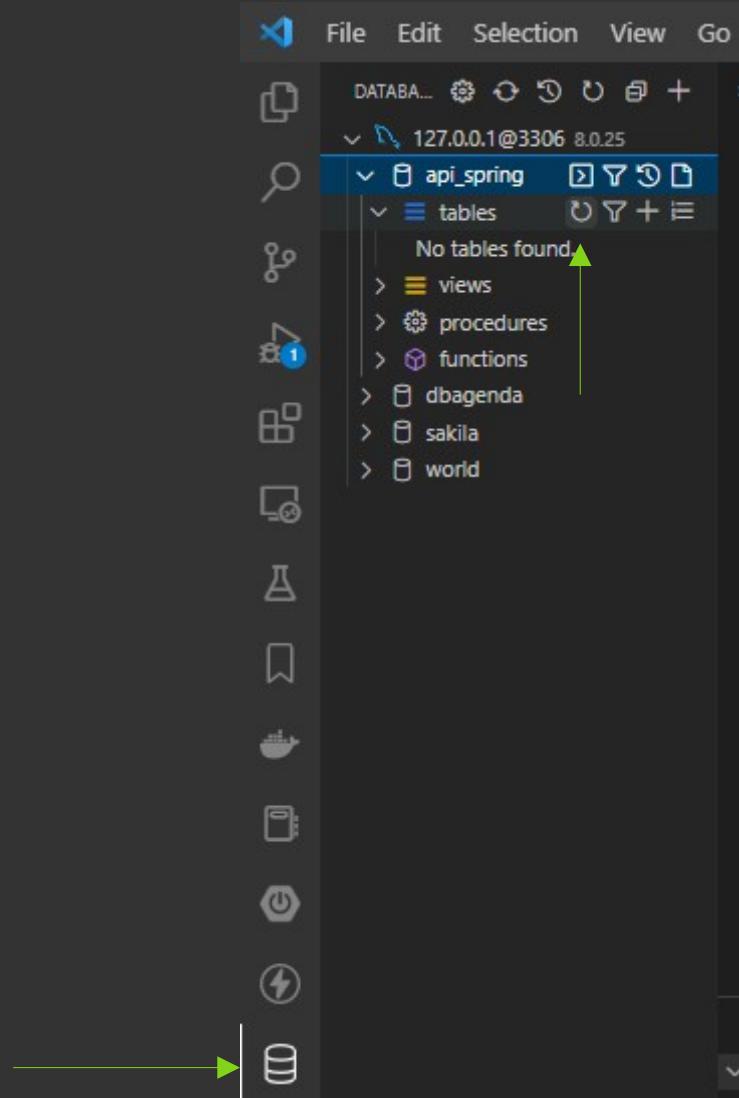


Abra o bad pra ver que ainda não existem tabelas

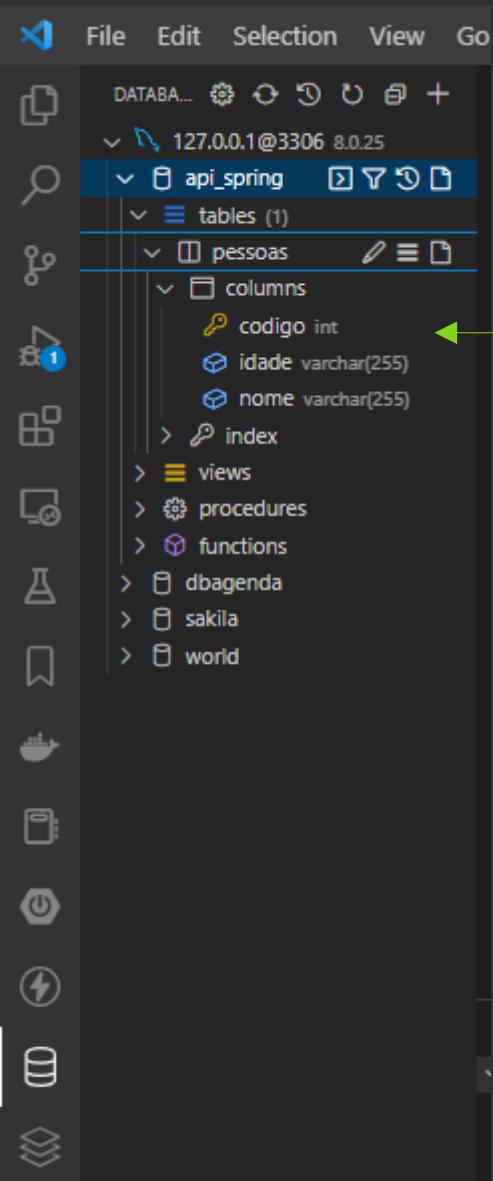
The screenshot shows the Visual Studio Code interface with the following details:

- 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": SPRING BOOT DASHBOARD, ..., APPS, api. The "api" folder is selected.
- Right Editor Area:** Displays the contents 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}: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
```
- Bottom Status Bar:** Shows a green arrow icon pointing right, followed by a power button icon and a lightning bolt icon.
- Center Message:** Execute o projeto (Execute the project).



Depois de executar
atualize refresh



Os dados foram criados

#14
Criando repositorio

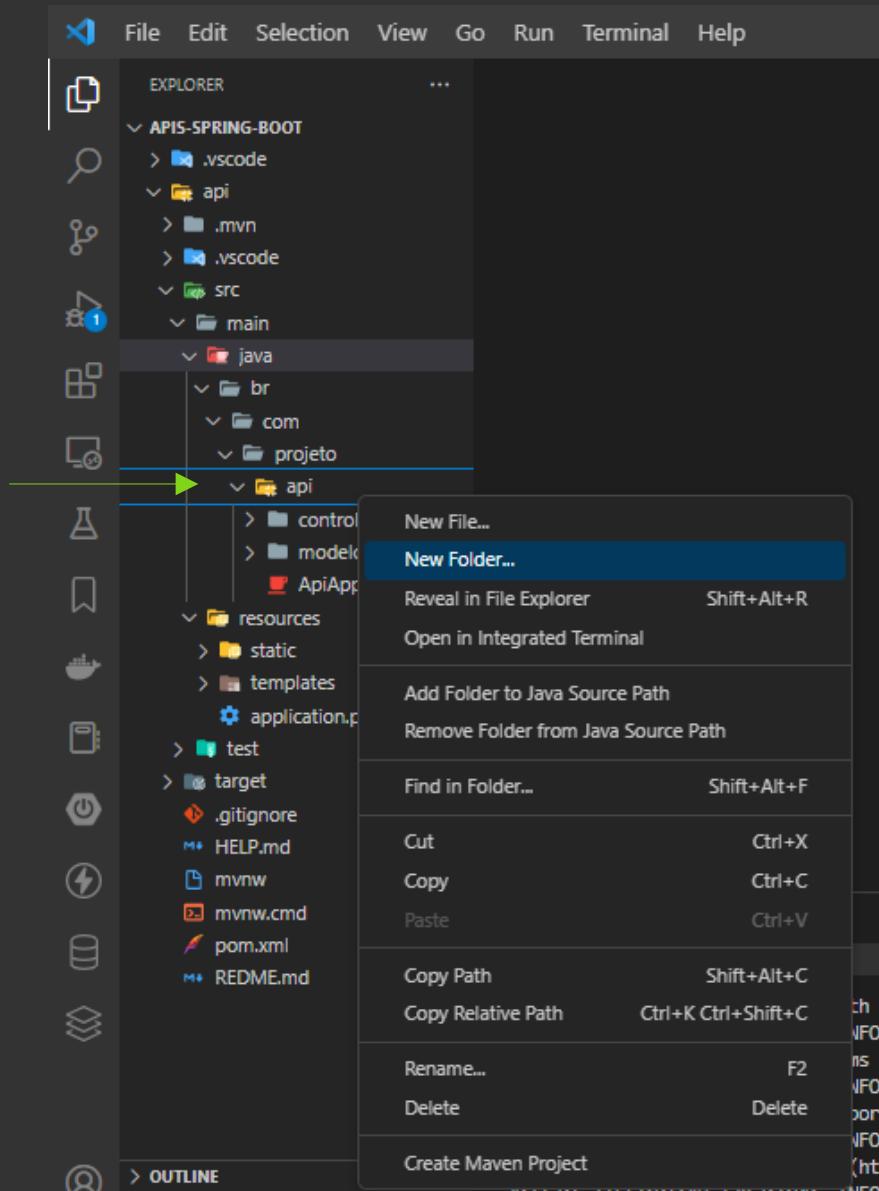
o que é um repositorio?

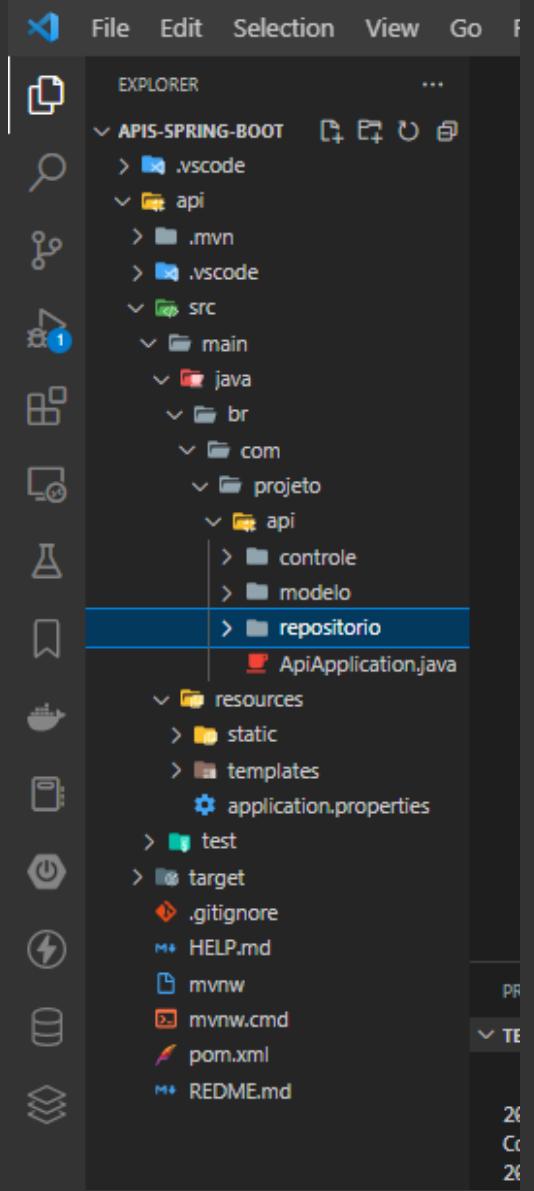
Uma camada de persistencia

O que é uma camada de persistencia?

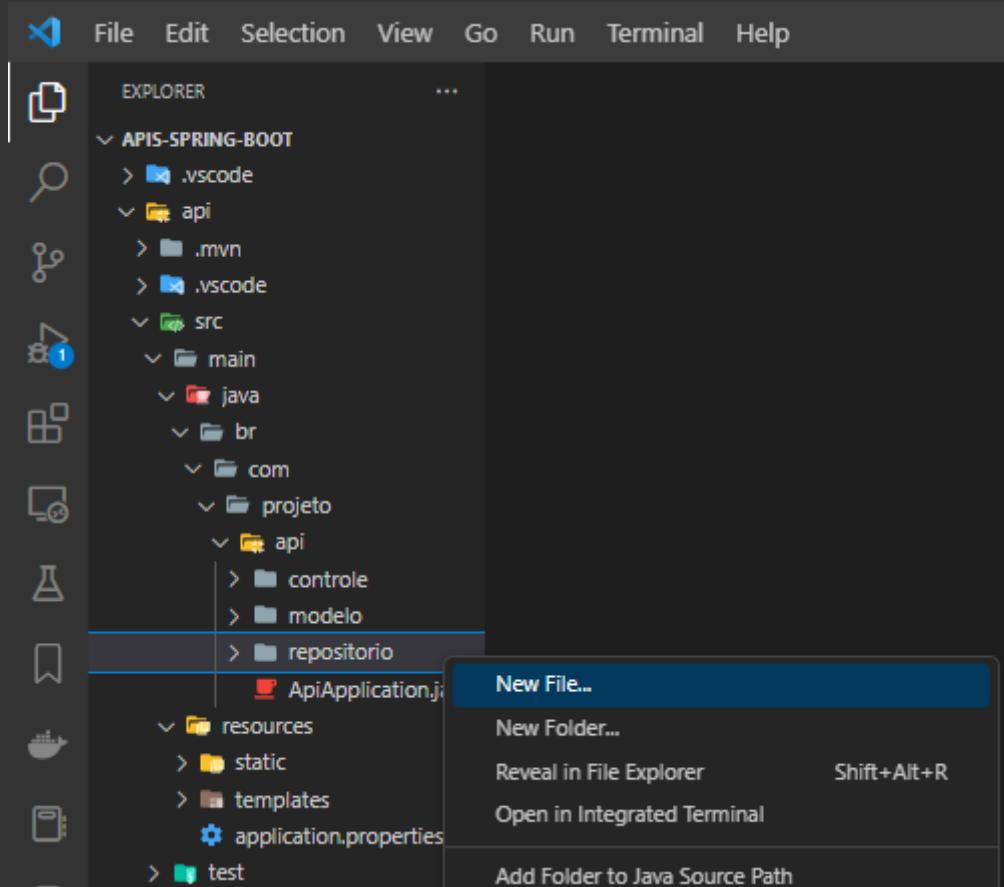
Parte do projeto onde fazemos as ações do banco de dados

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:

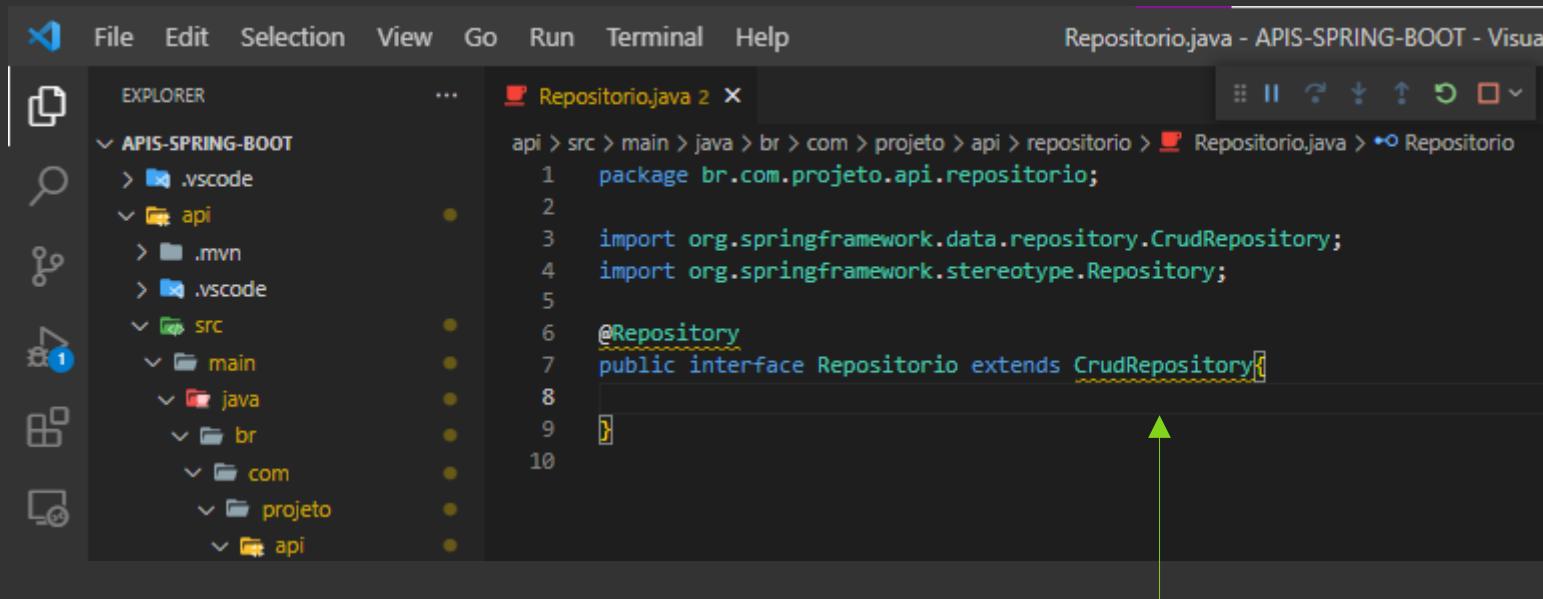
```
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public interface Repositorio {
7
8 }
9
```

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 (with main, java, br, com, projeto, api, controle, modelo, repository), 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 full documentation for the CrudMethodMetadata interface, including the interface itself, its author (Oliver Gierke, Thomas Darimont, Christoph Strobl, Mark Paluch, Jens Schauder, Greg Turnquist), and its purpose: "Interface to abstract CrudMethodMetadata that provide the LockModeType to be used for query execution."

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



The screenshot shows a dark-themed interface of Visual Studio Code. In the top navigation bar, the tabs "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help" are visible. The active tab is "File". The title bar displays "Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code". On the left, the Explorer sidebar shows the project structure: ".vscode", "api", ".mvn", ".vscode", "src", "main", "java", "br", "com", "projeto", and "api". The main editor area contains the following Java code:

```
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<{
8
9 }
10
```

A yellow arrow points upwards from the bottom of the code editor towards the title bar.

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 Explorer:** Shows the project structure under "API-SPRING-BOOT". The "src/main/java" folder contains packages for ".api", "br.com.projeto", and "br.com.projeto.api.repository". Inside "br.com.projeto.api.repository", there are files named "Repositorio.java" and "Repositorio.j...".
- Code Editor:** The file "Repositorio.java" is open. It contains Java code for a repository interface:

```
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<Pessoa>{}
```
- IntelliJ IDEA Quick Documentation:** A tooltip is displayed over the word "Pessoa", listing various related classes from the "org.hibernate" and "org.springframework" namespaces.
- Status Bar:** Shows the path "Repositorio.java - API-SPRING-BOOT - Visual Studio Code".

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.datab...
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:** The file "Repositorio.java" is open, showing Java code for a repository interface.

```
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 log output from a recent run.

```
2023-01-23T13:36:43.358-03:00 INFO 9796 --- [ restartedMain] o.s.b.d.a.OptionalLiveReload
LiveReload server is running on port 35729
```

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.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

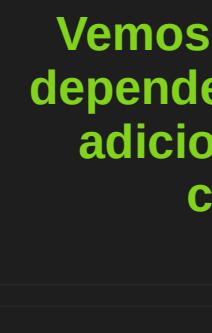
install never later

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

REPORARIO.java 1 pom.xml

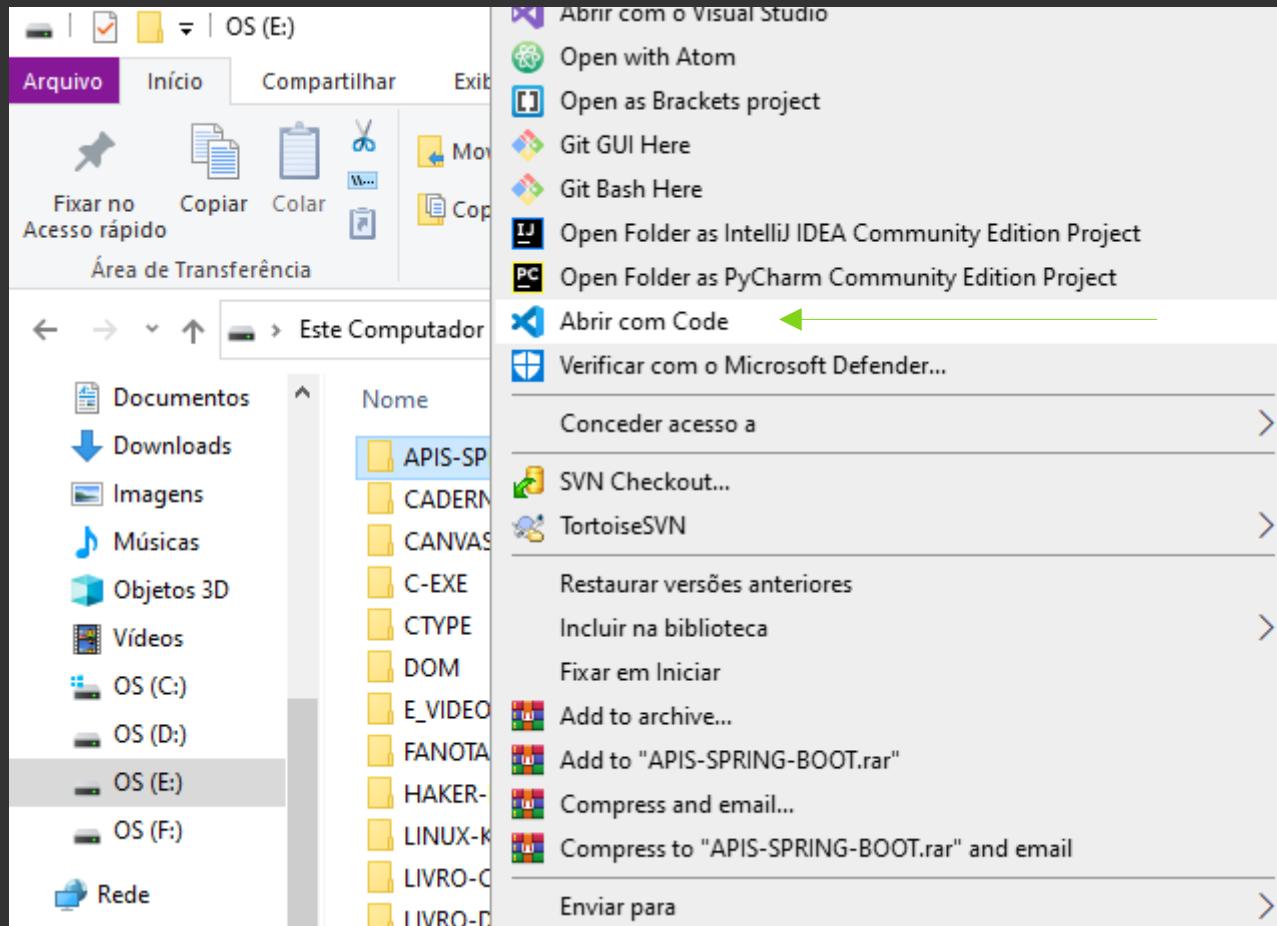
api > pom.xml

```
<?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/> 
  </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>
```



**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



A screenshot of the Visual Studio Code interface. The top bar shows the title "Repository.java - APIS-SPRING-BOOT - Visual Studio Code". The left sidebar contains icons for File, Edit, Selection, View, Go, Run, Terminal, and Help. Below these are sections for EXPLORER, OUTLINE, TIMELINE, JAVA PROJECTS, and MAVEN. The EXPLORER section shows the project structure:

- 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
- REDME.md

- OUTLINE
- TIMELINE
- JAVA PROJECTS
- MAVEN

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

```
api > src > main > java > br > com > projeto > api > repositorio > Repository.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 }
12 |
```

E o erro sumiu voltou o erro

PROBLEMS OUTPUT TERMINAL

TERMINAL

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>

Ln 12, Col 1 Spaces: 4 UTF-8 CRLF { } Java Go Live



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

> test

> target

< .gitignore

< HELP.md

< mvnw

< mvnw.cmd

< pom.xml

< README.md

Repositorio.java X pom.xml

```
api > src > main > java > br > com > projeto > api > repositorio >  Repositorio.java > Language Support for Java
1     package br.com.projeto.api.repositorio;
2
3     import org.springframework.data.jpa.repository.JpaRepository;
4     //import org.springframework.data.repository.CrudRepository;
5     //import org.springframework.stereotype.Repository;
6
7     import br.com.projeto.api.modelo.Pessoa;
8     import jakarta.transaction.Transactional;
9
10    //@Repository
11    @Transactional
12    public interface Repositorio extends JpaRepository<Pessoa, Integer>{
13
14    }
15
```

O único jeito que o erro sumiu até agora
vou deixar os links para futuras pesquisas

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

<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 a Java file named `Controle.java` in the `src/main/java/br/com/projeto/api/controle` package. The code defines a `Controle` class with a `@RestController` annotation. A tooltip is open over the `@Auto` annotation, which is highlighted in red. The tooltip provides information about the `org.springframework.boot.autoconfigure` package and the `AutoConfigureOrder` annotation.

```
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 {

    @Auto
    @GetMapping("/pessoa")
    public Pessoa pessoa(@RequestBody Pessoa nome) {
        return "Seja bem vindo(a) " + nome;
    }
}
```

Explorador (EXPLORER)

- APIS-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - Controle.java
 - resources
 - static
 - templates
 - application.properties
 - test
 - .gitignore
 - HFI P.mrd

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 under APIS-SPRING-BOOT:
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - Controle.java
 - modelo
 - repository
 - Repositorio.java
 - ApiApplication.java
- Editor View:** The file Controle.java is open, showing the following code:

```
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 }
```

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> []

DEBUG CONSOLE

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

Criamos o objeto e adicionamos seu import

```
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
```

**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**

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

EXPLORER Repository.java Controle.java 1

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

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>

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 }

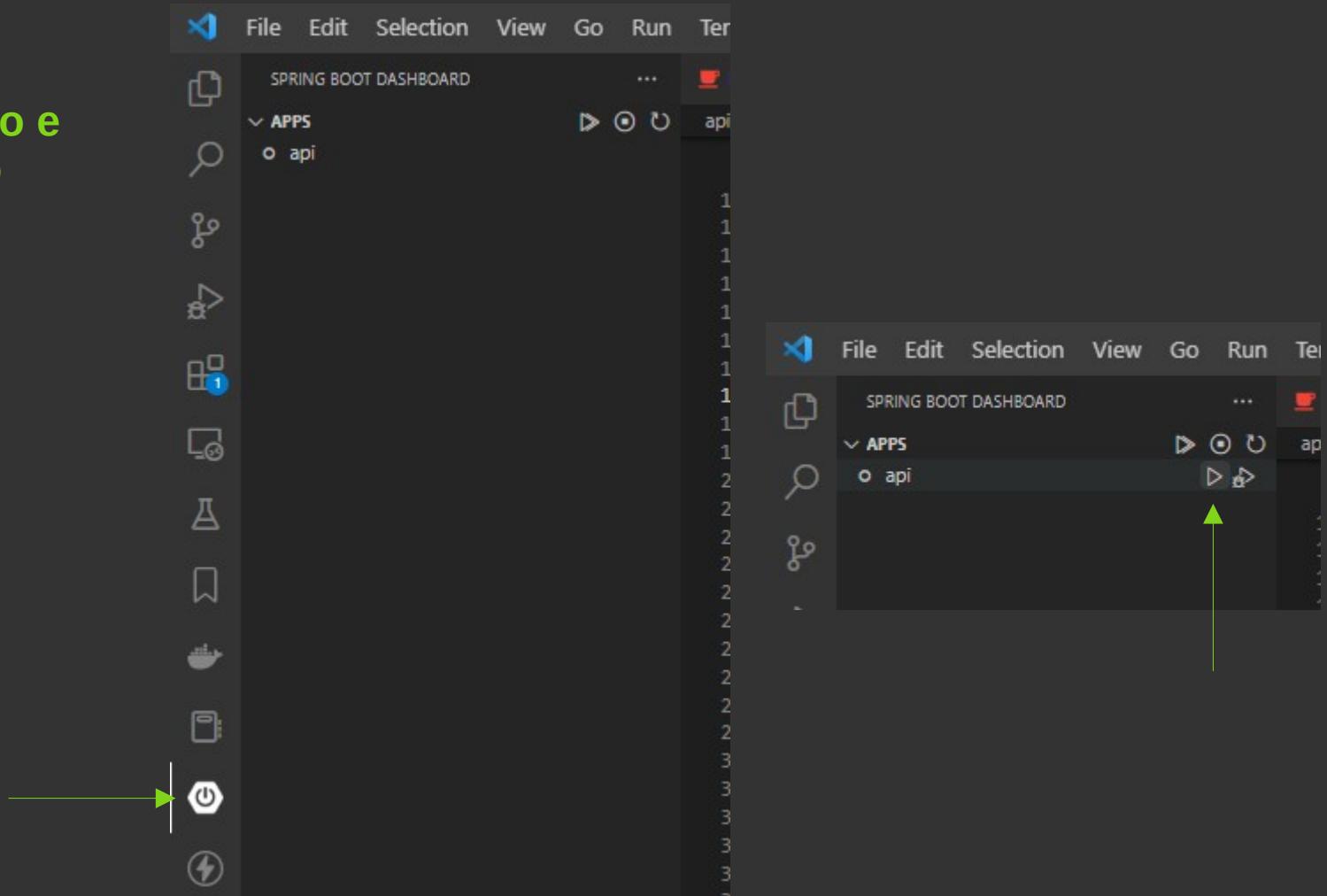
Nesse caso o objeto acao pode chamar esses metodos crud's

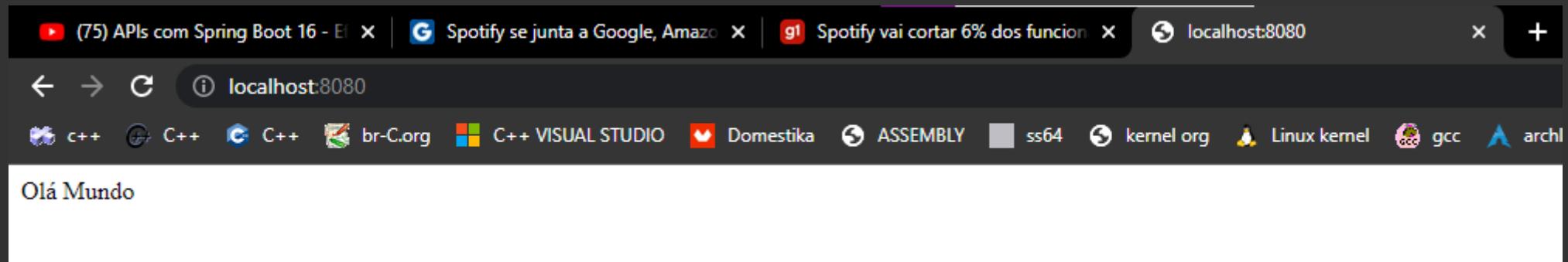
E onde esta a instancia onde esta o new para criar um novo objeto?

O spring mesmo faz isso com a anotatio @AutoWired

#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
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - modelo
 - repositorio
 - 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.
- Right Panel:** A green box contains the text "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 Documentation:** A tooltip for the `@RequestBody` annotation is displayed, 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

EXPLORER

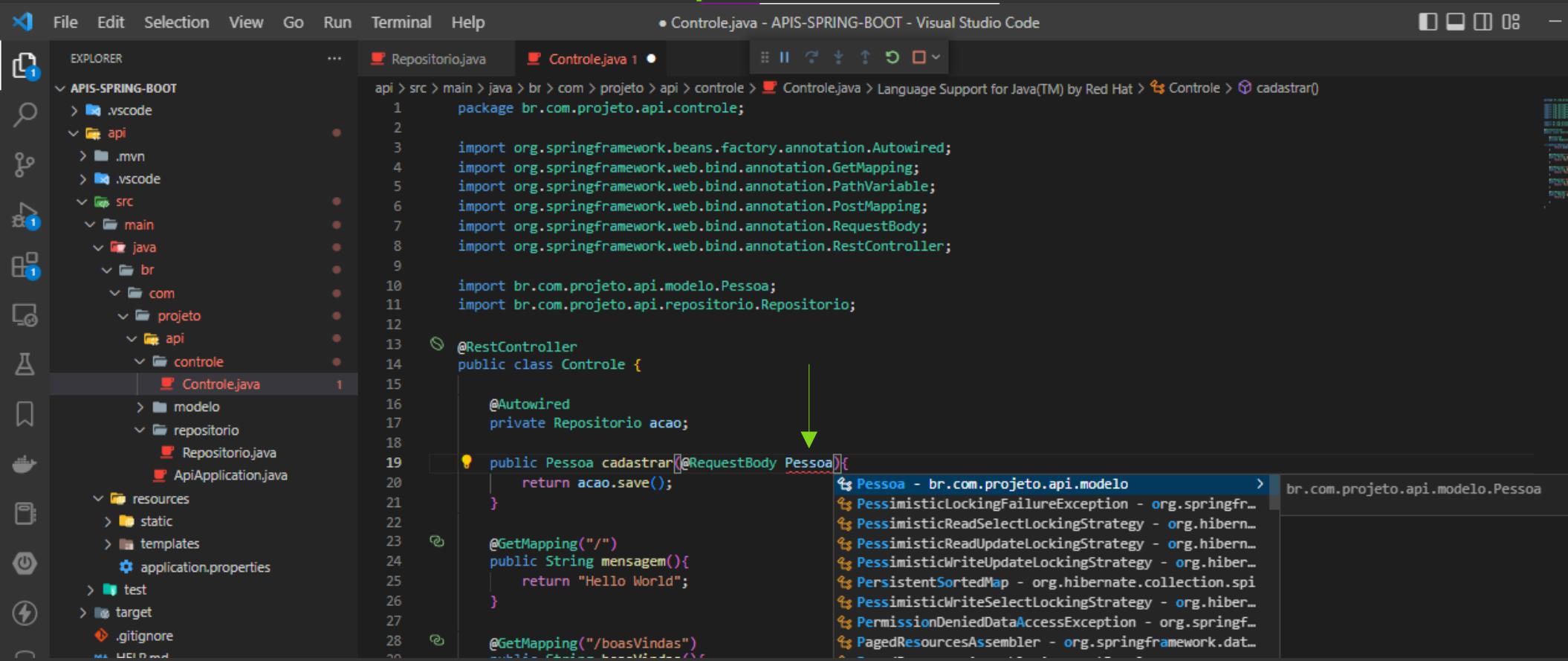
APIS-SPRING-BOOT

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

Repository.java Controle.java 1

```
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
 - 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 }
```

E damos o nome de obj

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
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT:
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - resources
 - static
- Code Editor:** The Controle.java file is open, showing the following code:

```
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 }
```

A green arrow points from the text "O save efetua o cadastramento do objeto do tipo Pessoa" to the closing brace of the method at line 21.

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

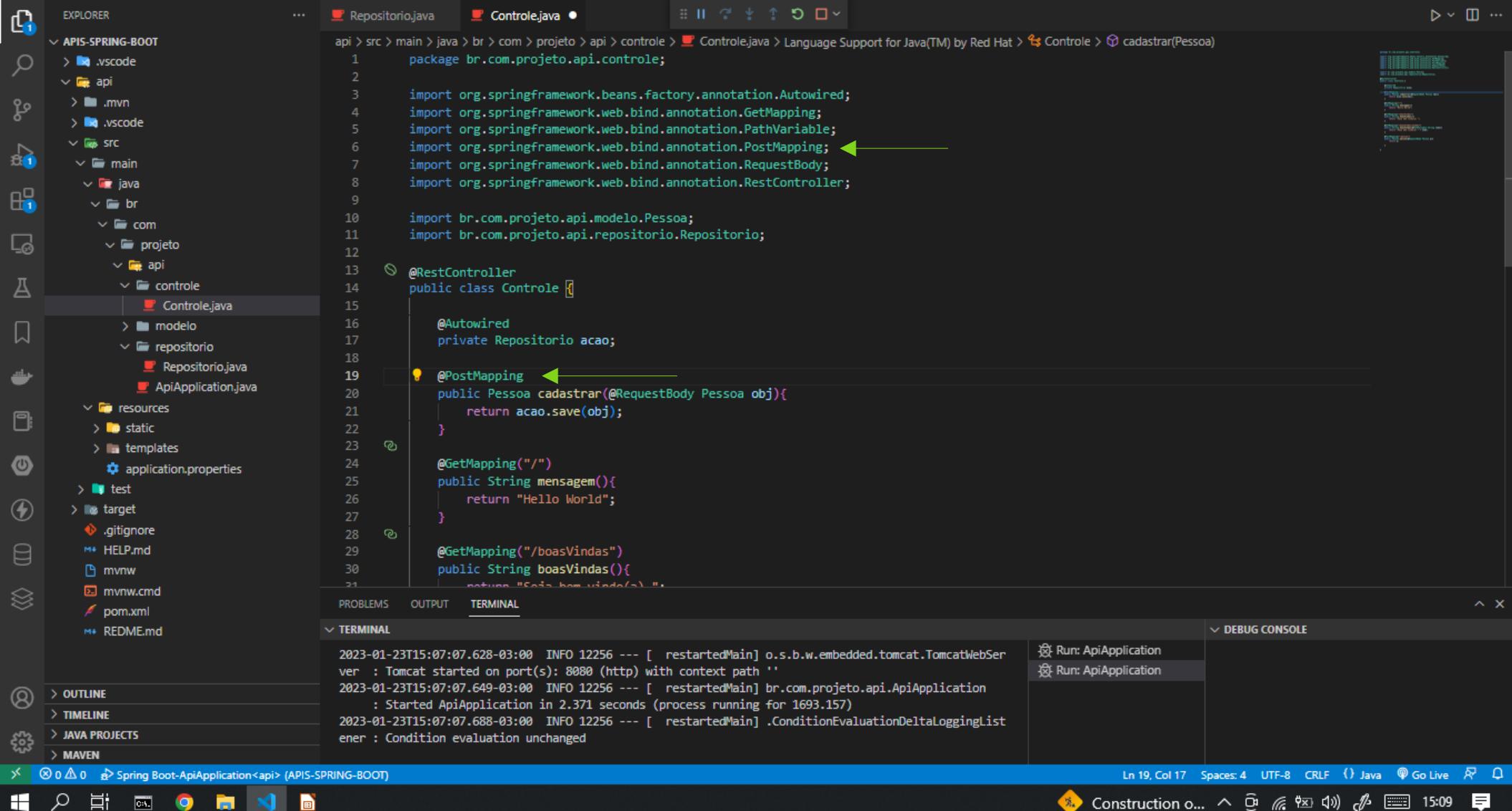
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import 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     @Post
20     publi...
```

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`



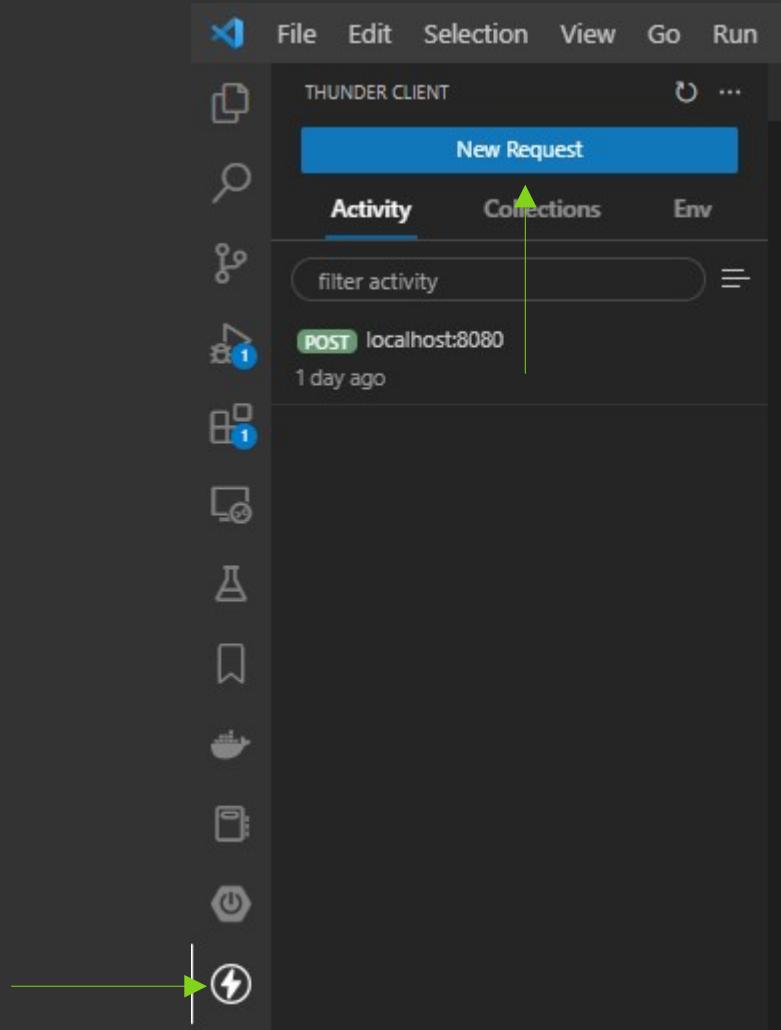
The screenshot shows the Visual Studio Code interface with the following details:

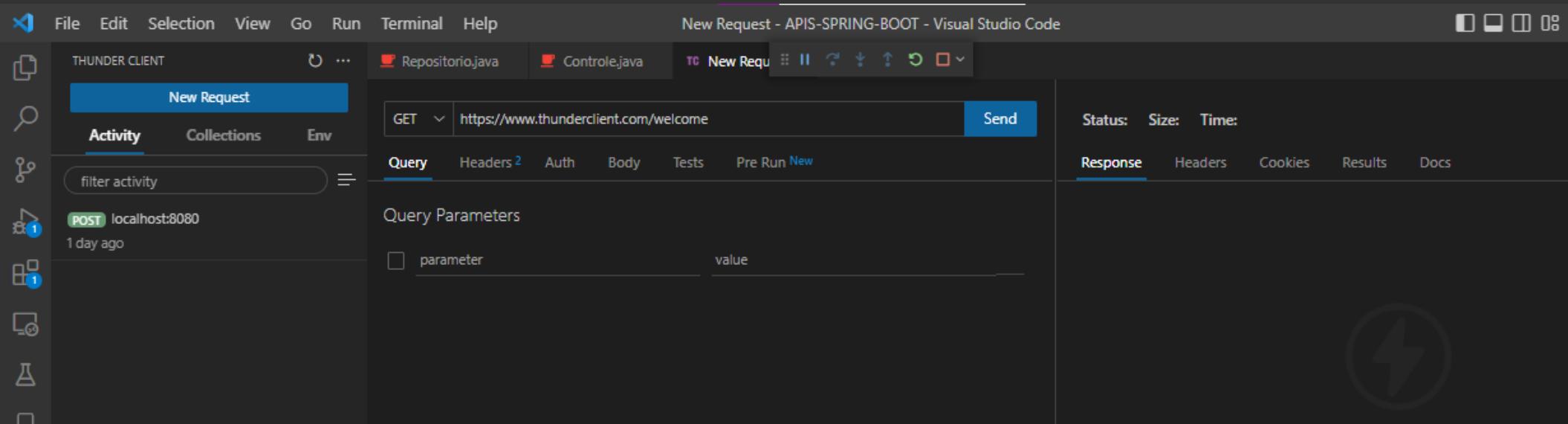
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
 - APIS-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle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

1

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)

Chuva chegando

Go Live

15:15

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 }
```

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/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

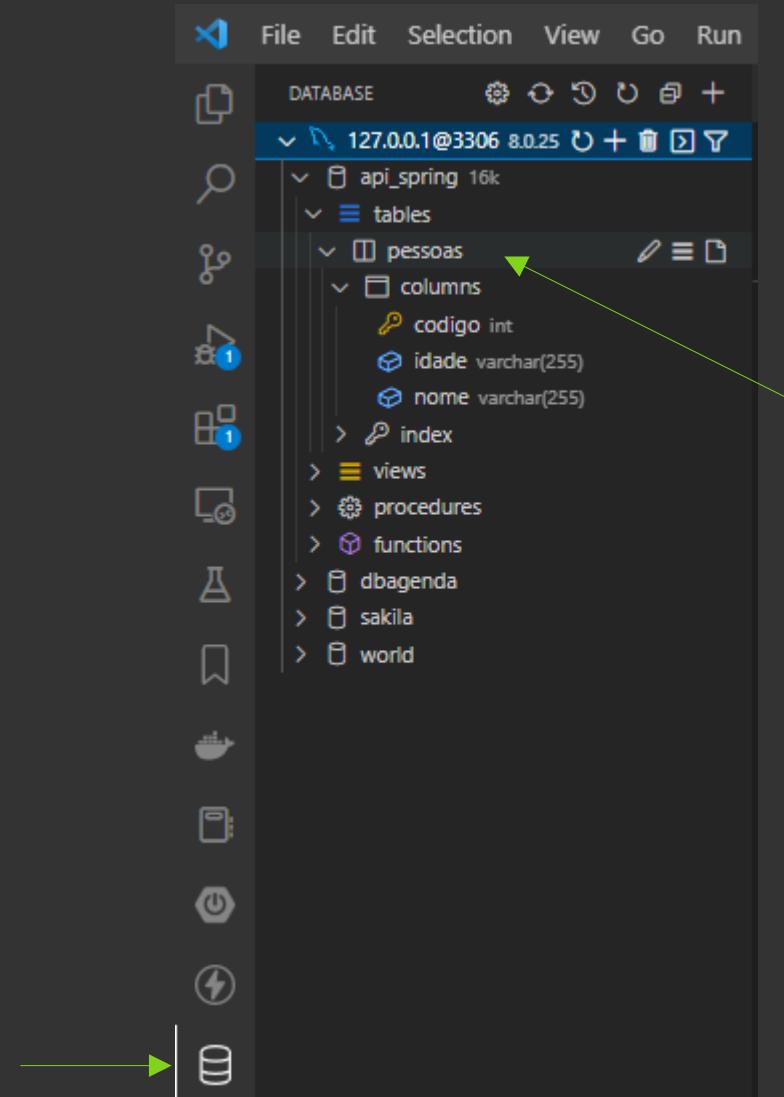
```
2023-01-23T15:20:43.133-03:00 INFO 12256 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat-1].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-01-23T15:20:43.135-03:00 INFO 12256 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
: Initializing Servlet 'dispatcherServlet'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

| | * codigo | idade | nome |
|--|----------|--------------|--------------|
| | int | varchar(255) | varchar(255) |
| | 1 | 1 | 40 |
| | 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. In the code editor, the file `Repositorio.java` is open, showing a Java interface definition:

```
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 Repositorio extends JpaRepository<Pessoa, Integer> {
```

A code completion dropdown is open at the end of the line `<Integer>`, showing various options for the type parameter. The option `List - java.util` is highlighted. To the right of the editor, a tooltip provides detailed information about the `List` interface.

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

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:

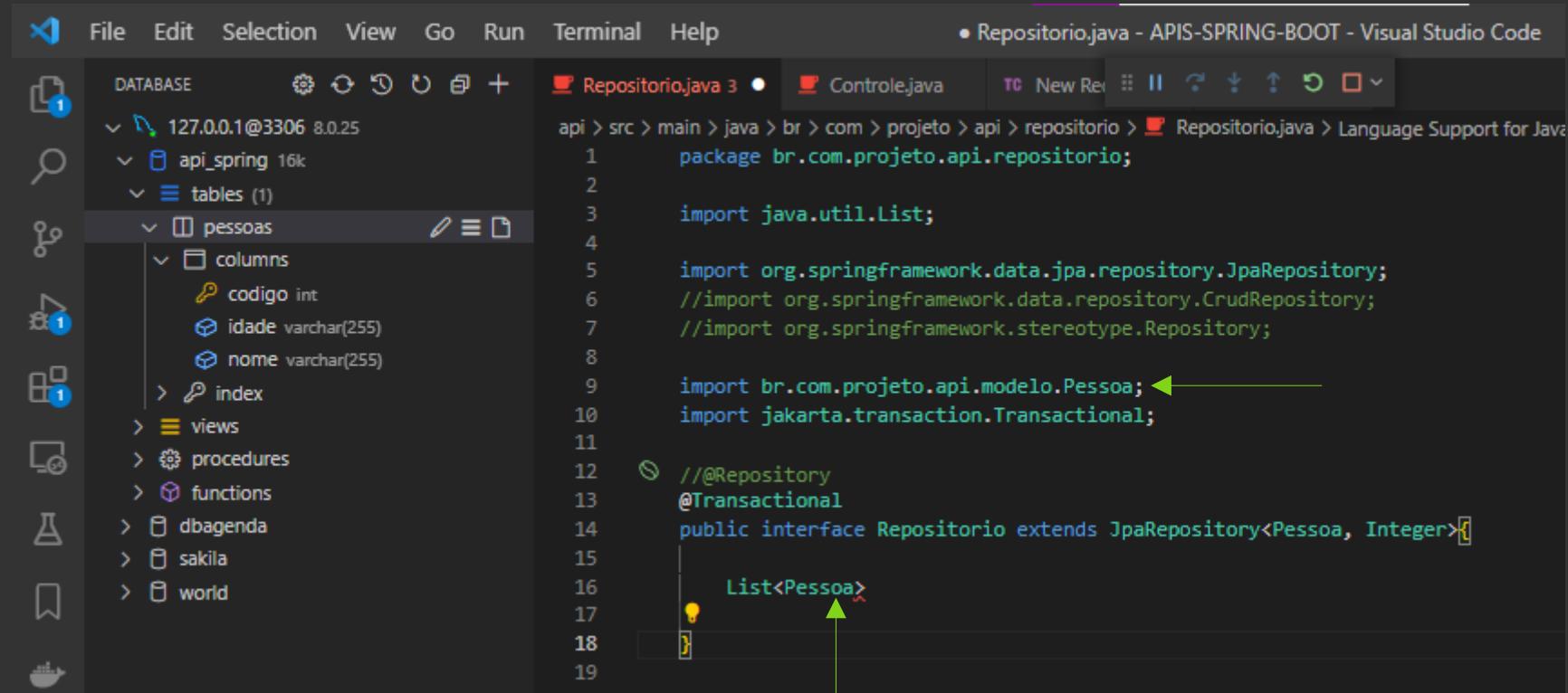
```
• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code
```

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 the Visual Studio Code interface with a dark theme. On the left is a sidebar for a database connection named 'DATABASE' to '127.0.0.1@3306 8.0.25'. Under this connection, there is a schema 'api_spring' containing a table 'pessoas'. The 'pessoas' table has columns 'codigo' (int), 'idade' (varchar(255)), and 'nome' (varchar(255)). Other objects listed include 'index', 'views', 'procedures', 'functions', 'dbagenda', 'sakila', and 'world'. The main editor area displays a Java interface named 'Repositorio.java' with 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> ↑
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 (Database Explorer):** Shows a connection to "127.0.0.1@3306" (8.0.25) and the schema "api_spring" (16k). It lists tables (1), specifically "pessoas". Under "pessoas", it shows columns: "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). It also shows an index, views, procedures, functions, and other tables like "dbagenda", "sakila", and "world".
- Code Editor:** The file "Repositorio.java" is open. The code defines a public interface "Repositorio" that extends "JpaRepository<Pessoa, Integer>". It contains a single method "findAll()".

```
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 }
```

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] INFO o.s.boot.SpringApplication - Started ApiApplication 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

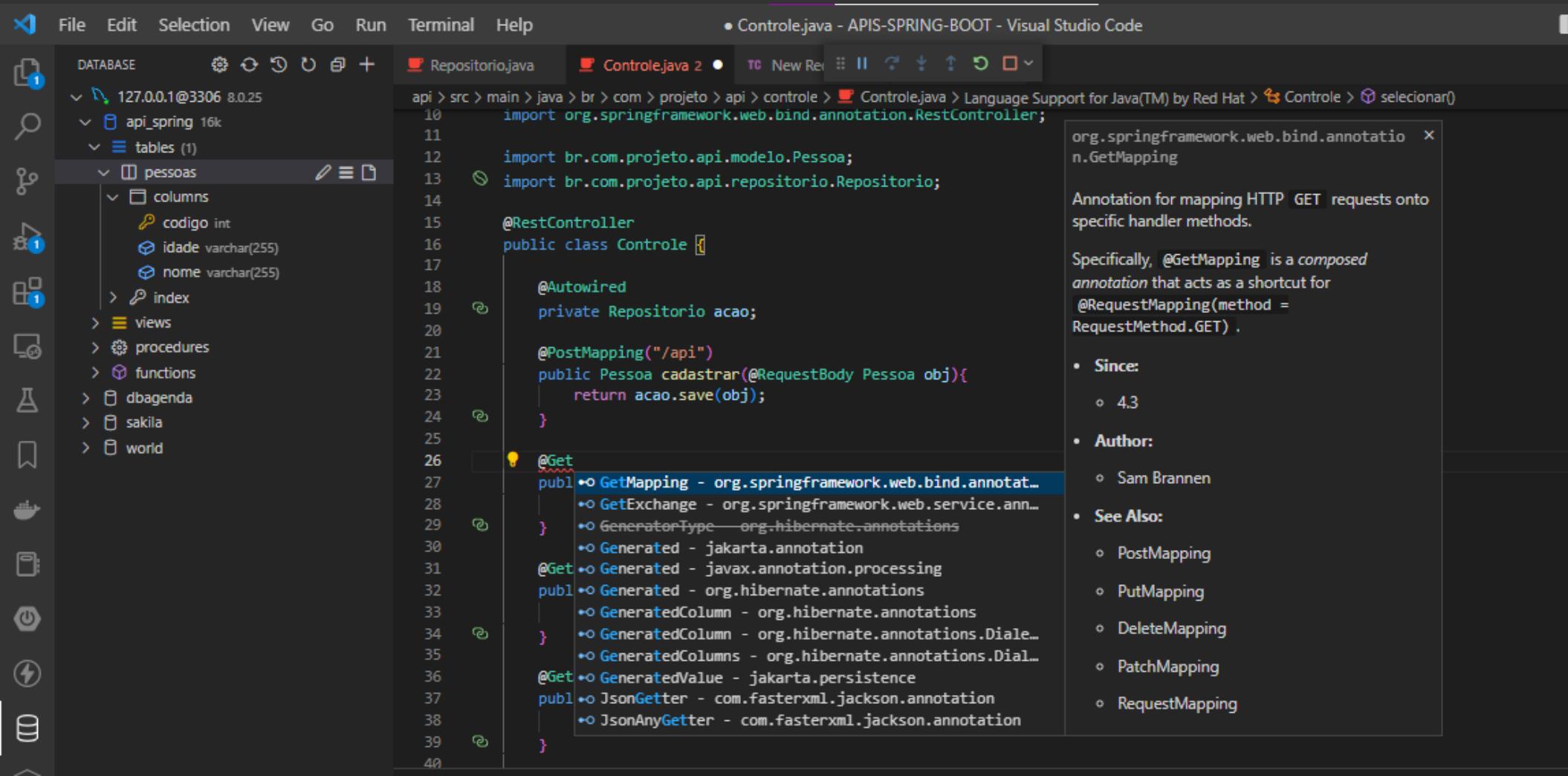


api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM)

```
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
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 Req...

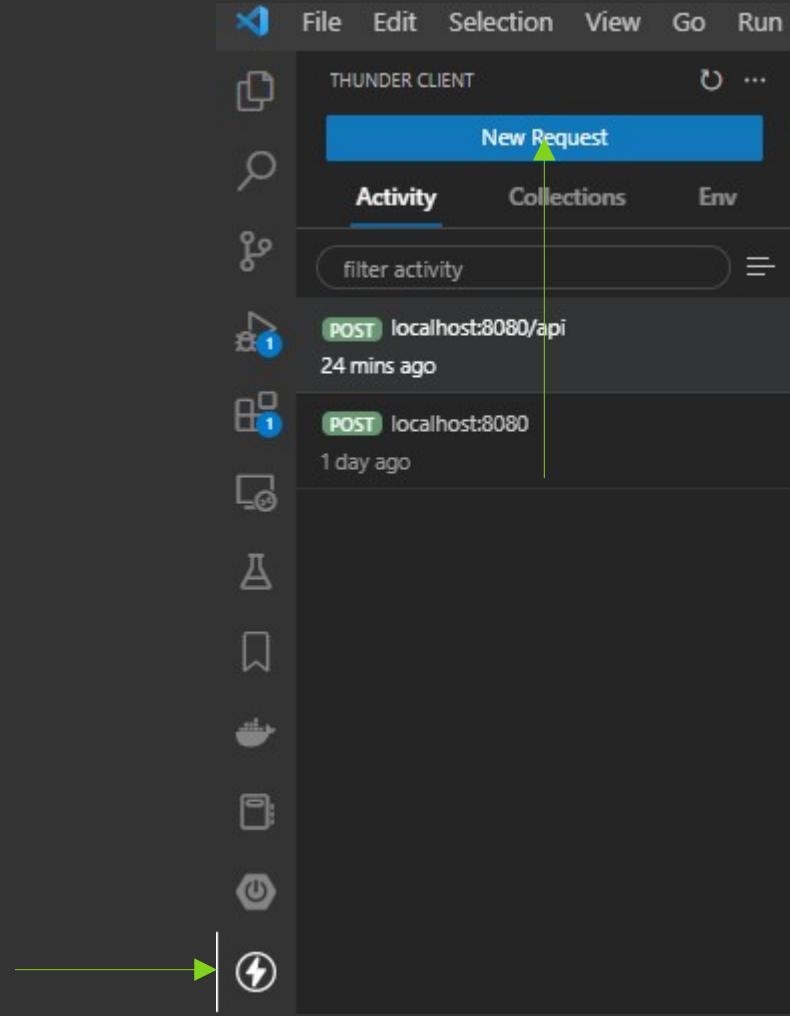


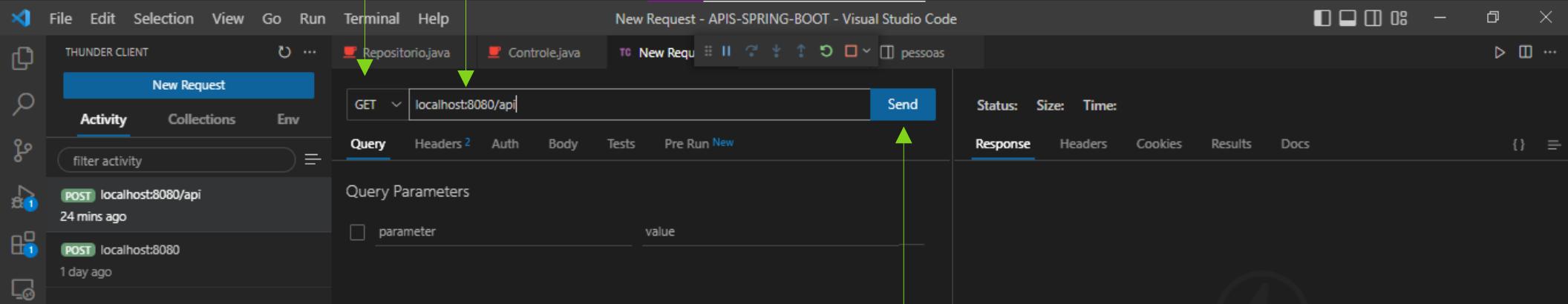
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support

```
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 ]
```

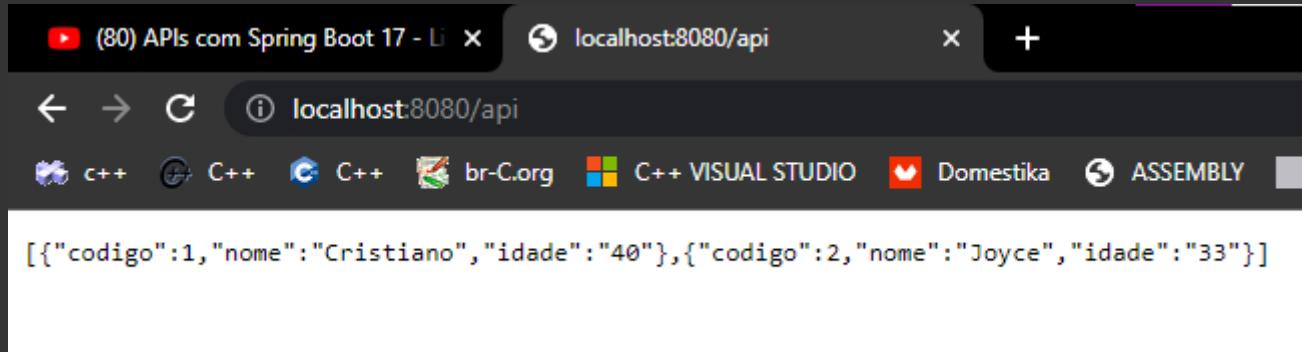
Dados passados

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".
 - API folder contains .vscode and .mvn.
 - src folder contains main, repository, and ApiApplication.java.
 - main folder contains java, br, com, projeto, api, controle, and modelo.
 - java folder contains Pessoa.java, Controle.java, and Repository.java.
- Terminal (Top Right):** Shows the command "Pessoajava - APIS-SPRING-BOOT - Visu".
- Code Editor (Right):** Displays the code for 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;
```

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

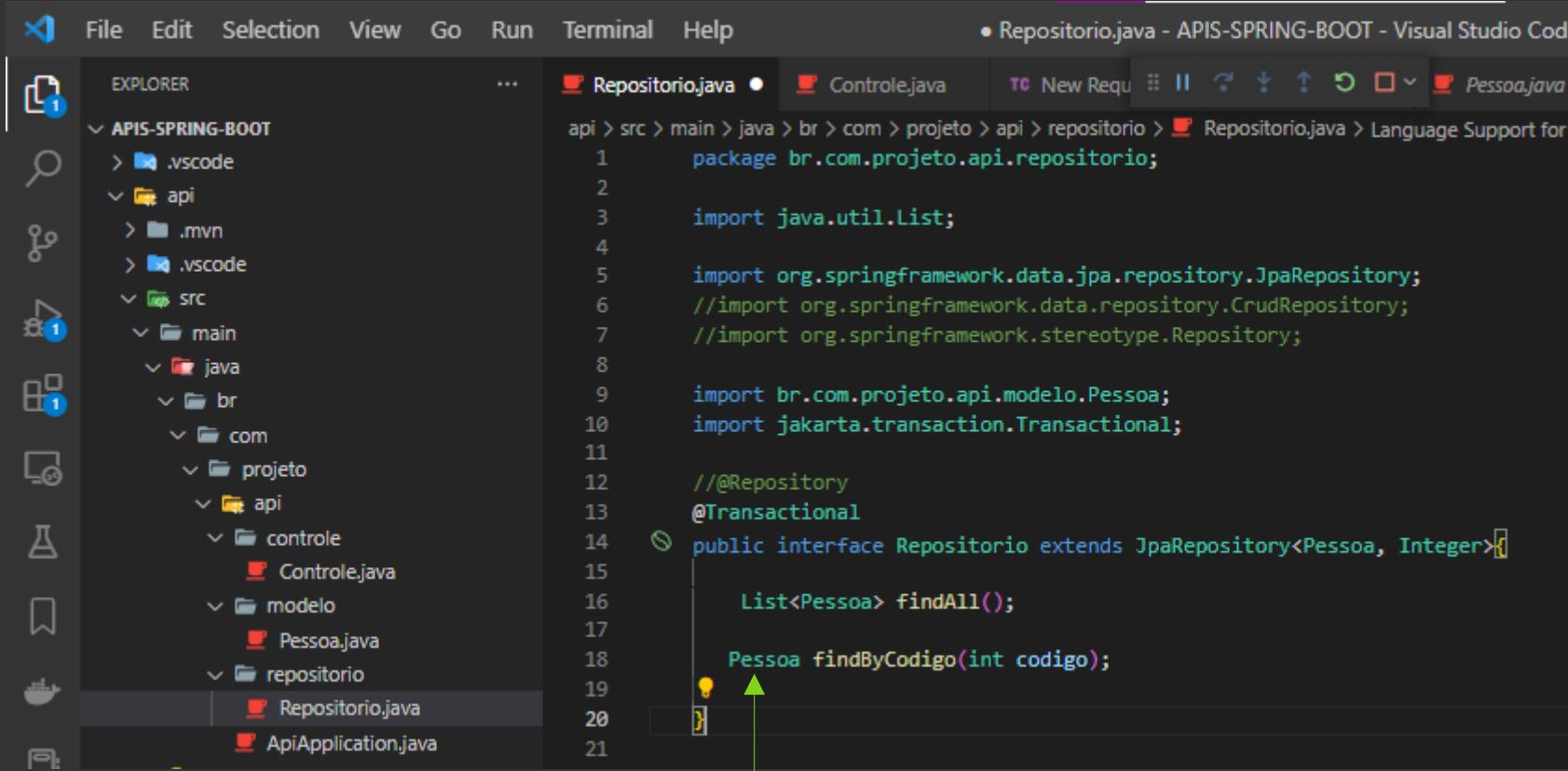
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure for "APIS-SPRING-BOOT". The "src" folder contains "main" and "java". The "java" folder has packages "br.com.projeto.api", "br.com.projeto.modelo", and "br.com.projeto.repository". Inside "repository", there are files "Repositorio.java" and "ApiApplication.java".
- Code Editor:** The main area displays the content of "Repositorio.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 Repositorio extends JpaRepository<Pessoa, Integer>{
15
16     List<Pessoa> findAll();
17
18     findByCodigo(int codigo);
19
20 }
```

A yellow lightbulb icon is positioned above the method signature "findByCodigo(int codigo);". Two green arrows point upwards from the bottom of the screen towards this lightbulb icon.
- Terminal:** At the top right, there is a terminal tab labeled "Terminal".
- Status Bar:** At the very bottom, there is a status bar showing "Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code".

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. The code defines a public interface Repository that extends JpaRepository<Pessoa, Integer>. It includes methods for findAll() and findByCodigo(int codigo). A yellow lightbulb icon with a green arrow points to the closing brace '}' at the bottom of the code editor.
- Terminal:** Not visible in the screenshot.
- Status Bar:** Shows 'New Req' and other status indicators.

```
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 }
```

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.java 2
 - > resources
 - > test
 - > target
 - ↳ .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - REDME.md

REPOSITORY

TERMINAL

CONTROLE

API

PESSOA

REPOSITORY

MAIN

BR

COM

PROJETO

API

CONTROLE

CONTROLE.JAVA

PESSOA.JAVA

REPOSITORY.JAVA

APPLICATION.JAVA

```
api > src > main > java > br > com > projeto > controle > Controle.java
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         return "Seja";
42
43     @GetMapping("/ex")
44     public String ex @exists(Example<S> example) : boolean
45         return "Existe";
46
47     @GetMapping("/all")
48     public List<Pessoa> getAll() : List<Pessoa>
49
50     @GetMapping("/count")
51     public long count() : long
52
53     @GetMapping("/exists")
54     public boolean exists(@Example<S> example) : boolean
55
56     @GetMapping("/existsId")
57     public boolean existsById(Integer id) : boolean
58
59     @GetMapping("/findAll")
60     public List<Pessoa> findAll() : List<Pessoa>
61
62     @GetMapping("/getOne")
63     public Pessoa getOne(@Example<S> example) : Pessoa
64
65     @GetMapping("/getById")
66     public Pessoa getById(@Example<S> example) : Pessoa
67
68     @GetMapping("/getReference")
69     public S getReference(@Example<S> example) : S
70
71     @GetMapping("/getReferenceById")
72     public S getReferenceById(@Example<S> example) : S
73
74     @GetMapping("/getReferenceCount")
75     public long getReferenceCount(@Example<S> example) : long
76
77     @GetMapping("/getReferenceById")
78     public S getReferenceById(@Example<S> example) : S
79
80     @GetMapping("/getReferenceById")
81     public S getReferenceById(@Example<S> example) : S
82
83     @GetMapping("/getReferenceById")
84     public S getReferenceById(@Example<S> example) : S
85
86     @GetMapping("/getReferenceById")
87     public S getReferenceById(@Example<S> example) : S
88
89     @GetMapping("/getReferenceById")
90     public S getReferenceById(@Example<S> example) : S
91
92     @GetMapping("/getReferenceById")
93     public S getReferenceById(@Example<S> example) : S
94
95     @GetMapping("/getReferenceById")
96     public S getReferenceById(@Example<S> example) : S
97
98     @GetMapping("/getReferenceById")
99     public S getReferenceById(@Example<S> example) : S
100
101     @GetMapping("/getReferenceById")
102     public S getReferenceById(@Example<S> example) : S
103
104     @GetMapping("/getReferenceById")
105     public S getReferenceById(@Example<S> example) : S
106
107     @GetMapping("/getReferenceById")
108     public S getReferenceById(@Example<S> example) : S
109
110     @GetMapping("/getReferenceById")
111     public S getReferenceById(@Example<S> example) : S
112
113     @GetMapping("/getReferenceById")
114     public S getReferenceById(@Example<S> example) : S
115
116     @GetMapping("/getReferenceById")
117     public S getReferenceById(@Example<S> example) : S
118
119     @GetMapping("/getReferenceById")
120     public S getReferenceById(@Example<S> example) : S
121
122     @GetMapping("/getReferenceById")
123     public S getReferenceById(@Example<S> example) : S
124
125     @GetMapping("/getReferenceById")
126     public S getReferenceById(@Example<S> example) : S
127
128     @GetMapping("/getReferenceById")
129     public S getReferenceById(@Example<S> example) : S
130
131     @GetMapping("/getReferenceById")
132     public S getReferenceById(@Example<S> example) : S
133
134     @GetMapping("/getReferenceById")
135     public S getReferenceById(@Example<S> example) : S
136
137     @GetMapping("/getReferenceById")
138     public S getReferenceById(@Example<S> example) : S
139
140     @GetMapping("/getReferenceById")
141     public S getReferenceById(@Example<S> example) : S
142
143     @GetMapping("/getReferenceById")
144     public S getReferenceById(@Example<S> example) : S
145
146     @GetMapping("/getReferenceById")
147     public S getReferenceById(@Example<S> example) : S
148
149     @GetMapping("/getReferenceById")
150     public S getReferenceById(@Example<S> example) : S
151
152     @GetMapping("/getReferenceById")
153     public S getReferenceById(@Example<S> example) : S
154
155     @GetMapping("/getReferenceById")
156     public S getReferenceById(@Example<S> example) : S
157
158     @GetMapping("/getReferenceById")
159     public S getReferenceById(@Example<S> example) : S
160
161     @GetMapping("/getReferenceById")
162     public S getReferenceById(@Example<S> example) : S
163
164     @GetMapping("/getReferenceById")
165     public S getReferenceById(@Example<S> example) : S
166
167     @GetMapping("/getReferenceById")
168     public S getReferenceById(@Example<S> example) : S
169
170     @GetMapping("/getReferenceById")
171     public S getReferenceById(@Example<S> example) : S
172
173     @GetMapping("/getReferenceById")
174     public S getReferenceById(@Example<S> example) : S
175
176     @GetMapping("/getReferenceById")
177     public S getReferenceById(@Example<S> example) : S
178
179     @GetMapping("/getReferenceById")
180     public S getReferenceById(@Example<S> example) : S
181
182     @GetMapping("/getReferenceById")
183     public S getReferenceById(@Example<S> example) : S
184
185     @GetMapping("/getReferenceById")
186     public S getReferenceById(@Example<S> example) : S
187
188     @GetMapping("/getReferenceById")
189     public S getReferenceById(@Example<S> example) : S
190
191     @GetMapping("/getReferenceById")
192     public S getReferenceById(@Example<S> example) : S
193
194     @GetMapping("/getReferenceById")
195     public S getReferenceById(@Example<S> example) : S
196
197     @GetMapping("/getReferenceById")
198     public S getReferenceById(@Example<S> example) : S
199
200     @GetMapping("/getReferenceById")
201     public S getReferenceById(@Example<S> example) : S
202
203     @GetMapping("/getReferenceById")
204     public S getReferenceById(@Example<S> example) : S
205
206     @GetMapping("/getReferenceById")
207     public S getReferenceById(@Example<S> example) : S
208
209     @GetMapping("/getReferenceById")
210     public S getReferenceById(@Example<S> example) : S
211
212     @GetMapping("/getReferenceById")
213     public S getReferenceById(@Example<S> example) : S
214
215     @GetMapping("/getReferenceById")
216     public S getReferenceById(@Example<S> example) : S
217
218     @GetMapping("/getReferenceById")
219     public S getReferenceById(@Example<S> example) : S
220
221     @GetMapping("/getReferenceById")
222     public S getReferenceById(@Example<S> example) : S
223
224     @GetMapping("/getReferenceById")
225     public S getReferenceById(@Example<S> example) : S
226
227     @GetMapping("/getReferenceById")
228     public S getReferenceById(@Example<S> example) : S
229
230     @GetMapping("/getReferenceById")
231     public S getReferenceById(@Example<S> example) : S
232
233     @GetMapping("/getReferenceById")
234     public S getReferenceById(@Example<S> example) : S
235
236     @GetMapping("/getReferenceById")
237     public S getReferenceById(@Example<S> example) : S
238
239     @GetMapping("/getReferenceById")
240     public S getReferenceById(@Example<S> example) : S
241
242     @GetMapping("/getReferenceById")
243     public S getReferenceById(@Example<S> example) : S
244
245     @GetMapping("/getReferenceById")
246     public S getReferenceById(@Example<S> example) : S
247
248     @GetMapping("/getReferenceById")
249     public S getReferenceById(@Example<S> example) : S
250
251     @GetMapping("/getReferenceById")
252     public S getReferenceById(@Example<S> example) : S
253
254     @GetMapping("/getReferenceById")
255     public S getReferenceById(@Example<S> example) : S
256
257     @GetMapping("/getReferenceById")
258     public S getReferenceById(@Example<S> example) : S
259
260     @GetMapping("/getReferenceById")
261     public S getReferenceById(@Example<S> example) : S
262
263     @GetMapping("/getReferenceById")
264     public S getReferenceById(@Example<S> example) : S
265
266     @GetMapping("/getReferenceById")
267     public S getReferenceById(@Example<S> example) : S
268
269     @GetMapping("/getReferenceById")
270     public S getReferenceById(@Example<S> example) : S
271
272     @GetMapping("/getReferenceById")
273     public S getReferenceById(@Example<S> example) : S
274
275     @GetMapping("/getReferenceById")
276     public S getReferenceById(@Example<S> example) : S
277
278     @GetMapping("/getReferenceById")
279     public S getReferenceById(@Example<S> example) : S
280
281     @GetMapping("/getReferenceById")
282     public S getReferenceById(@Example<S> example) : S
283
284     @GetMapping("/getReferenceById")
285     public S getReferenceById(@Example<S> example) : S
286
287     @GetMapping("/getReferenceById")
288     public S getReferenceById(@Example<S> example) : S
289
290     @GetMapping("/getReferenceById")
291     public S getReferenceById(@Example<S> example) : S
292
293     @GetMapping("/getReferenceById")
294     public S getReferenceById(@Example<S> example) : S
295
296     @GetMapping("/getReferenceById")
297     public S getReferenceById(@Example<S> example) : S
298
299     @GetMapping("/getReferenceById")
300     public S getReferenceById(@Example<S> example) : S
301
302     @GetMapping("/getReferenceById")
303     public S getReferenceById(@Example<S> example) : S
304
305     @GetMapping("/getReferenceById")
306     public S getReferenceById(@Example<S> example) : S
307
308     @GetMapping("/getReferenceById")
309     public S getReferenceById(@Example<S> example) : S
310
311     @GetMapping("/getReferenceById")
312     public S getReferenceById(@Example<S> example) : S
313
314     @GetMapping("/getReferenceById")
315     public S getReferenceById(@Example<S> example) : S
316
317     @GetMapping("/getReferenceById")
318     public S getReferenceById(@Example<S> example) : S
319
320     @GetMapping("/getReferenceById")
321     public S getReferenceById(@Example<S> example) : S
322
323     @GetMapping("/getReferenceById")
324     public S getReferenceById(@Example<S> example) : S
325
326     @GetMapping("/getReferenceById")
327     public S getReferenceById(@Example<S> example) : S
328
329     @GetMapping("/getReferenceById")
330     public S getReferenceById(@Example<S> example) : S
331
332     @GetMapping("/getReferenceById")
333     public S getReferenceById(@Example<S> example) : S
334
335     @GetMapping("/getReferenceById")
336     public S getReferenceById(@Example<S> example) : S
337
338     @GetMapping("/getReferenceById")
339     public S getReferenceById(@Example<S> example) : S
340
341     @GetMapping("/getReferenceById")
342     public S getReferenceById(@Example<S> example) : S
343
344     @GetMapping("/getReferenceById")
345     public S getReferenceById(@Example<S> example) : S
346
347     @GetMapping("/getReferenceById")
348     public S getReferenceById(@Example<S> example) : S
349
350     @GetMapping("/getReferenceById")
351     public S getReferenceById(@Example<S> example) : S
352
353     @GetMapping("/getReferenceById")
354     public S getReferenceById(@Example<S> example) : S
355
356     @GetMapping("/getReferenceById")
357     public S getReferenceById(@Example<S> example) : S
358
359     @GetMapping("/getReferenceById")
360     public S getReferenceById(@Example<S> example) : S
361
362     @GetMapping("/getReferenceById")
363     public S getReferenceById(@Example<S> example) : S
364
365     @GetMapping("/getReferenceById")
366     public S getReferenceById(@Example<S> example) : S
367
368     @GetMapping("/getReferenceById")
369     public S getReferenceById(@Example<S> example) : S
370
371     @GetMapping("/getReferenceById")
372     public S getReferenceById(@Example<S> example) : S
373
374     @GetMapping("/getReferenceById")
375     public S getReferenceById(@Example<S> example) : S
376
377     @GetMapping("/getReferenceById")
378     public S getReferenceById(@Example<S> example) : S
379
380     @GetMapping("/getReferenceById")
381     public S getReferenceById(@Example<S> example) : S
382
383     @GetMapping("/getReferenceById")
384     public S getReferenceById(@Example<S> example) : S
385
386     @GetMapping("/getReferenceById")
387     public S getReferenceById(@Example<S> example) : S
388
389     @GetMapping("/getReferenceById")
390     public S getReferenceById(@Example<S> example) : S
391
392     @GetMapping("/getReferenceById")
393     public S getReferenceById(@Example<S> example) : S
394
395     @GetMapping("/getReferenceById")
396     public S getReferenceById(@Example<S> example) : S
397
398     @GetMapping("/getReferenceById")
399     public S getReferenceById(@Example<S> example) : S
400
401     @GetMapping("/getReferenceById")
402     public S getReferenceById(@Example<S> example) : S
403
404     @GetMapping("/getReferenceById")
405     public S getReferenceById(@Example<S> example) : S
406
407     @GetMapping("/getReferenceById")
408     public S getReferenceById(@Example<S> example) : S
409
410     @GetMapping("/getReferenceById")
411     public S getReferenceById(@Example<S> example) : S
412
413     @GetMapping("/getReferenceById")
414     public S getReferenceById(@Example<S> example) : S
415
416     @GetMapping("/getReferenceById")
417     public S getReferenceById(@Example<S> example) : S
418
419     @GetMapping("/getReferenceById")
420     public S getReferenceById(@Example<S> example) : S
421
422     @GetMapping("/getReferenceById")
423     public S getReferenceById(@Example<S> example) : S
424
425     @GetMapping("/getReferenceById")
426     public S getReferenceById(@Example<S> example) : S
427
428     @GetMapping("/getReferenceById")
429     public S getReferenceById(@Example<S> example) : S
429
```

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:

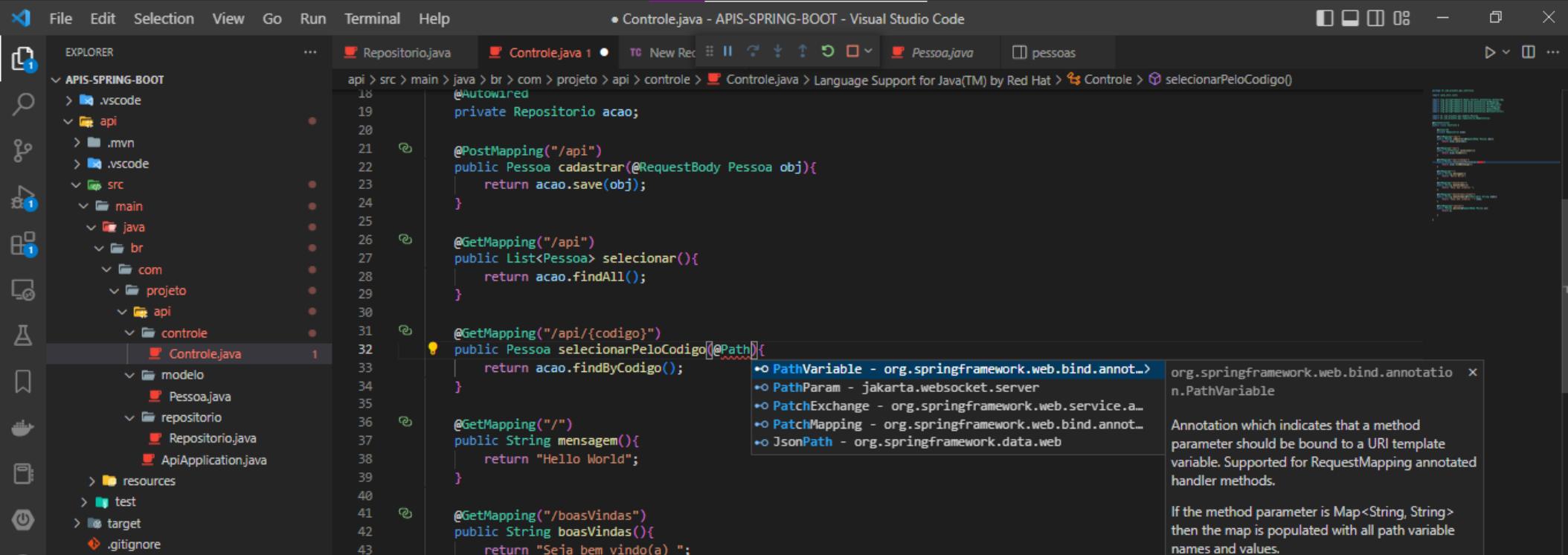
```
12     import br.com.projeto.api.modelo.Pessoa;
13     import br.com.projeto.api.repositorio.Repositorio;
14
15     @RestController
16     public class Controle {
17
18         @Autowired
19         private Repositorio acao;
20
21         @PostMapping("/api")
22         public Pessoa cadastrar(@RequestBody 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 ● TC 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 }
```



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
 - modelo
 - Pessoa.java
 - repository

REPOSITORY

Controle.java 1

New Rec

Language Support for Java(TM) by Red Hat

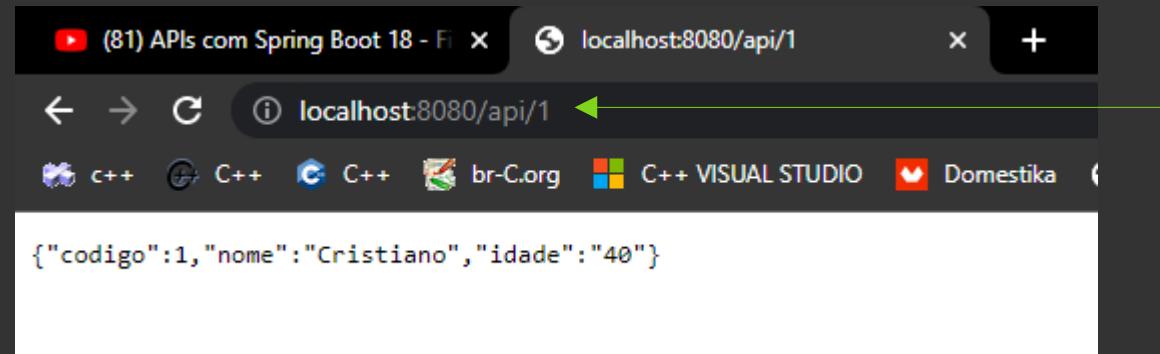
```
18 @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
31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

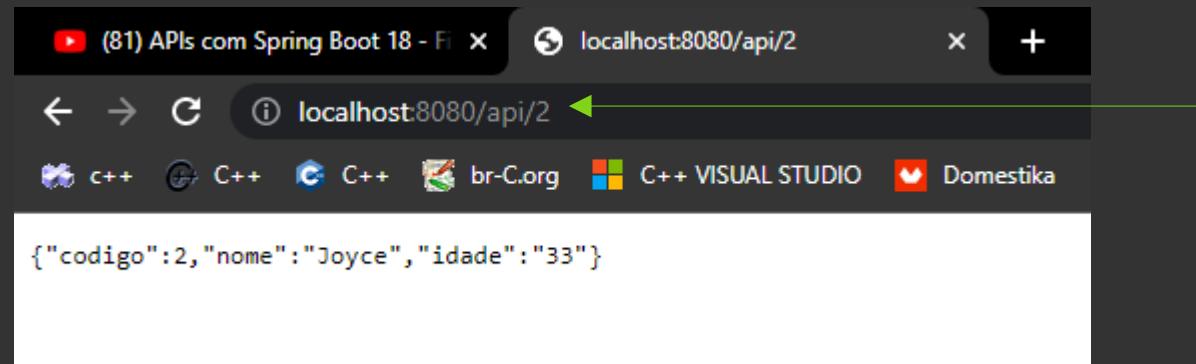
```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30
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". 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". 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.
- Status Bar:** Shows tabs for Repository.java, Controle.java (active), New Record, and Pessoa.java, along with other icons.

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:

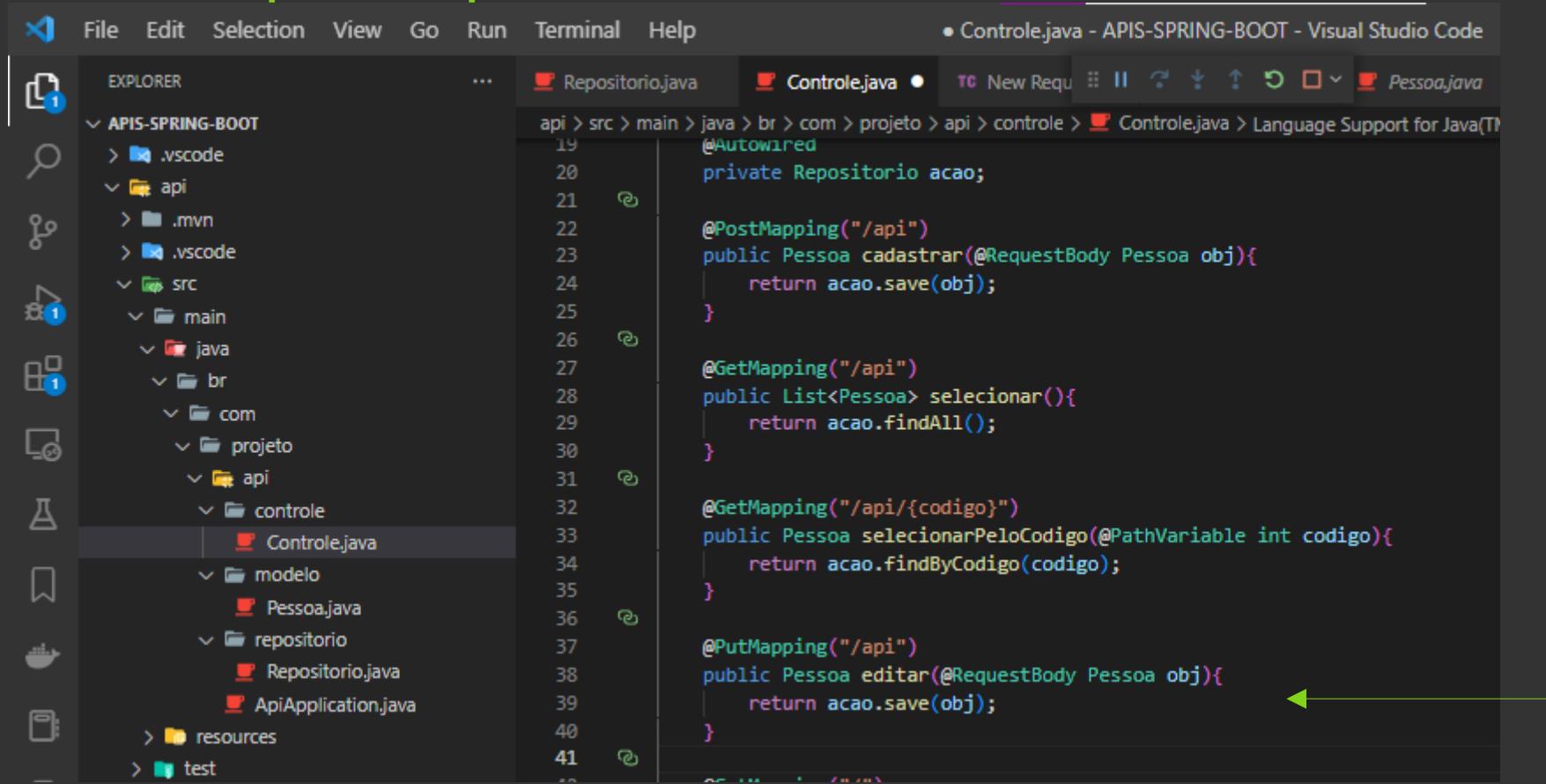
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Terminal:** Shows the command "Controle.java - APIS-SPRING-BOOT - Visual Studio Code".
- Code Editor:** Displays Java code for "Controle.java".

```
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



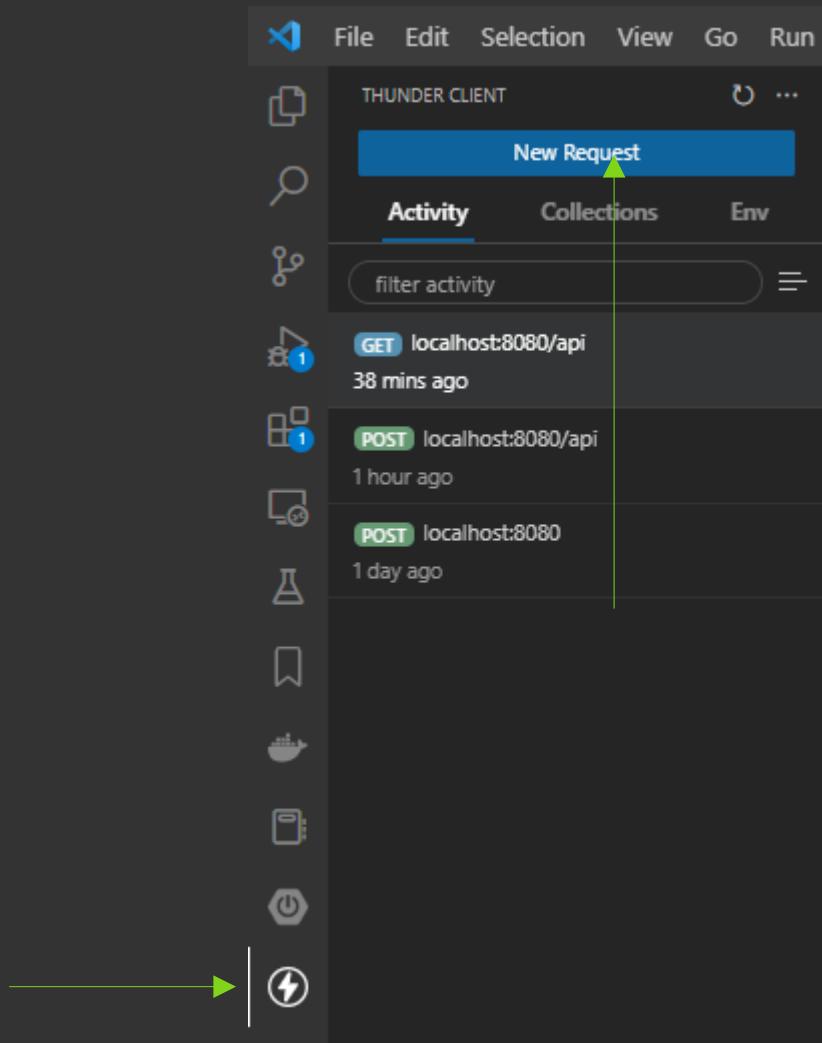
The screenshot shows the Visual Studio Code interface with the following details:

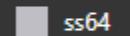
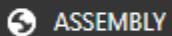
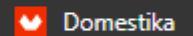
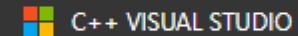
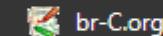
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle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

```
[{"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

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

Json Content

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

The screenshot shows the Thunder Client extension in Visual Studio Code. A green arrow points from the JSON response in the browser at the top to the JSON body in the Thunder Client interface below. Another green arrow points from the 'Send' button in the Thunder Client interface to the JSON content itself.

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 displays 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
 - resources
 - test
 - target
 - .gitignore
 - HELP.md

REPOSITORY

Repository.java

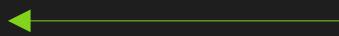
Controle.java

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 a tooltip for the `@Delete` annotation. The code snippet is as follows:

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

@Controller
public class Controle {
    @Delete
    public void delete() {
    }
}
```

The tooltip for `@Delete` provides the following information:

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

- Since:
 - 6.0
- Author:
 - Rossen Stoyanchev

Annotations shown in the tooltip include: DeleteExchange, DeleteMapping, DeleteOperation, Delimiter, DefaultValue, DeclareAnnotation, DeclareParents, and DependsOnDatabaseInitialization.

IDE status bar: LEMS 1 OUTPUT 1

Java error message: Delet 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
 - repositorio
 - Repositorio.java
 - ApiApplication.java
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - Code Editor:** The file Controle.java is open, showing the following Java 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(){}
```

Adicionamos a annotation 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:** Shows the current file path: api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java.
 - Code Editor:** Displays the Controle.java code with syntax highlighting. The code includes methods for creating, reading, updating, and deleting a Pessoa object based on their ID.

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

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". Inside "controle", the file "Controle.java" is selected. Other files like "Pessoa.java", "Repositorio.java", and "ApiApplication.java" are also listed under "controle".
- Code Editor:** Displays the Java code for the "Controle.java" file. 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

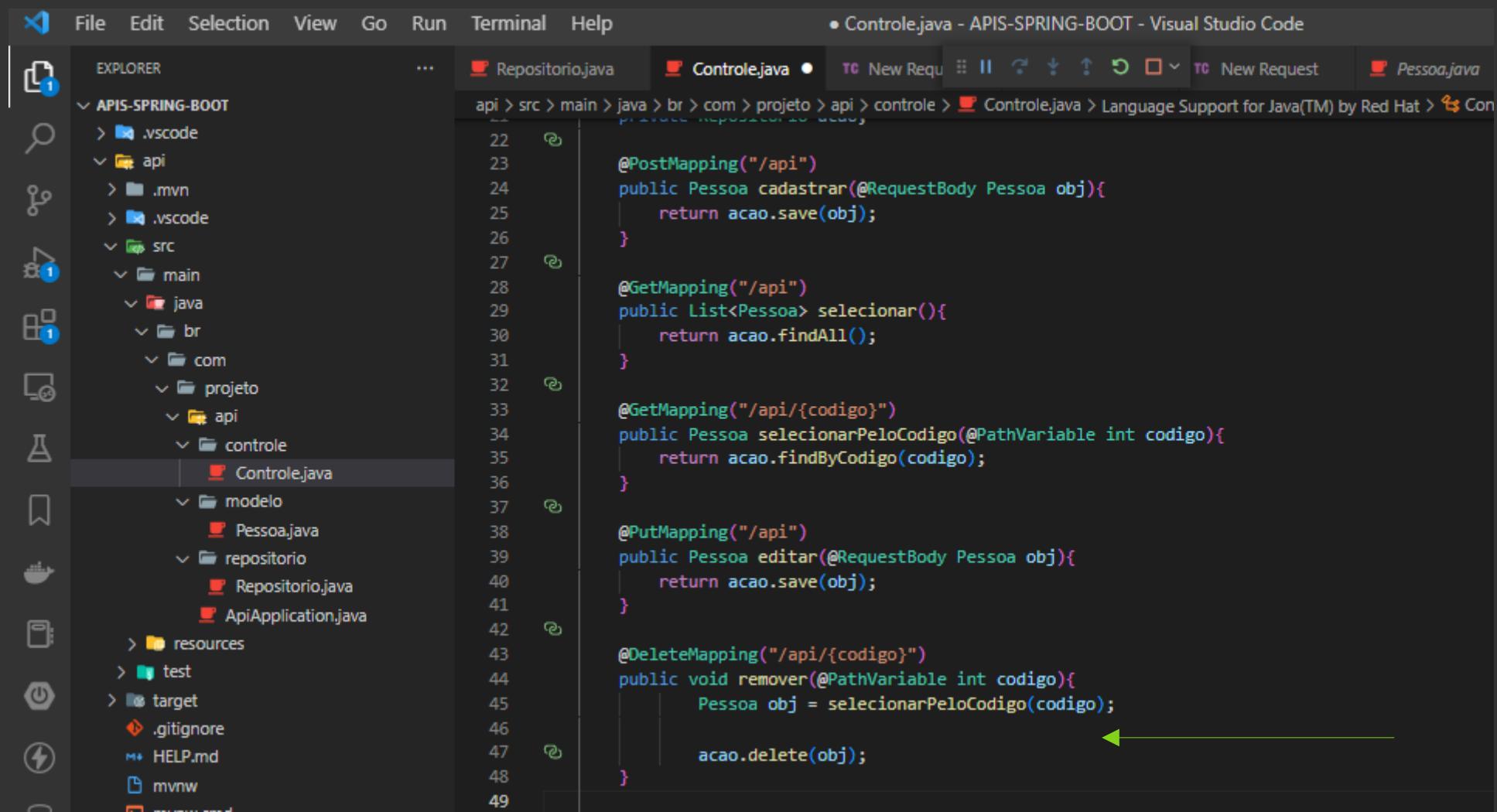
Deletes a given entity.

- **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

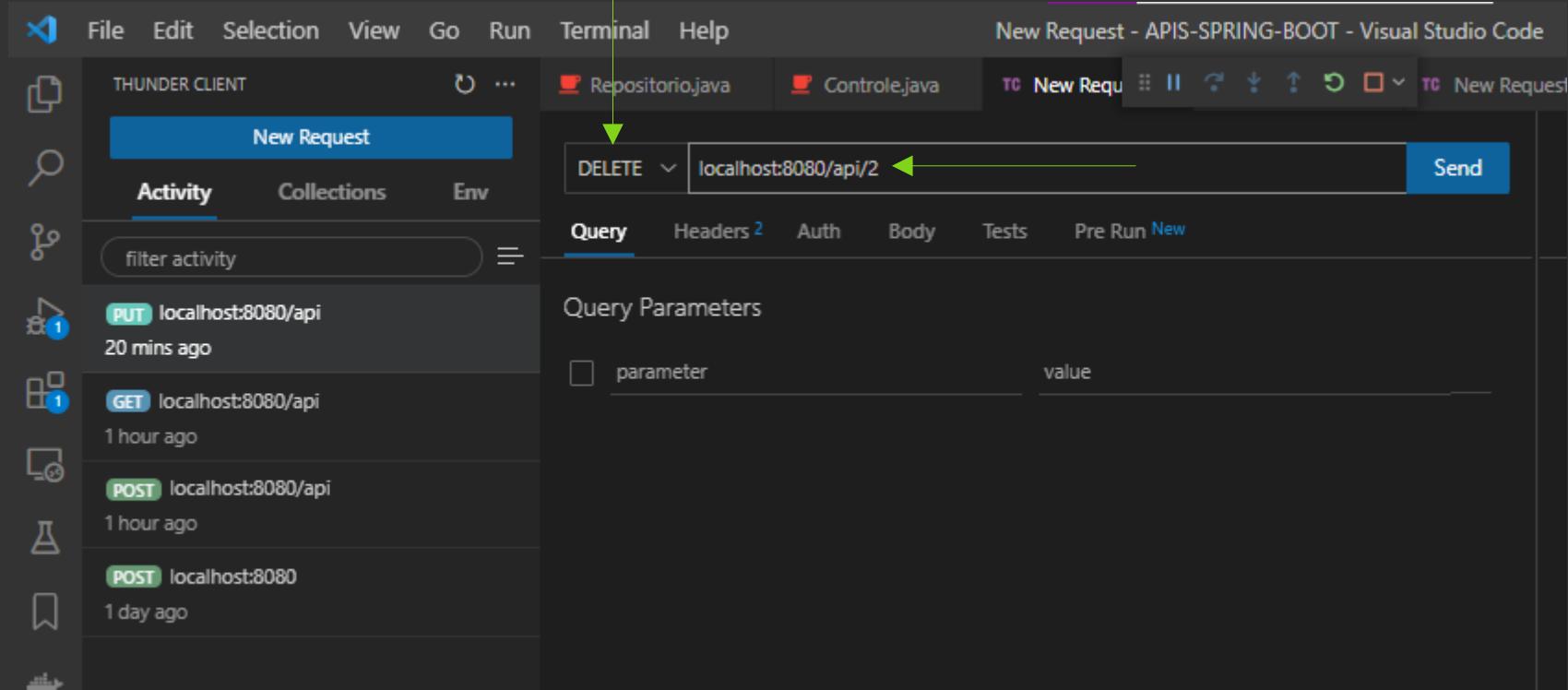
Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

Filter (e.g. text, **/*.ts, **/node_modules/**)



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

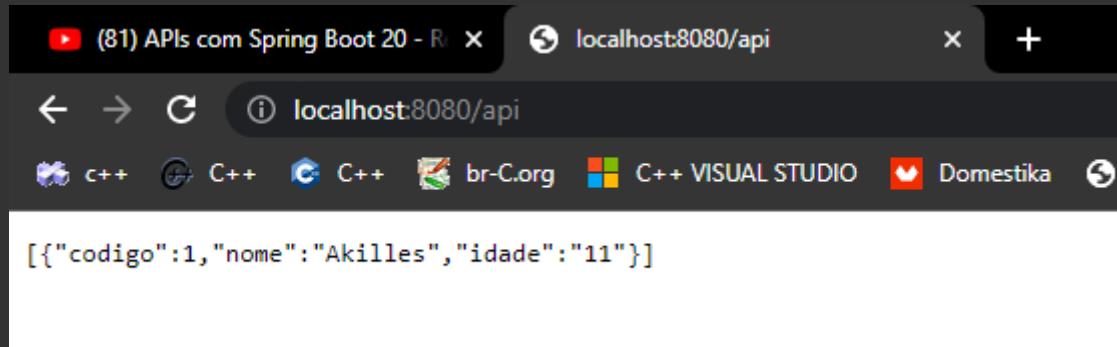
parameter value

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

The screenshot shows the Thunder Client extension in Visual Studio Code. A POST request is being sent to `http://localhost:8080/api`. The request body is a JSON object:

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

The response status is **200 OK**, size is **44 Bytes**, and time is **120 ms**. The response body is identical to the request body.

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
```

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 }
```

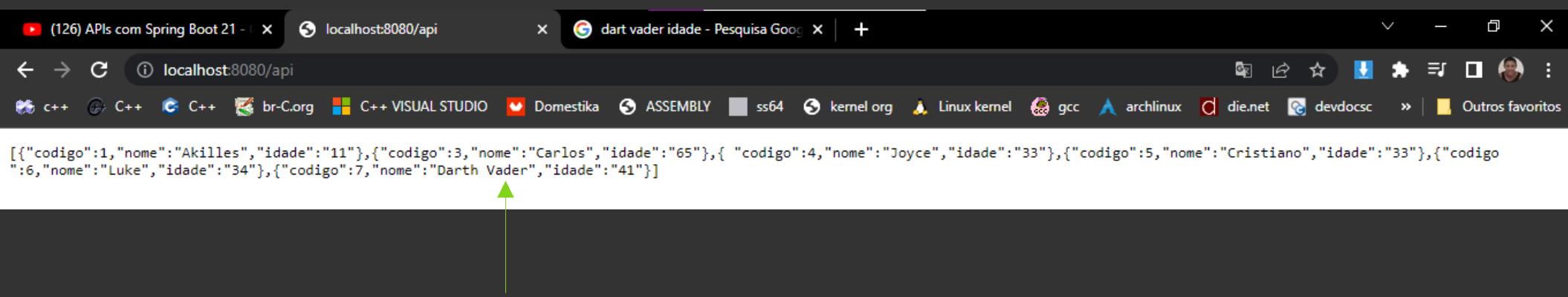
localhost:8080/pessoa just now

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

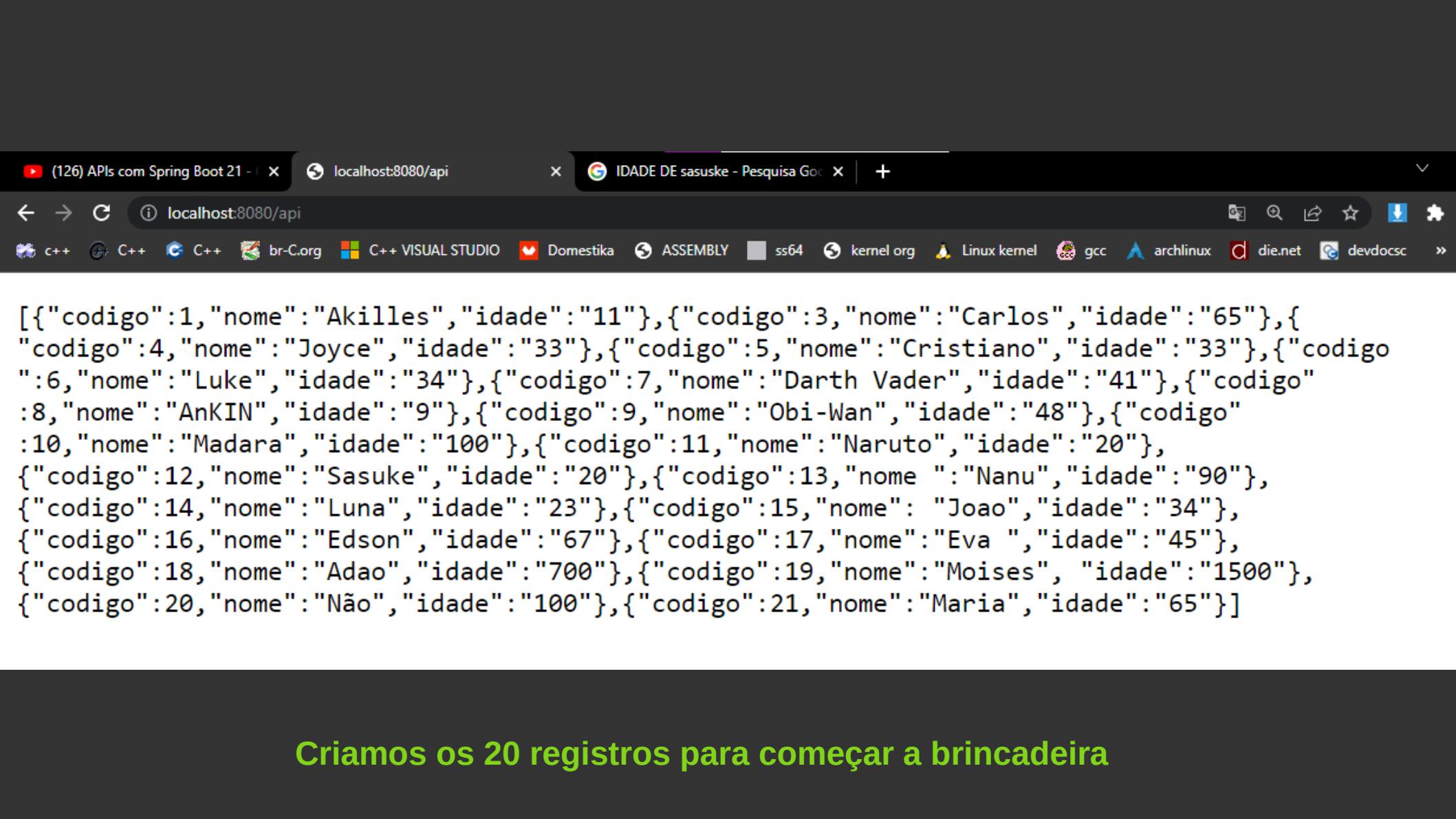


A screenshot of a browser window with a dark theme. The address bar shows 'localhost:8080/api'. The main content area displays a JSON array of objects:

```
[{"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

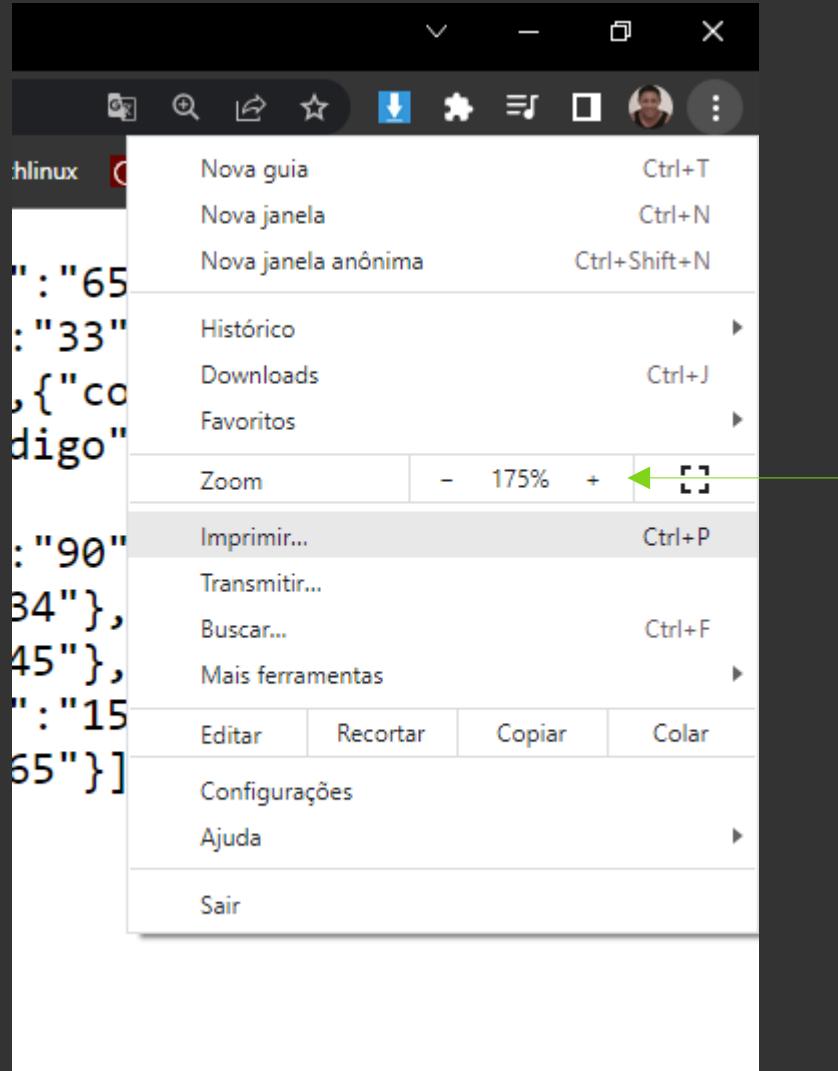


The screenshot shows a browser window with three tabs open. The active tab is titled "localhost:8080/api". Below the tabs, the address bar also displays "localhost:8080/api". The page content is a JSON array containing 20 objects, each representing a person with fields "codigo", "nome", and "idade".

```
[{"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):**
 - Connection: 127.0.0.1@3306 8.0.25
 - Schemas: api_spring (16k)
 - Tables (1):
 - pessoas
 - columns:
 - codigo int
 - idade varchar(255)
 - nome varchar(255)
 - Views, Procedures, Functions, dbagenda, sakila, world.
- Right Panel (Code Editor):** Controle.java 1 (highlighted). 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 }
```

Começamos criando um metodo publico que retorna um Long

Porque Long?

Porque é o tipo de dado que o metodo count vai retornar

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
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 }
51
52
53 @GetMapping("/")
54 public String mensagem(){
55     return "Hello World";
56 }
57
```

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

Repository.java

Controle.java

localhost:

New Request

```
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 the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Database Explorer:** Shows a connection to 127.0.0.1@3306 (8.0.25) and a schema named api_spring (16k). Inside the schema, there is a table named pessoas with columns: codigo (int), idade (varchar(255)), and nome (varchar(255)). Other tables listed are index, views, procedures, functions, dbagenda, sakila, and world.
- Code Editor:** The current file is Controle.java, which contains the following Java code:

```
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 @GetMapping("/")
55 public String mensagem(){
56     return "Hello World";
57 }
```

**Salve va no navegador
E coloque a rota *api/contador*
Ela deve mostrar quantos registros temos**

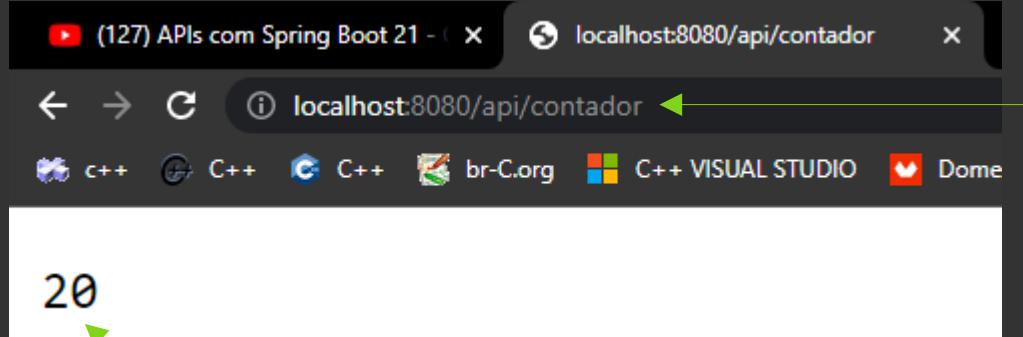
The screenshot shows a MySQL Workbench interface with the following details:

- 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 |

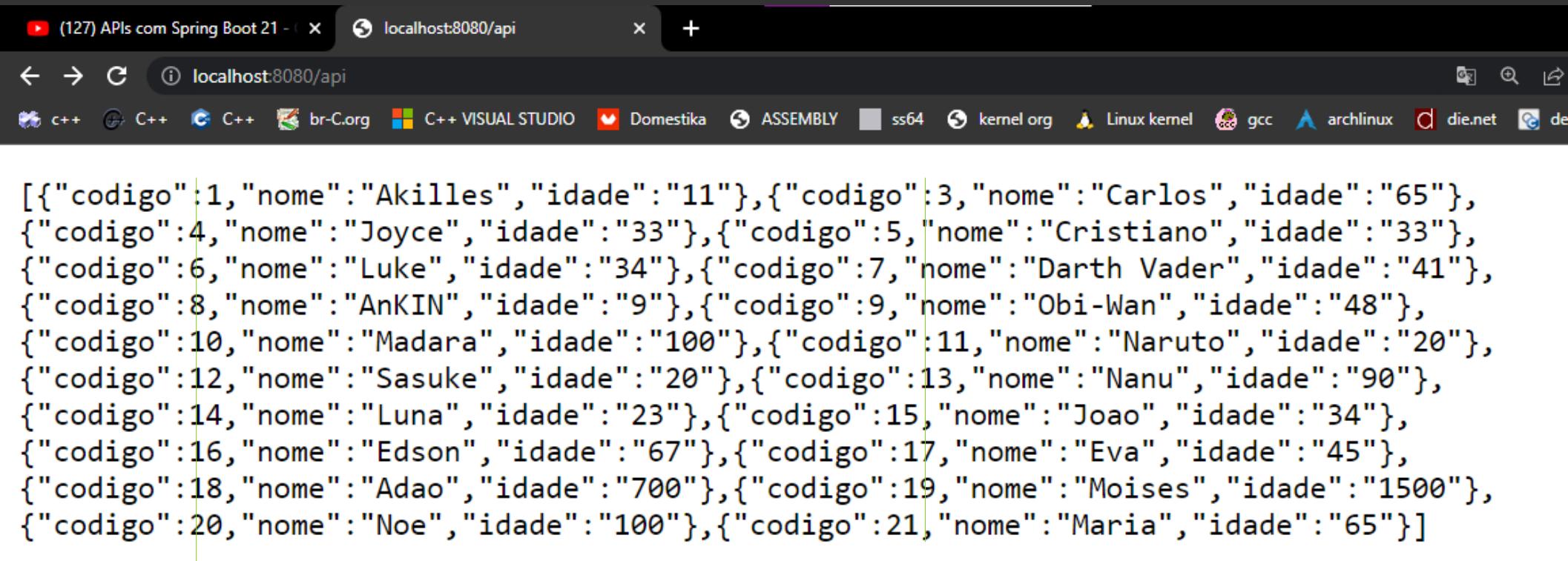
Nosso banco de dados mostra 21 registros



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 "api_spring" (16k), "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)".
- Code Completion Pop-up:** A tooltip for the word "List" is displayed, showing its definition as "java.util.List". It provides a detailed description of the List interface, noting it's an ordered collection, allows duplicates, and provides methods like add, remove, equals, and hashCode.
- Completion Suggestions:** Below the tooltip, a list of suggestions for "List" is shown, including "List - java.util" (selected), "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.BasicLis...", "ListDataListener - javax.swing.event", and "ListELResolver - jakarta.el".
- Bottom Status Bar:** Shows "PROBLEMS 3", "OUTPUT", and "TERMINAL". The TERMINAL tab is active, displaying the output of a Spring application startup.
- Right Side:** A "DEBUG CONSOLE" tab is visible.

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". The "tables" section contains one entry: "pessoas". Other sections like "views", "procedures", and "functions" are also listed.
- Terminal:** Shows a local host connection: localhost:3306
- Code Editor:** The file "Repositorio.java" is open. The code defines a public interface named "Repositorio" that extends "JpaRepository<Pessoa, Integer>". It includes three methods: "findAll()", "findByCodigo(int codigo)", and "findByOrderByName()". The "findByOrderByName()" method has a yellow lightbulb icon, indicating a potential issue or suggestion.

```
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> findByOrderByName(); ←
21
22 }
```

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 named "api".
- Editor:** The active file is "Controle.java 1" located at "api > src > main > java > br > com > projeto > api > controle".
- Code:** Java code for a controller class:

```
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<Pessoas> findAll() {
55 }
```

- Completion Panel:** A tooltip for the word "List" is displayed, showing its definition from the Java.util.List interface.
- Definition:** The tooltip for "java.util.List" provides the following text:

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_1 and e_2 such that $e_1.equals(e_2)$, 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.

**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 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".
 - 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
 - resources, test, target, .gitignore.
 - Terminal View:** localhost:3001
 - Code Editor:** Controle.java file open, showing Java code for a Spring Boot API controller.

```
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 }
```

A red arrow points from the text "Salve va no navegador e coloque a rota ela deve ordenar" at the bottom to the line of code "@GetMapping(\"/api/ordenarNomes\")".

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:

- Left Sidebar (Database Explorer):** Shows a connection to "127.0.0.1@3306 8.0.25" with the schema "api_spring". Under "tables", there is one entry: "pessoas" (1 row). It lists columns: "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other tables listed are "index", "views", "procedures", "functions", "dbagenda", "sakila", and "world".
- Top Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Code Editor:** The file "Controle.java" is open, showing the following 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

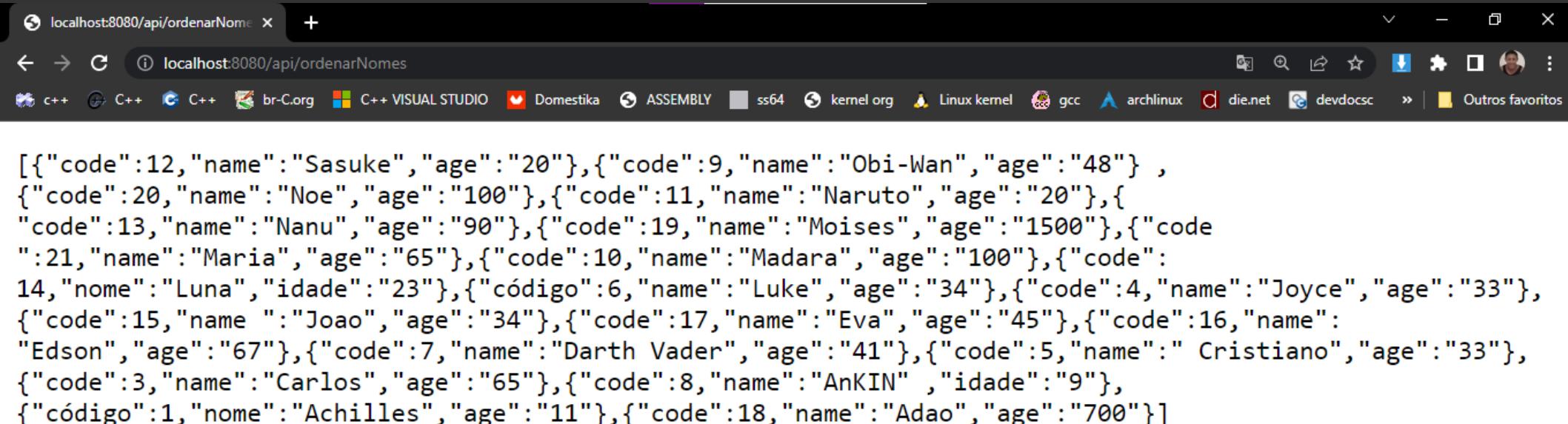
DATABASE 127.0.0.1@3306 8.0.25
api_spring 16k
tables (1)
pessoas
columns
codigo int
idade varchar(255)
nome varchar(255)
index
views
procedures
functions
dbagenda
sakila
world

```
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 API. The code includes annotations for PUT, DELETE, and GET methods, and calls to a DAO named "acao". To the left of the code editor is a "DATABASE" sidebar showing a connection to "127.0.0.1@3306 8.0.25" and a schema named "api_spring". Inside the schema, there is a table named "pessoas" with columns "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). There are also entries for "index", "views", "procedures", and "functions". Other tables listed in the schema include "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 titled "localhost:8080/api/ordenarNomes". The browser interface includes a back/forward button, search bar, and various bookmark icons. The main content area displays a JSON array of objects:

```
[{"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,"name":"Luna","idade":"23"}, {"code":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"}, {"code":1,"name":"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.

```
Repositorio.java 3 • Controle.java localhos II ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌍ ⌎ New Request
api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java > Language Support for Java(TM) by Red Hat > • Repositorio
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> findByOrderByNomeDesc();
21
22        List<Pessoa> ←
23
24    }
25
```

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:

```
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     List<Pessoa> findByNome
23 }
24
25 }
```

A tooltip is visible at the bottom right of the code editor, showing two method suggestions:

- [e] findByNList : List<br.com.projeto.api.modelo.Pess...
- [e] findByNPessoas : List<br.com.projeto.api.modelo.P...

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".
 - folders: .vscode, api, .mvn, .vscode, src.
 - src folder contains:
 - main folder
 - java folder
 - br folder
 - com folder
 - projeto folder
 - api folder
 - controle folder (containing Controle.java)
 - modelo folder (containing Pessoa.java)
 - repositorio folder (containing Repository.java)
 - files: ApiApplication.java, resources, test, target, .gitignore.
- Code Editor:** Displays the content of **Repository.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();
    Pessoa findByCodigo(int codigo);
    List<Pessoa> findByOrderByNameDesc();
    List<Pessoa> findByNomeOrderByIdadeDesc();
}
```
- Terminal:** localhost:3001 (with icons for copy, paste, refresh, etc.)
- Activity Bar:** Includes icons for file operations like Open, Save, Find, Replace, and others.

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



```
1 package br.com.projeto.api.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
6
7 import br.com.projeto.api.modelo.Pessoa;
8 import jakarta.transaction.Transactional;
9
10 //Repository
11 @Transactional
12
13 public interface Repositorio extends JpaRepository<Pessoa, Integer>{
14
15     List<Pessoa> findAll();
16
17     Pessoa findByCodigo(int codigo);
18
19     List<Pessoa> findByOrderByNameDesc();
20
21     List<Pessoa> findByNameOrderByidadeDesc(String nome); ←
22
23
24 }
```

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.

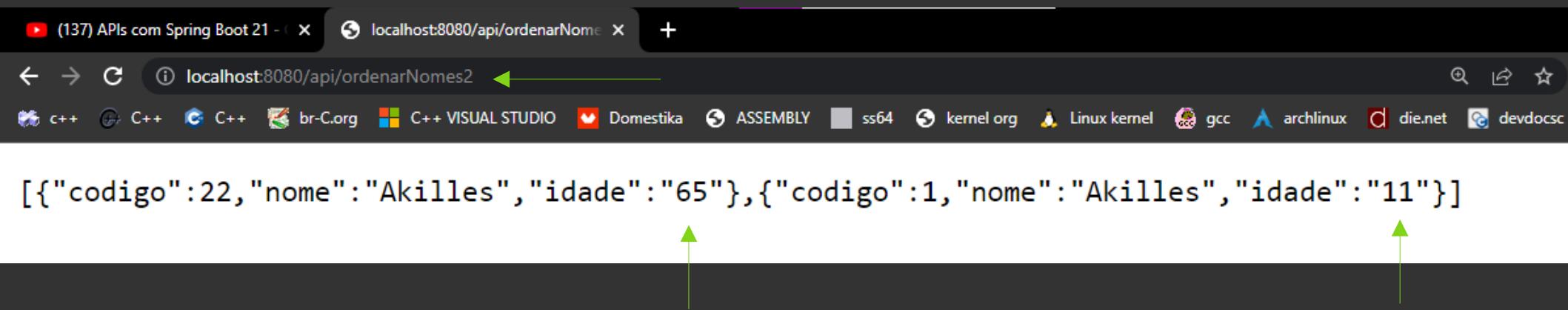
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- 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 in the 'src/main/java/com/projeto/api/controle' folder.
- Terminal View:** Shows the URL 'localhost:3001' and a 'New Request' button.
- Code Editor:** Displays the Java code for 'Controle.java'. The code includes methods for editing, removing, and counting entities, as well as ordering them by name or age.

```
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
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()

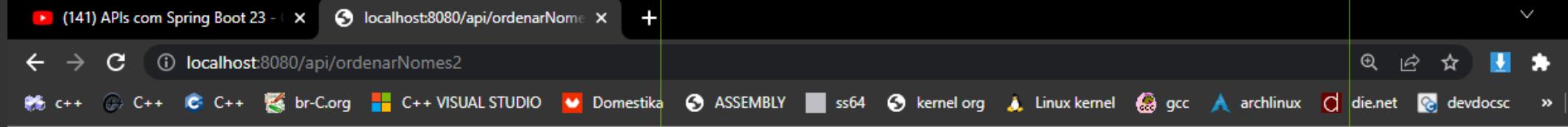
```
38     @PostMapping("/api")
39     public Pessoa editar(@RequestBody Pessoa obj){
40         return acao.save(obj);
41     }
42
43     @DeleteMapping("/api/{codigo}")
44     public void remover(@PathVariable int codigo){
45         Pessoa obj = selecionarPeloCodigo(codigo);
46         acao.delete(obj);
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:** 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
 - resources
 - test
 - target
 - .aitianore
 - Code Editor:** Displays the Repository.java file content.

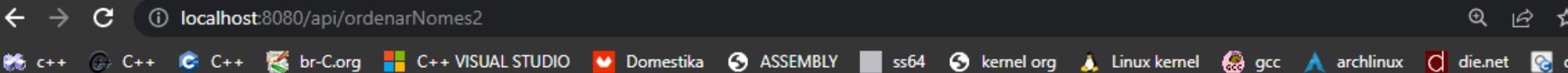
```
package br.com.projeto.api.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
//import org.springframework.data.repository.CrudRepository;
//import org.springframework.stereotype.Repository;
import br.com.projeto.api.modelo.Pessoa;
import jakarta.transaction.Transactional;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



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
 - Repositorio.java
 - modelo
 - Pessoa.java
 - repositorio
 - Repositorio.java
 - 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

> .vscode

> API-SPRING-BOOT

> .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

mvnw

mvnw.cmd

pom.xml

REDME.md

Repositorio.java

Controle.java 2



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

Returns all instances of the type.

- Returns:
 - all entities

PROBLEMS

OUTPUT

TERMINAL

TERMINAL

DEBUG

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:
 - 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 }
```

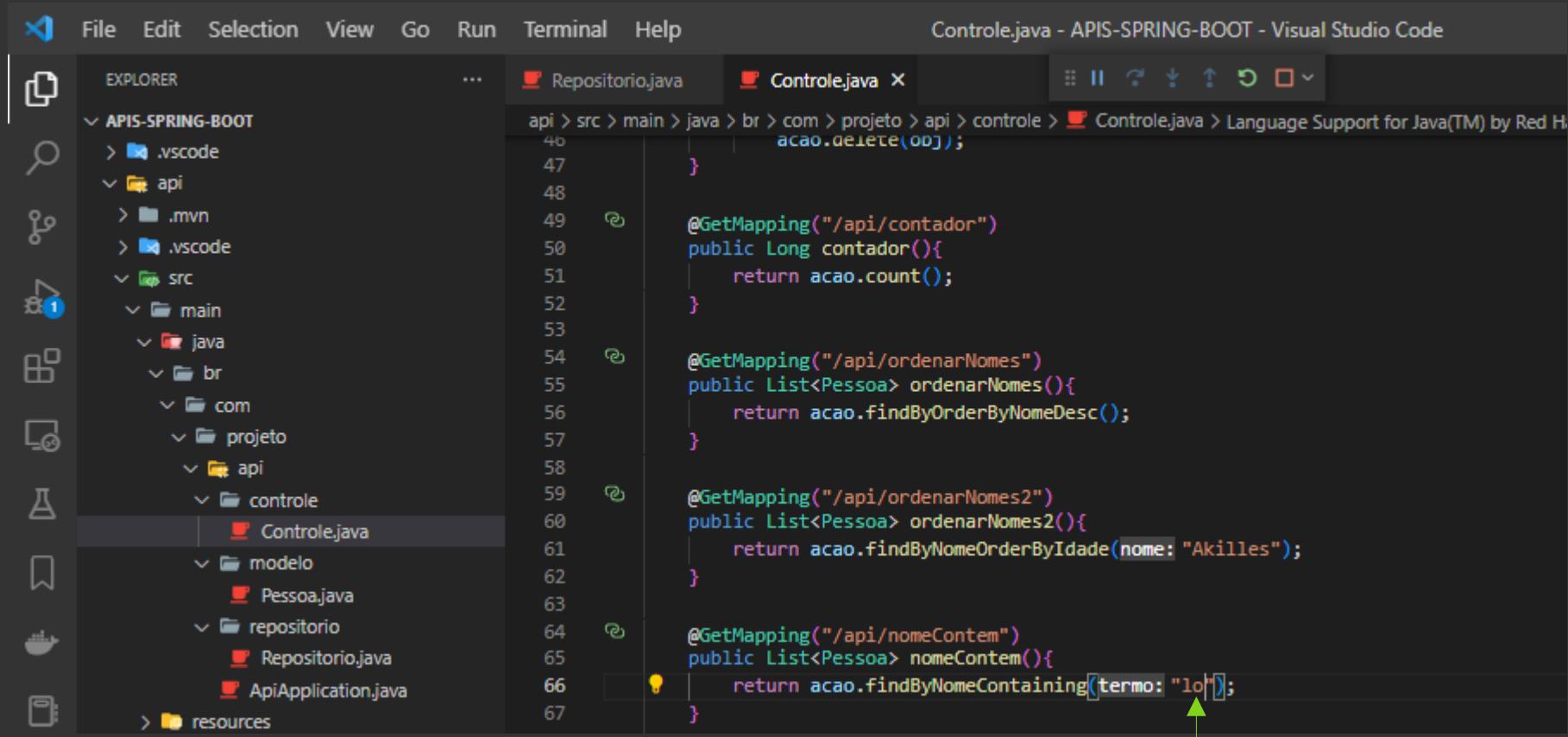
Em tabelas não importa a letra ser maiuscula ou minuscula nesse caso colocamos I mais poderiamos usar L tanto faz
Salve e coloque a rota no navegador

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:

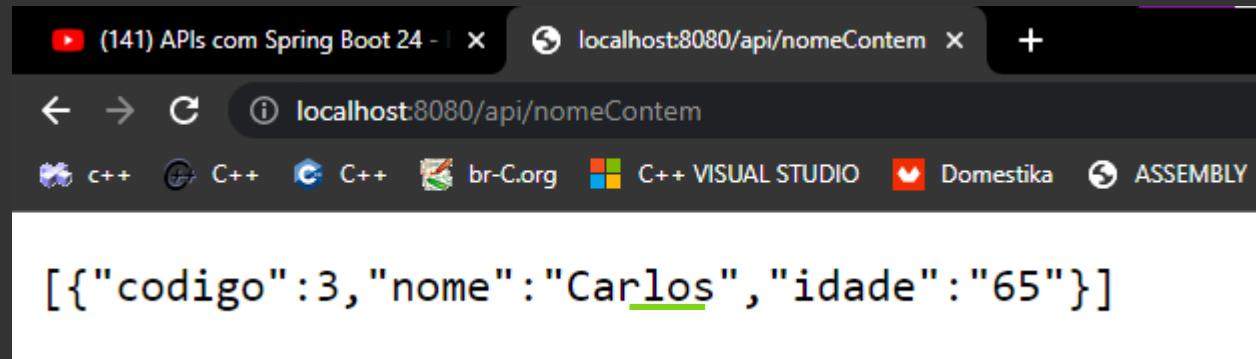
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
 - src folder contains .mvn, .vscode, and java.
 - java folder contains br, com, projeto, and 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

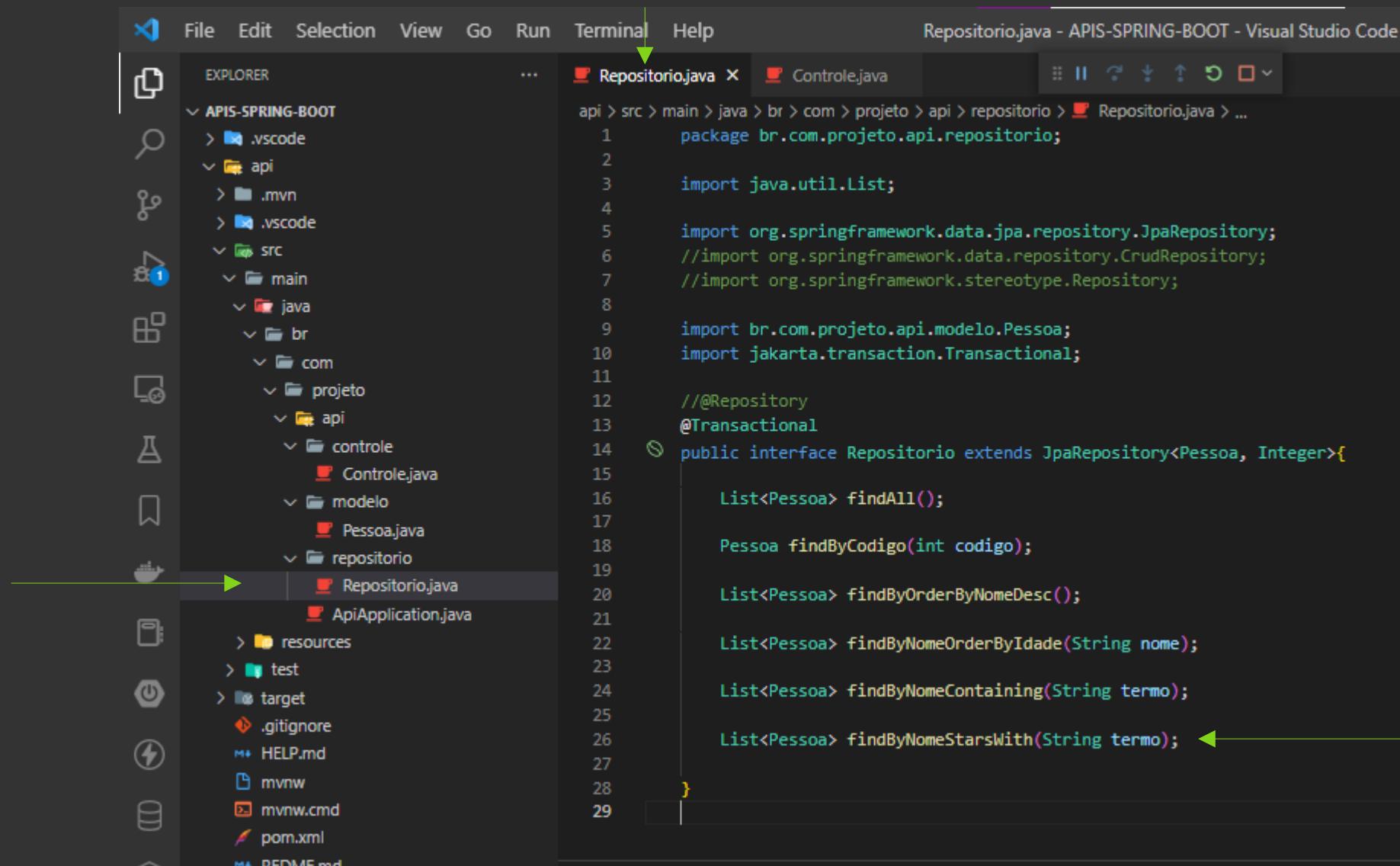


A screenshot of a dark-themed browser window. The address bar shows the URL `localhost:8080/api/nomeContem`. The main content area displays a JSON array:

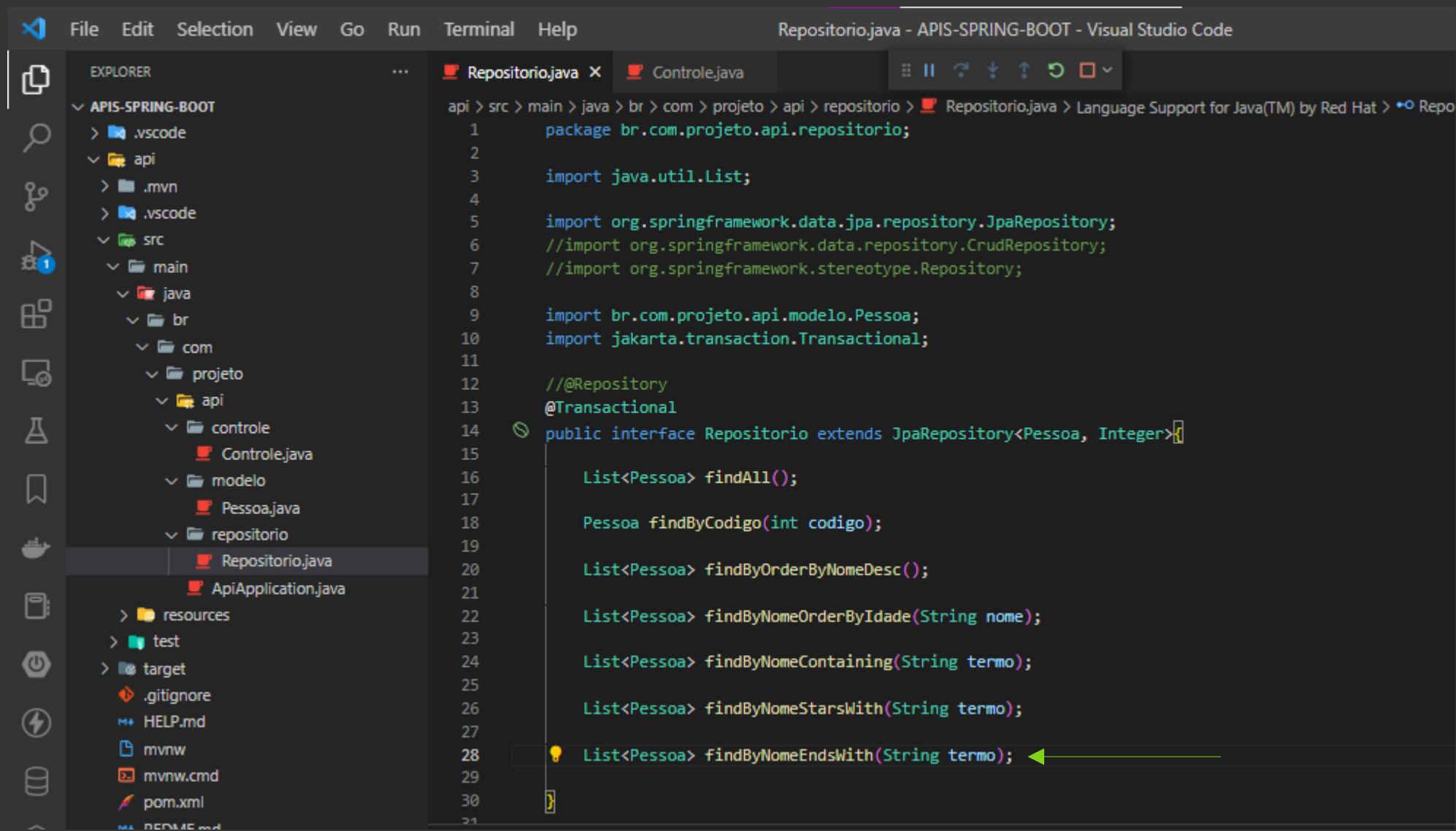
```
[{"codigo":3,"nome":"Carlos","idade":"65"}]
```

#25
StarsWith e EndsWith
São funções

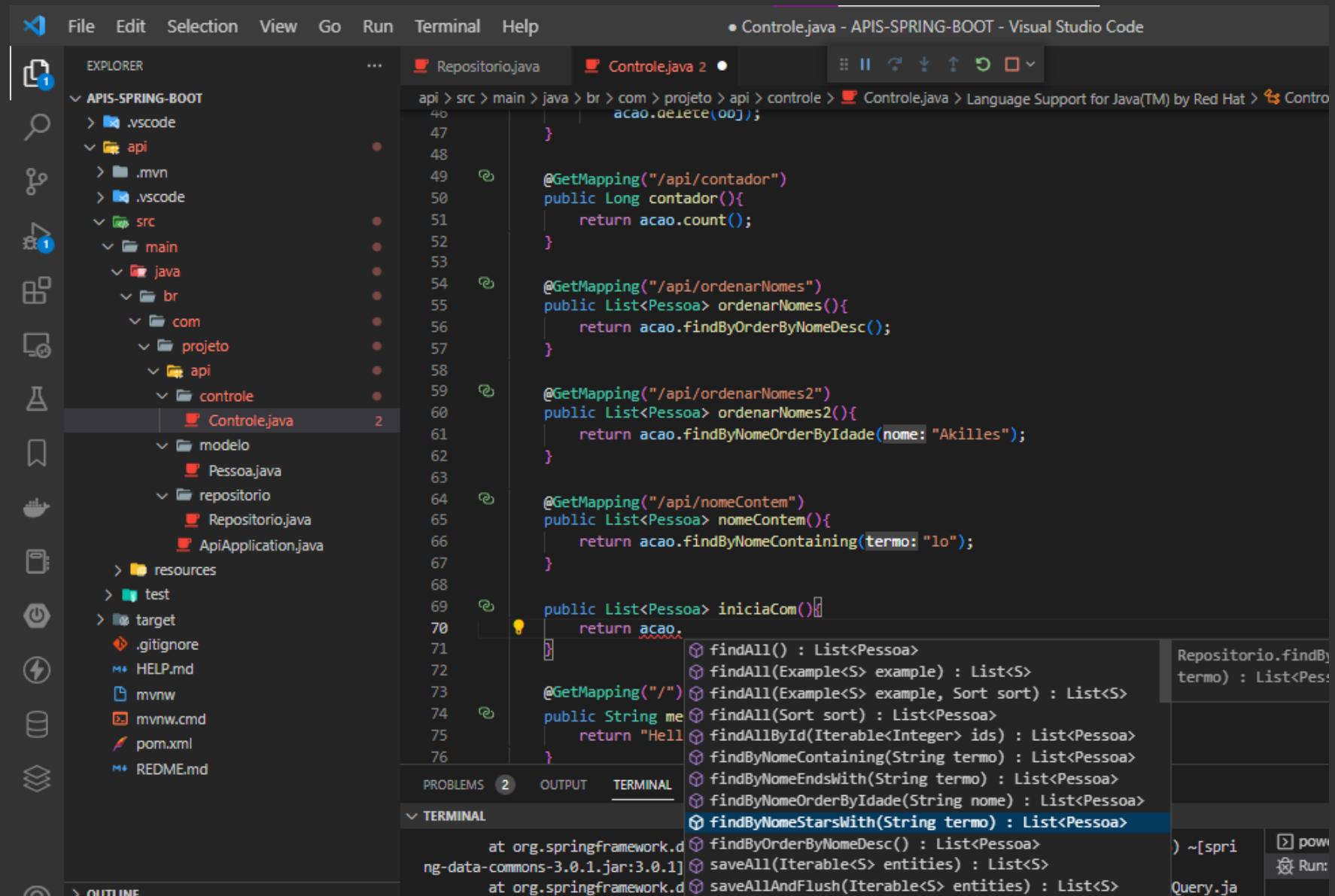
Com essas funções podemos filtrar por coisas que começam ou finalizam com alguma coisa que queremos



Ou seja se eu digitar o termo “a” todos os nomes que iniciam com a letra a serão listados



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

The screenshot shows a browser window with two tabs open. The left tab, titled 'localhost:8080/api', displays a JSON array of names:

```
[{"codigo":1,"nome":"Akilles"}, {"codigo":4,"nome":"Joyce","idade":25}, {"codigo":6,"nome":"Luke","idade":30}, {"codigo":8,"nome":"AnKIN","idade":22}, {"codigo":10,"nome":"Madara","idade":45}, {"codigo":12,"nome":"Sasuke","idade":28}, {"codigo":14,"nome":"Luna","idade":20}, {"codigo":16,"nome":"Edson","idade":55}, {"codigo":18,"nome":"Adao","idade":35}, {"codigo":20,"nome":"Noe","idade":40}, {"codigo":22,"nome":"Akilles", "idade":32}]
```

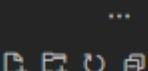
The right tab, also titled 'localhost:8080/api/nomeContem', displays a single JSON object:

```
{"codigo":3,"nome":"Carlos","idade":65}
```

Below the browser window, the text 'Eu tinha digitado errado' is displayed in green.



EXPLORER



ApiApplication.java

Repositorio.java

Con

||

^

v

↑

↓

↻

□

▼

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

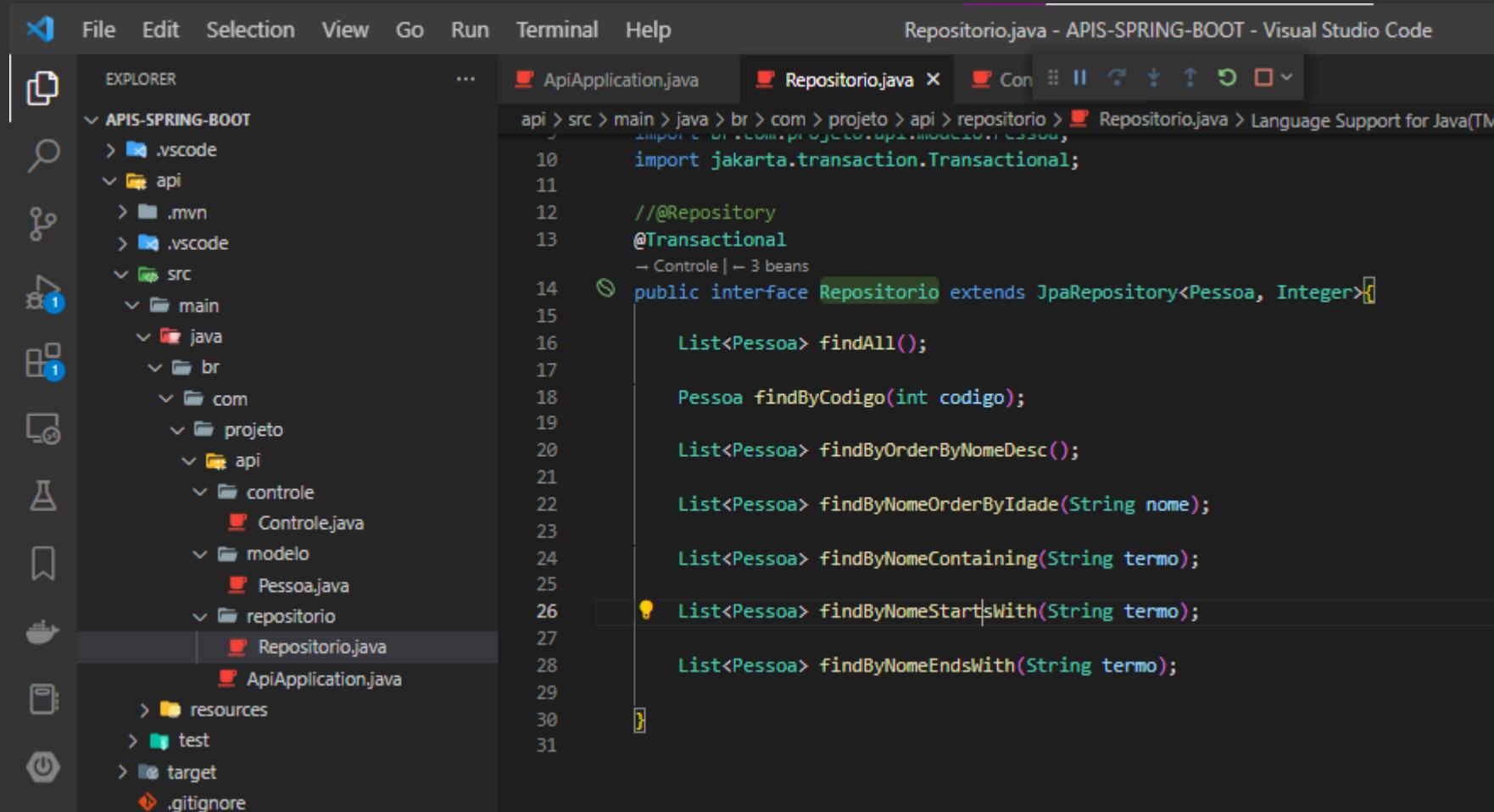
< HELP.md

< mvnw

< mvnw.cmd

< pom.xml

```
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
65    http://127.0.0.1:8080/api/nomeContem
66    @GetMapping("/api/nomeContem")
67    public List<Pessoa> nomeContem(){
68        return acao.findByNomeContaining(termo: "lo");
69    }
70
71    // @GetMapping("/api/iniciaCom")
72    // public List<Pessoa> iniciaCom(){
73    //     return acao.findByNomeStarsWith("a");
74    // }
75
76    // @GetMapping("/api/terminaCom")
77    // public List<Pessoa> terminaCom(){
78    //     return acao.findByNomeStarsWith("a");
79    // }
80
81    http://127.0.0.1:8080/
82    @GetMapping("/")
83    public String mensagem(){
        return "Hello World";
    }
84
85    http://127.0.0.1:8080/boasVindas
```



```
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

> .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

```
    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);

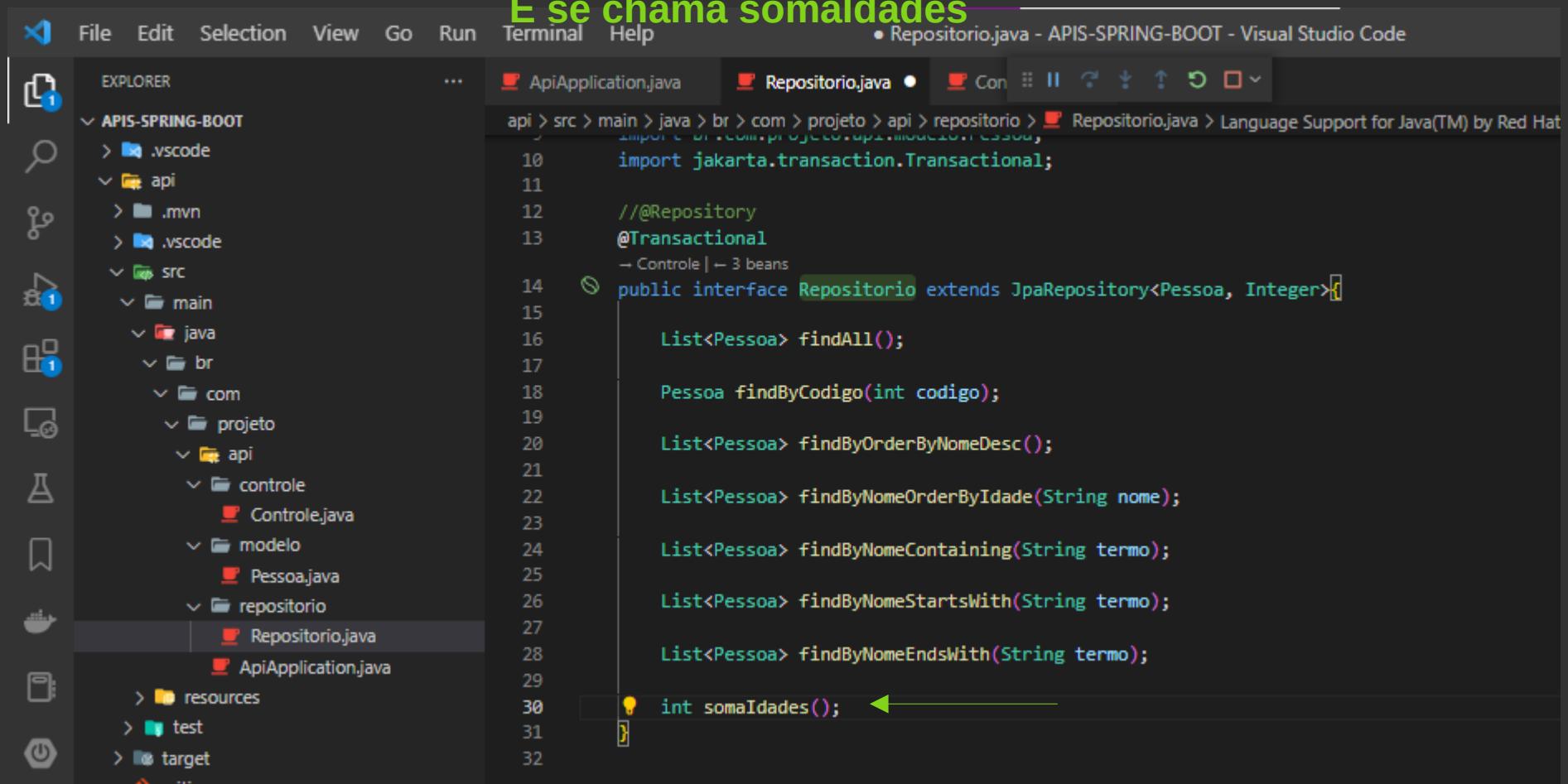
    List<Pessoa> findByNomeContaining(String termo);

    List<Pessoa> findByNomeStartsWith(String termo);

    List<Pessoa> findByNomeEndsWith(String termo);
}
```


#26
Utilizando annotation @Query

Criamos um metodo que retorna um numero inteiro E se chama somaldades



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

- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT". The "repositorio" folder contains "Repositorio.java" and "ApiApplication.java".
- Terminal (Top):** Shows the current file path: "api > src > main > java > br > com > projeto > api > repositorio > Repositorio.java".
- Code Editor (Right):** Displays the content of "Repositorio.java". The code defines an interface "Repositorio" that extends "JpaRepository<Pessoa, Integer>". It includes several methods for finding Pessoa objects based on various criteria. A new method, "somaIdades()", is being typed at the bottom of the code editor.

```
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> findByOrderByNomeDesc();
21
22         List<Pessoa> findByNomeOrderByIdade(String nome);
23
24         List<Pessoa> findByNomeContaining(String termo);
25
26         List<Pessoa> findByNomeStartsWith(String termo);
27
28         List<Pessoa> findByNomeEndsWith(String termo);
29
30         int somaIdades(); ←
31     }
```

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

com

▼ projeto

api

controle

 Control

▼ modelo

Pessoa.java

repositorio

Repositorio.java

Apr

res

> test

> target

↳ .gitignore

HELP.HTM

1

www.w3.org

```
api > src > main > java > br > com > projeto > api > repositorio > Repository.java > Language Support for Java(TM) by Red Hat > Repository > somidades
  5     import org.springframework.data.jpa.repository.JpaRepository;
  6     //import org.springframework.data.repository.CrudRepository;
  7     //import org.springframework.stereotype.Repository;
  8     import org.springframework.data.jpa.repository.Query; ◀
  9
 10    import br.com.projeto.api.modelo.Pessoa;
 11    import jakarta.transaction.Transactional;
 12
 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| ◀
 32        int somaIdades();
 33    }
```



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

...

```
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

.vscode

api

.mvn

.vscode

src

main

java

br

com

projeto

api

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 > Con  
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 World";  
87 }  
88 http://127.0.0.1:8080/b  
89 @GetMapping("/boas-vindas")  
90 public String boasVindas(){  
91     return "Seja bem-vindo!";  
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.

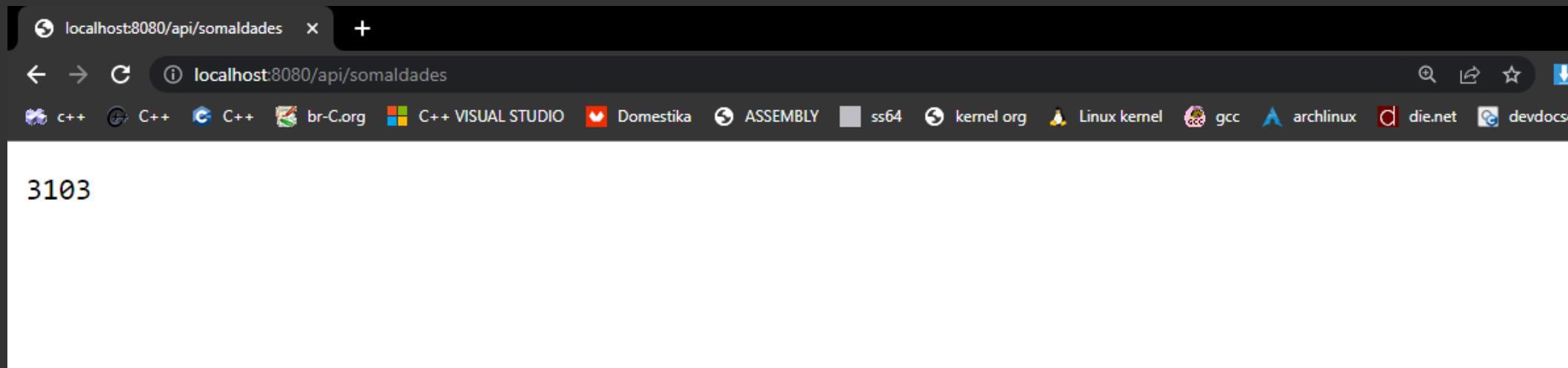
```
67    }
68
69    @GetMapping("/api/iniciaCom")
70    public List<Pessoa> iniciaCom(){
71        return acao.findByNomeStartsWith(termo: "a");
72    }
73
74    @GetMapping("/api/terminaCom")
75    public List<Pessoa> terminaCom(){
76        return acao.findByNomeEndsWith(termo: "a");
77    }
78
79    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 "api" folder is expanded, showing "controle" and "repository" packages, each containing a Java file: "Controle.java" and "Repository.java". The "src/main/java" tab is active, displaying the content of "Controle.java". The code implements several REST endpoints using annotations like @GetMapping and @PostMapping. A red arrow points from the line "return acao.somaIdades();" to the URL "http://127.0.0.1:8080/api/somaIdades" in the code.

```
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

> .vscode

v api

> .mvn

> .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

> resources

> test

> 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 ↴ ↑ ↵ □ ▾

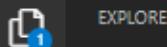
APIS-SPRING-BOOT

- .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

> .vscode

api

> .mvn

> .vscode

> src

> main

> java

> br

> com

> projeto

> api

> 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 > 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         ↑  
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("/ba  
95         public void m  
96             return "  
97         }  
98  
99         http://127.0.0.1:8080/ba  
100        @GetMapping("/ba  
101        public void m  
102            return "  
103        }  
104  
105        http://127.0.0.1:8080/ba  
106        @GetMapping("/ba  
107        public void m  
108            return "  
109        }  
110  
111        http://127.0.0.1:8080/ba  
112        @GetMapping("/ba  
113        public void m  
114            return "  
115        }  
116  
117        http://127.0.0.1:8080/ba  
118        @GetMapping("/ba  
119        public void m  
120            return "  
121        }  
122  
123        http://127.0.0.1:8080/ba  
124        @GetMapping("/ba  
125        public void m  
126            return "  
127        }  
128  
129        http://127.0.0.1:8080/ba  
130        @GetMapping("/ba  
131        public void m  
132            return "  
133        }  
134  
135        http://127.0.0.1:8080/ba  
136        @GetMapping("/ba  
137        public void m  
138            return "  
139        }  
140  
141        http://127.0.0.1:8080/ba  
142        @GetMapping("/ba  
143        public void m  
144            return "  
145        }  
146  
147        http://127.0.0.1:8080/ba  
148        @GetMapping("/ba  
149        public void m  
150            return "  
151        }  
152  
153        http://127.0.0.1:8080/ba  
154        @GetMapping("/ba  
155        public void m  
156            return "  
157        }  
158  
159        http://127.0.0.1:8080/ba  
160        @GetMapping("/ba  
161        public void m  
162            return "  
163        }  
164  
165        http://127.0.0.1:8080/ba  
166        @GetMapping("/ba  
167        public void m  
168            return "  
169        }  
170  
171        http://127.0.0.1:8080/ba  
172        @GetMapping("/ba  
173        public void m  
174            return "  
175        }  
176  
177        http://127.0.0.1:8080/ba  
178        @GetMapping("/ba  
179        public void m  
180            return "  
181        }  
182  
183        http://127.0.0.1:8080/ba  
184        @GetMapping("/ba  
185        public void m  
186            return "  
187        }  
188  
189        http://127.0.0.1:8080/ba  
190        @GetMapping("/ba  
191        public void m  
192            return "  
193        }  
194  
195        http://127.0.0.1:8080/ba  
196        @GetMapping("/ba  
197        public void m  
198            return "  
199        }  
200  
201        http://127.0.0.1:8080/ba  
202        @GetMapping("/ba  
203        public void m  
204            return "  
205        }  
206  
207        http://127.0.0.1:8080/ba  
208        @GetMapping("/ba  
209        public void m  
210            return "  
211        }  
212  
213        http://127.0.0.1:8080/ba  
214        @GetMapping("/ba  
215        public void m  
216            return "  
217        }  
218  
219        http://127.0.0.1:8080/ba  
220        @GetMapping("/ba  
221        public void m  
222            return "  
223        }  
224  
225        http://127.0.0.1:8080/ba  
226        @GetMapping("/ba  
227        public void m  
228            return "  
229        }  
230  
231        http://127.0.0.1:8080/ba  
232        @GetMapping("/ba  
233        public void m  
234            return "  
235        }  
236  
237        http://127.0.0.1:8080/ba  
238        @GetMapping("/ba  
239        public void m  
240            return "  
241        }  
242  
243        http://127.0.0.1:8080/ba  
244        @GetMapping("/ba  
245        public void m  
246            return "  
247        }  
248  
249        http://127.0.0.1:8080/ba  
250        @GetMapping("/ba  
251        public void m  
252            return "  
253        }  
254  
255        http://127.0.0.1:8080/ba  
256        @GetMapping("/ba  
257        public void m  
258            return "  
259        }  
260  
261        http://127.0.0.1:8080/ba  
262        @GetMapping("/ba  
263        public void m  
264            return "  
265        }  
266  
267        http://127.0.0.1:8080/ba  
268        @GetMapping("/ba  
269        public void m  
270            return "  
271        }  
272  
273        http://127.0.0.1:8080/ba  
274        @GetMapping("/ba  
275        public void m  
276            return "  
277        }  
278  
279        http://127.0.0.1:8080/ba  
280        @GetMapping("/ba  
281        public void m  
282            return "  
283        }  
284  
285        http://127.0.0.1:8080/ba  
286        @GetMapping("/ba  
287        public void m  
288            return "  
289        }  
290  
291        http://127.0.0.1:8080/ba  
292        @GetMapping("/ba  
293        public void m  
294            return "  
295        }  
296  
297        http://127.0.0.1:8080/ba  
298        @GetMapping("/ba  
299        public void m  
300            return "  
301        }  
302  
303        http://127.0.0.1:8080/ba  
304        @GetMapping("/ba  
305        public void m  
306            return "  
307        }  
308  
309        http://127.0.0.1:8080/ba  
310        @GetMapping("/ba  
311        public void m  
312            return "  
313        }  
314  
315        http://127.0.0.1:8080/ba  
316        @GetMapping("/ba  
317        public void m  
318            return "  
319        }  
320  
321        http://127.0.0.1:8080/ba  
322        @GetMapping("/ba  
323        public void m  
324            return "  
325        }  
326  
327        http://127.0.0.1:8080/ba  
328        @GetMapping("/ba  
329        public void m  
330            return "  
331        }  
332  
333        http://127.0.0.1:8080/ba  
334        @GetMapping("/ba  
335        public void m  
336            return "  
337        }  
338  
339        http://127.0.0.1:8080/ba  
340        @GetMapping("/ba  
341        public void m  
342            return "  
343        }  
344  
345        http://127.0.0.1:8080/ba  
346        @GetMapping("/ba  
347        public void m  
348            return "  
349        }  
350  
351        http://127.0.0.1:8080/ba  
352        @GetMapping("/ba  
353        public void m  
354            return "  
355        }  
356  
357        http://127.0.0.1:8080/ba  
358        @GetMapping("/ba  
359        public void m  
360            return "  
361        }  
362  
363        http://127.0.0.1:8080/ba  
364        @GetMapping("/ba  
365        public void m  
366            return "  
367        }  
368  
369        http://127.0.0.1:8080/ba  
370        @GetMapping("/ba  
371        public void m  
372            return "  
373        }  
374  
375        http://127.0.0.1:8080/ba  
376        @GetMapping("/ba  
377        public void m  
378            return "  
379        }  
380  
381        http://127.0.0.1:8080/ba  
382        @GetMapping("/ba  
383        public void m  
384            return "  
385        }  
386  
387        http://127.0.0.1:8080/ba  
388        @GetMapping("/ba  
389        public void m  
390            return "  
391        }  
392  
393        http://127.0.0.1:8080/ba  
394        @GetMapping("/ba  
395        public void m  
396            return "  
397        }  
398  
399        http://127.0.0.1:8080/ba  
400        @GetMapping("/ba  
401        public void m  
402            return "  
403        }  
404  
405        http://127.0.0.1:8080/ba  
406        @GetMapping("/ba  
407        public void m  
408            return "  
409        }  
410  
411        http://127.0.0.1:8080/ba  
412        @GetMapping("/ba  
413        public void m  
414            return "  
415        }  
416  
417        http://127.0.0.1:8080/ba  
418        @GetMapping("/ba  
419        public void m  
420            return "  
421        }  
422  
423        http://127.0.0.1:8080/ba  
424        @GetMapping("/ba  
425        public void m  
426            return "  
427        }  
428  
429        http://127.0.0.1:8080/ba  
430        @GetMapping("/ba  
431        public void m  
432            return "  
433        }  
434  
435        http://127.0.0.1:8080/ba  
436        @GetMapping("/ba  
437        public void m  
438            return "  
439        }  
440  
441        http://127.0.0.1:8080/ba  
442        @GetMapping("/ba  
443        public void m  
444            return "  
445        }  
446  
447        http://127.0.0.1:8080/ba  
448        @GetMapping("/ba  
449        public void m  
450            return "  
451        }  
452  
453        http://127.0.0.1:8080/ba  
454        @GetMapping("/ba  
455        public void m  
456            return "  
457        }  
458  
459        http://127.0.0.1:8080/ba  
460        @GetMapping("/ba  
461        public void m  
462            return "  
463        }  
464  
465        http://127.0.0.1:8080/ba  
466        @GetMapping("/ba  
467        public void m  
468            return "  
469        }  
470  
471        http://127.0.0.1:8080/ba  
472        @GetMapping("/ba  
473        public void m  
474            return "  
475        }  
476  
477        http://127.0.0.1:8080/ba  
478        @GetMapping("/ba  
479        public void m  
480            return "  
481        }  
482  
483        http://127.0.0.1:8080/ba  
484        @GetMapping("/ba  
485        public void m  
486            return "  
487        }  
488  
489        http://127.0.0.1:8080/ba  
490        @GetMapping("/ba  
491        public void m  
492            return "  
493        }  
494  
495        http://127.0.0.1:8080/ba  
496        @GetMapping("/ba  
497        public void m  
498            return "  
499        }  
500  
501        http://127.0.0.1:8080/ba  
502        @GetMapping("/ba  
503        public void m  
504            return "  
505        }  
506  
507        http://127.0.0.1:8080/ba  
508        @GetMapping("/ba  
509        public void m  
510            return "  
511        }  
512  
513        http://127.0.0.1:8080/ba  
514        @GetMapping("/ba  
515        public void m  
516            return "  
517        }  
518  
519        http://127.0.0.1:8080/ba  
520        @GetMapping("/ba  
521        public void m  
522            return "  
523        }  
524  
525        http://127.0.0.1:8080/ba  
526        @GetMapping("/ba  
527        public void m  
528            return "  
529        }  
530  
531        http://127.0.0.1:8080/ba  
532        @GetMapping("/ba  
533        public void m  
534            return "  
535        }  
536  
537        http://127.0.0.1:8080/ba  
538        @GetMapping("/ba  
539        public void m  
540            return "  
541        }  
542  
543        http://127.0.0.1:8080/ba  
544        @GetMapping("/ba  
545        public void m  
546            return "  
547        }  
548  
549        http://127.0.0.1:8080/ba  
550        @GetMapping("/ba  
551        public void m  
552            return "  
553        }  
554  
555        http://127.0.0.1:8080/ba  
556        @GetMapping("/ba  
557        public void m  
558            return "  
559        }  
560  
561        http://127.0.0.1:8080/ba  
562        @GetMapping("/ba  
563        public void m  
564            return "  
565        }  
566  
567        http://127.0.0.1:8080/ba  
568        @GetMapping("/ba  
569        public void m  
570            return "  
571        }  
572  
573        http://127.0.0.1:8080/ba  
574        @GetMapping("/ba  
575        public void m  
576            return "  
577        }  
578  
579        http://127.0.0.1:8080/ba  
580        @GetMapping("/ba  
581        public void m  
582            return "  
583        }  
584  
585        http://127.0.0.1:8080/ba  
586        @GetMapping("/ba  
587        public void m  
588            return "  
589        }  
590  
591        http://127.0.0.1:8080/ba  
592        @GetMapping("/ba  
593        public void m  
594            return "  
595        }  
596  
597        http://127.0.0.1:8080/ba  
598        @GetMapping("/ba  
599        public void m  
600            return "  
601        }  
602  
603        http://127.0.0.1:8080/ba  
604        @GetMapping("/ba  
605        public void m  
606            return "  
607        }  
608  
609        http://127.0.0.1:8080/ba  
610        @GetMapping("/ba  
611        public void m  
612            return "  
613        }  
614  
615        http://127.0.0.1:8080/ba  
616        @GetMapping("/ba  
617        public void m  
618            return "  
619        }  
620  
621        http://127.0.0.1:8080/ba  
622        @GetMapping("/ba  
623        public void m  
624            return "  
625        }  
626  
627        http://127.0.0.1:8080/ba  
628        @GetMapping("/ba  
629        public void m  
630            return "  
631        }  
632  
633        http://127.0.0.1:8080/ba  
634        @GetMapping("/ba  
635        public void m  
636            return "  
637        }  
638  
639        http://127.0.0.1:8080/ba  
640        @GetMapping("/ba  
641        public void m  
642            return "  
643        }  
644  
645        http://127.0.0.1:8080/ba  
646        @GetMapping("/ba  
647        public void m  
648            return "  
649        }  
650  
651        http://127.0.0.1:8080/ba  
652        @GetMapping("/ba  
653        public void m  
654            return "  
655        }  
656  
657        http://127.0.0.1:8080/ba  
658        @GetMapping("/ba  
659        public void m  
660            return "  
661        }  
662  
663        http://127.0.0.1:8080/ba  
664        @GetMapping("/ba  
665        public void m  
666            return "  
667        }  
668  
669        http://127.0.0.1:8080/ba  
670        @GetMapping("/ba  
671        public void m  
672            return "  
673        }  
674  
675        http://127.0.0.1:8080/ba  
676        @GetMapping("/ba  
677        public void m  
678            return "  
679        }  
680  
681        http://127.0.0.1:8080/ba  
682        @GetMapping("/ba  
683        public void m  
684            return "  
685        }  
686  
687        http://127.0.0.1:8080/ba  
688        @GetMapping("/ba  
689        public void m  
690            return "  
691        }  
692  
693        http://127.0.0.1:8080/ba  
694        @GetMapping("/ba  
695        public void m  
696            return "  
697        }  
698  
699        http://127.0.0.1:8080/ba  
700        @GetMapping("/ba  
701        public void m  
702            return "  
703        }  
704  
705        http://127.0.0.1:8080/ba  
706        @GetMapping("/ba  
707        public void m  
708            return "  
709        }  
710  
711        http://127.0.0.1:8080/ba  
712        @GetMapping("/ba  
713        public void m  
714            return "  
715        }  
716  
717        http://127.0.0.1:8080/ba  
718        @GetMapping("/ba  
719        public void m  
720            return "  
721        }  
722  
723        http://127.0.0.1:8080/ba  
724        @GetMapping("/ba  
725        public void m  
726            return "  
727        }  
728  
729        http://127.0.0.1:8080/ba  
730        @GetMapping("/ba  
731        public void m  
732            return "  
733        }  
734  
735        http://127.0.0.1:8080/ba  
736        @GetMapping("/ba  
737        public void m  
738            return "  
739        }  
740  
741        http://127.0.0.1:8080/ba  
742        @GetMapping("/ba  
743        public void m  
744            return "  
745        }  
746  
747        http://127.0.0.1:8080/ba  
748        @GetMapping("/ba  
749        public void m  
750            return "  
751        }  
752  
753        http://127.0.0.1:8080/ba  
754        @GetMapping("/ba  
755        public void m  
756            return "  
757        }  
758  
759        http://127.0.0.1:8080/ba  
760        @GetMapping("/ba  
761        public void m  
762            return "  
763        }  
764  
765        http://127.0.0.1:8080/ba  
766        @GetMapping("/ba  
767        public void m  
768            return "  
769        }  
770  
771        http://127.0.0.1:8080/ba  
772        @GetMapping("/ba  
773        public void m  
774            return "  
775        }  
776  
777        http://127.0.0.1:8080/ba  
778        @GetMapping("/ba  
779        public void m  
780            return "  
781        }  
782  
783        http://127.0.0.1:8080/ba  
784        @GetMapping("/ba  
785        public void m  
786            return "  
787        }  
788  
789        http://127.0.0.1:8080/ba  
790        @GetMapping("/ba  
791        public void m  
792            return "  
793        }  
794  
795        http://127.0.0.1:8080/ba  
796        @GetMapping("/ba  
797        public void m  
798            return "  
799        }  
800  
801        http://127.0.0.1:8080/ba  
802        @GetMapping("/ba  
803        public void m  
804            return "  
805        }  
806  
807        http://127.0.0.1:8080/ba  
808        @GetMapping("/ba  
809        public void m  
810            return "  
811        }  
812  
813        http://127.0.0.1:8080/ba  
814        @GetMapping("/ba  
815        public void m  
816            return "  
817        }  
818  
819        http://127.0.0.1:8080/ba  
820        @GetMapping("/ba  
821        public void m  
822            return "  
823        }  
824  
825        http://127.0.0.1:8080/ba  
826        @GetMapping("/ba  
827        public void m  
828            return "  
829        }  
830  
831        http://127.0.0.1:8080/ba  
832        @GetMapping("/ba  
833        public void m  
834            return "  
835        }  
836  
837        http://127.0.0.1:8080/ba  
838        @GetMapping("/ba  
839        public void m  
840            return "  
841        }  
842  
843        http://127.0.0.1:8080/ba  
844        @GetMapping("/ba  
845        public void m  
846            return "  
847        }  
848  
849        http://127.0.0.1:8080/ba  
850        @GetMapping("/ba  
851        public void m  
852            return "  
853        }  
854  
855        http://127.0.0.1:8080/ba  
856        @GetMapping("/ba  
857        public void m  
858            return "  
859        }  
860  
861        http://127.0.0.1:8080/ba  
862        @GetMapping("/ba  
863        public void m  
864            return "  
865        }  
866  
867        http://127.0.0.1:8080/ba  
868        @GetMapping("/ba  
869        public void m  
870            return "  
871        }  
872  
873        http://127.0.0.1:8080/ba  
874        @GetMapping("/ba  
875        public void m  
876            return "  
877        }  
878  
879        http://127.0.0.1:8080/ba  
880        @GetMapping("/ba  
881        public void m  
882            return "  
883        }  
884  
885        http://127.0.0.1:8080/ba  
886        @GetMapping("/ba  
887        public void m  
888            return "  
889        }  
890  
891        http://127.0.0.1:8080/ba  
892        @GetMapping("/ba  
893        public void m  
894            return "  
895        }  
896  
897        http://127.0.0.1:8080/ba  
898        @GetMapping("/ba  
899        public void m  
900            return "  
901        }  
902  
903        http://127.0.0.1:8080/ba  
904        @GetMapping("/ba  
905        public void m  
906            return "  
907        }  
908  
909        http://127.0.0.1:8080/ba  
910        @GetMapping("/ba  
911        public void m  
912            return "  
913        }  
914  
915        http://127.0.0.1:8080/ba  
916        @GetMapping("/ba  
917        public void m  
918            return "  
919        }  
920  
921        http://127.0.0.1:8080/ba  
922        @GetMapping("/ba  
923        public void m  
924            return "  
925        }  
926  
927        http://127.0.0.1:8080/ba  
928        @GetMapping("/ba  
929        public void m  
930            return "  
931        }  
932  
933        http://127.0.0.1:8080/ba  
934        @GetMapping("/ba  
935        public void m  
936            return "  
937        }  
938  
939        http://127.0.0.1:8080/ba  
940        @GetMapping("/ba  
941        public void m  
942            return "  
943        }  
944  
945        http://127.0.0.1:8080/ba  
946        @GetMapping("/ba  
947        public void m  
948            return "  
949        }  
950  
951        http://127.0.0.1:8080/ba  
952        @GetMapping("/ba  
953        public void m  
954            return "  
955        }  
956  
957        http://127.0.0.1:8080/ba  
958        @GetMapping("/ba  
959        public void m  
960            return "  
961        }  
962  
963        http://127.0.0.1:8080/ba  
964        @GetMapping("/ba  
965        public void m  
966            return "  
967        }  
968  
969        http://127.0.0.1:8080/ba  
970        @GetMapping("/ba  
971        public void m  
972            return "  
973        }  
974  
975        http://127.0.0.1:8080/ba  
976        @GetMapping("/ba  
977        public void m  
978            return "  
979        }  
980  
981        http://127.0.0.1:8080/ba  
982        @GetMapping("/ba  
983        public void m  
984            return "  
985        }  
986  
987        http://127.0.0.1:8080/ba  
988        @GetMapping("/ba  
989        public void m  
990            return "  
991        }  
992  
993        http://127.0.0.1:8080/ba  
994        @GetMapping("/ba  
995        public void m  
996            return "  
997        }  
998  
999        http://127.0.0.1:8080/ba  
1000        @GetMapping("/ba  
1001        public void m  
1002            return "  
1003        }  
1004  
1005        http://127.0.0.1:8080/ba  
1006        @GetMapping("/ba  
1007        public void m  
1008            return "  
1009        }  
1010  
1011        http://127.0.0.1:8080/ba  
1012        @GetMapping("/ba  
1013        public void m  
1014            return "  
1015        }  
1016  
1017        http://127.0.0.1:8080/ba  
1018        @GetMapping("/ba  
1019        public void m  
1020            return "  
1021        }  
1022  
1023        http://127.0.0.1:8080/ba  
1024        @GetMapping("/ba  
1025        public void m  
1026            return "  
1027        }  
1028  
1029        http://127.0.0.1:8080/ba  
1030        @GetMapping("/ba  
1031        public void m  
1032            return "  
1033        }  
1034  
1035        http://127.0.0.1:8080/ba  
1036        @GetMapping("/ba  
1037        public void m  
1038            return "  
1039        }  
1040  
1041        http://127.0.0.1:8080/ba  
1042        @GetMapping("/ba  
1043        public void m  
1044            return "  
1045        }  
1046  
1047        http://127.0.0.1:8080/ba  
1048        @GetMapping("/ba  
1049        public void m  
1050            return "  
1051        }  
1052  
1053        http://127.0.0.1:8080/ba  
1054        @GetMapping("/ba  
1055        public void m  
1056            return "  
1057        }  
1058  
1059        http://127.0.0.1:8080/ba  
1060        @GetMapping("/ba  
1061        public void m  
1062            return "  
1063        }  
1064  
1065        http://127.0.0.1:8080/ba  
1066        @GetMapping("/ba  
1067        public void m  
1068            return "  
1069        }  
1070  
1071        http://127.0.0.1:8080/ba  
1072        @GetMapping("/ba  
1073        public void m  
1074            return "  
1075        }  
1076  
1077        http://127.0.0.1:8080/ba  
1078        @GetMapping("/ba  
1079        public void m  
1080            return "  
1081        }  
1082  
1083        http://127.0.0.1:8080/ba  
1084        @GetMapping("/ba  
1085        public void m  
1086            return "  
1087        }  
1088  
1089        http://127.0.0.1:8080/ba  
1090        @GetMapping("/ba  
1091        public void m  
1092            return "  
1093        }  
1094  
1095        http://127.0.0.1:8080/ba  
1096        @GetMapping("/ba  
1097        public void m  
1098            return "  
1099        }  
1100  
1101        http://127.0.0.1:8080/ba  
1102        @GetMapping("/ba  
1103        public void m  
1104            return "  
1105        }  
1106  
1107        http://127.0.0.1:8080/ba  
1108        @GetMapping("/ba  
1109        public void m  
1110            return "  
1111        }  
1112  
1113        http://127.0.0.1:8080/ba  
1114        @GetMapping("/ba  
1115        public void m  
1116            return "  
1117        }  
1118
```



EXPLORER



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



mvnw



mvnw.cmd



pom.xml



README.md

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.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

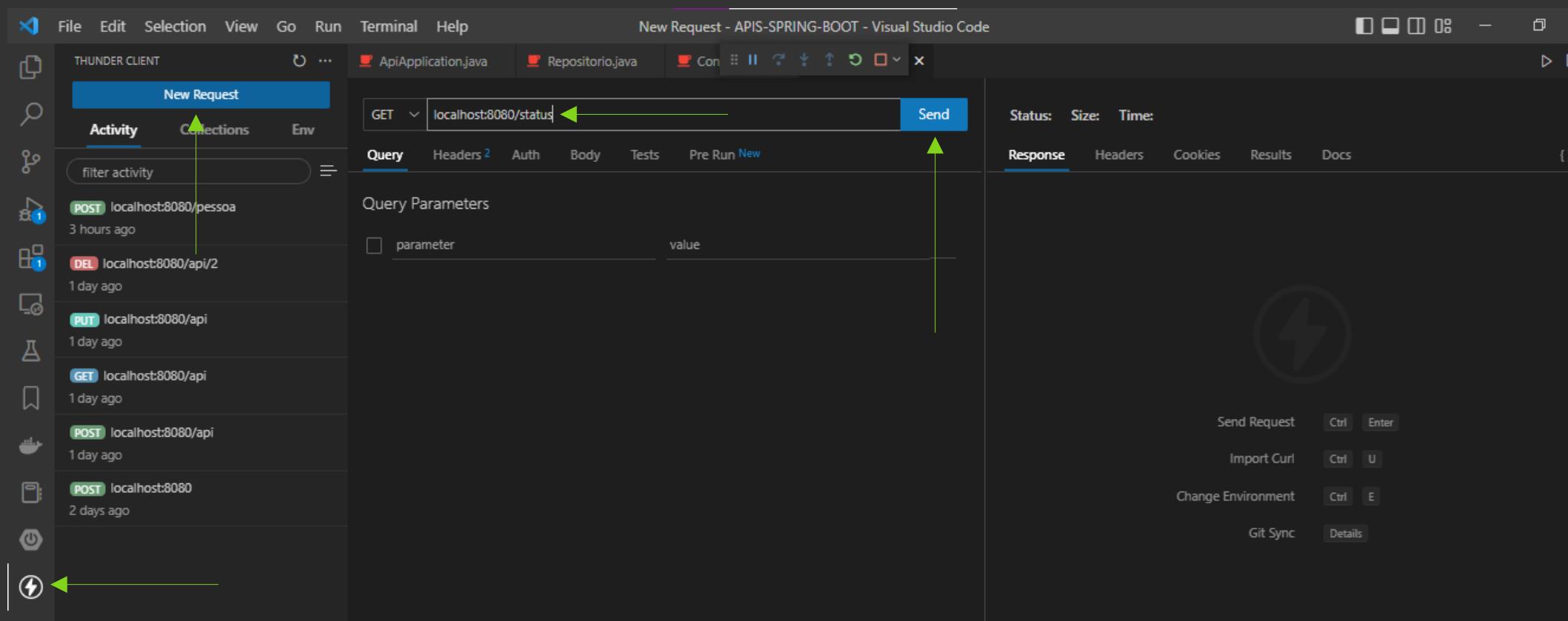
Controle.java

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 "pessoa". The "pessoa" endpoint uses a RequestBody annotation. 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 "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



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 }

```
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

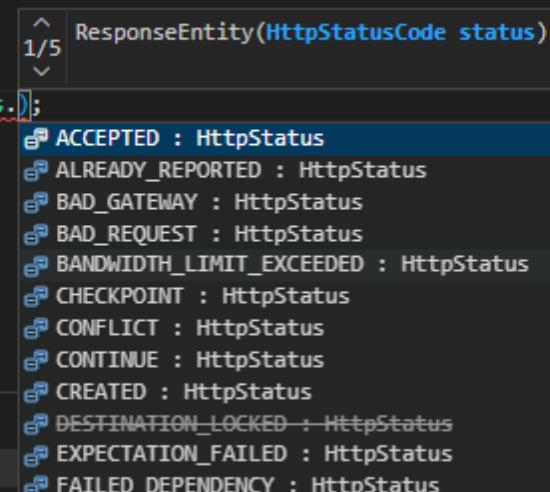
TERMINAL

```
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 }
```



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
```

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.

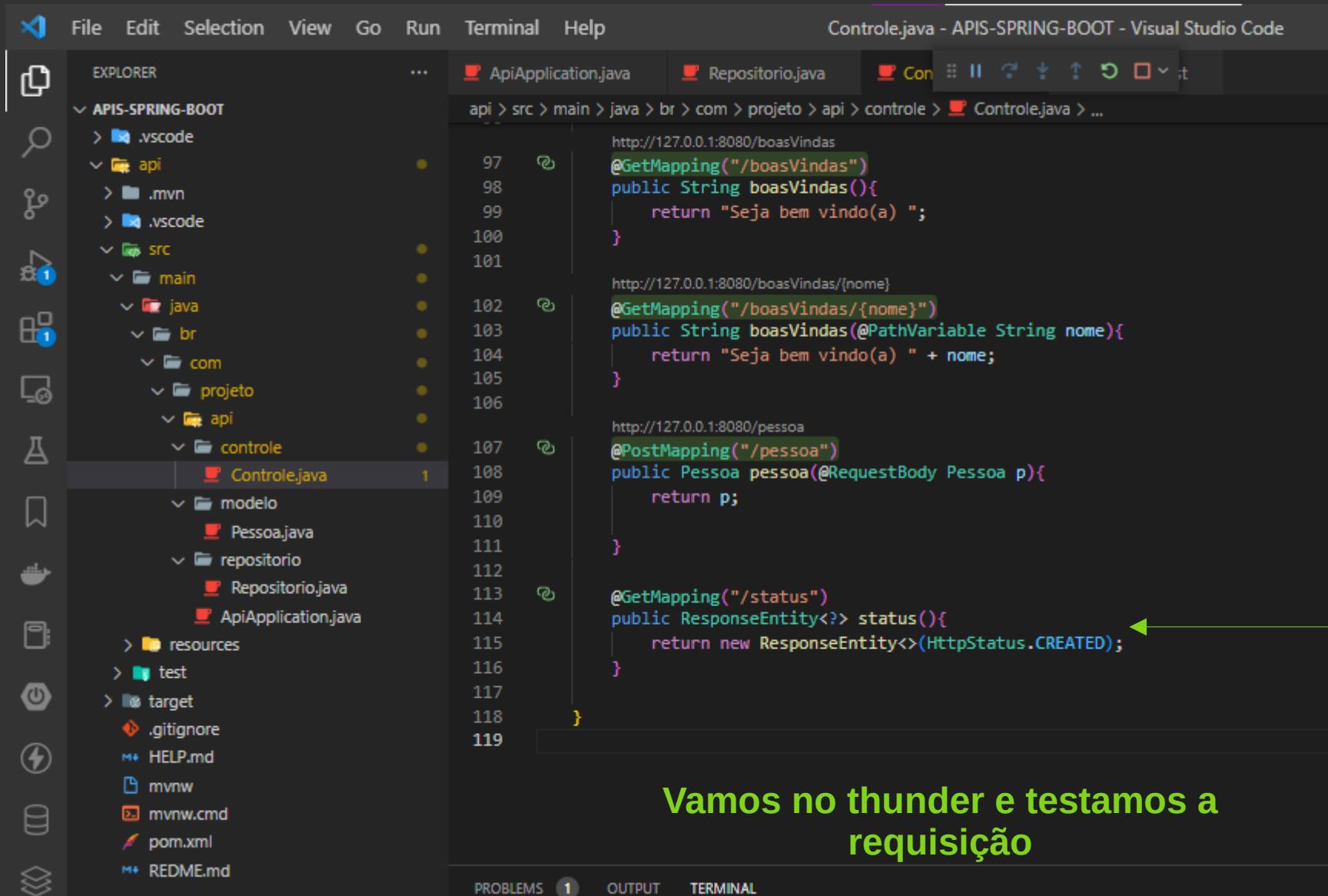
PROBLEMS 1 OUTPUT TERMINAL

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

This 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 ApiApplication.java, Repositorio.java, and Con. A green arrow points down to the main request panel. The panel shows a 'New Request' tab selected. The method is set to GET, and the URL is localhost:8080/status. The 'Send' button is highlighted in blue. To the right, the response details are shown: Status: 201 Created, Size: 0 Bytes, and Time: 41 ms. Below the URL input, there are tabs for Query, Headers (2), Auth, Body, Tests, Pre Run, and New. The 'Query' tab is active. On the left, a sidebar lists recent activities: a GET request to localhost:8080/status (11 mins ago), a POST request to localhost:8080/pessoa (3 hours ago), a DEL request to localhost:8080/api/2 (1 day ago), a PUT request to localhost:8080/api (1 day ago), and another GET request to localhost:8080/api (1 day ago). The bottom section is titled 'Query Parameters' with a table header for parameter and value.

#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 Spring Boot project named "APIS-SPRING-BOOT".

File Explorer:

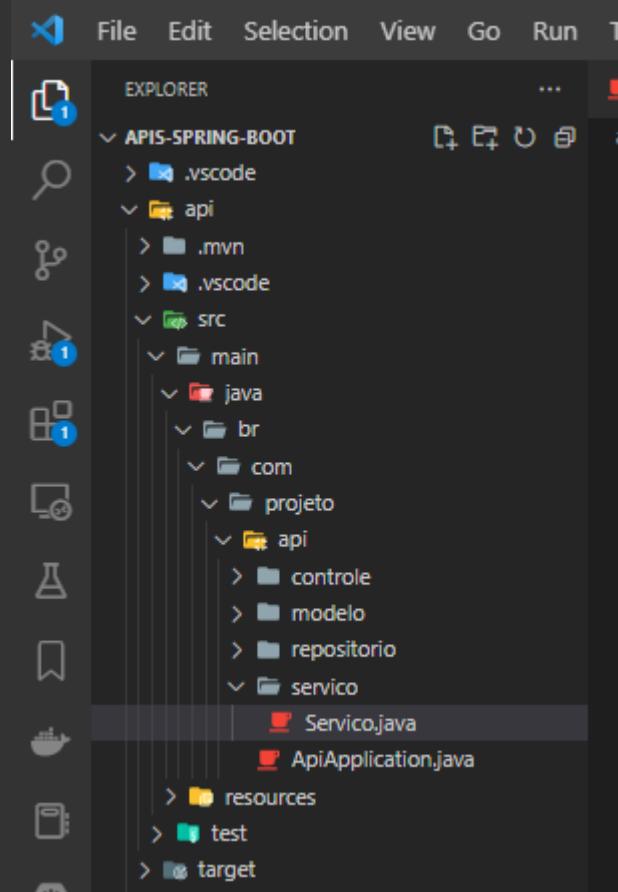
- Project structure:
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - modelo
 - repositorio
 - servico
- Resources: resources, test, target, .gitignore, HELP.md

Code Editor:

```
api > src > main > java > br > com > projeto
  2
  3     import java.util.List;
  4
  5 import org.springframework
  6 import org.springframework
  7 import org.springframework
  8 import org.springframework
  9 import org.springframework
 10 import org.springframework
 11 import org.springframework
 12 import org.springframework
 13 import org.springframework
 14 import org.springframework
 15
 16 import br.com.projeto.ap
 17 import br.com.projeto.ap
 18
```

Contextual Menu (opened at the bottom of the "servico" folder):

- New File...
- New Folder...
- Reveal in File Explorer
- Open in Integrated Terminal
- Add Folder to Java Source Path



**Criamos a pasta serviço e dentro dela o arquivo
Serviço.java**



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

< README.md

...

ApiApplication.java

Repositorio.java

Con



New Request

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.

Annotation Description:

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

> .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

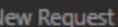
> README.md

...

ApiApplication.java

Repositorio.java

Con



```
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, explaining its purpose and usage.
- Terminal:** Shows "Con" (Connection).
- Status Bar:** Mensagem.java 1, New Request.

Tooltip Content:

- Annotation:** org.springframework.stereotype.Component
- Description:** 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 Annotations:** 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.
- Since:** 2.5

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".
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - modelo
 - resources
 - test
- Mensagem.java File:** The current file being edited.

```
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
```
- Status Bar:** Shows the path: api > src > main > java > br > com > projeto > api > modelo > Mensagem.java > Language Support for Java(TM) by Red Hat > Mensagem.java 1 TC N.

Selezione 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

> .vscode

< api

> .mvn

> .vscode

< src

< main

< java

< br

< com

< projeto

< api

> controle

< modelo

< Mensagem.java

< Pessoa.java

> repositorio

< 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 > modelo > Mensagem.java > ...

```
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

> .vscode

> api

> .mvn

> .vscode

> src

> main

> java

> br

> com

> projeto

> api

> controle

> modelo

> Mensagem.java

> Pessoa.java

> repository

> servico

> Servico.java

> ApiApplication.java

> resources

> test

> target

> .gitignore

> HELP.md

> mvnw

> mvnw.cmd

> pom.xml

> README.md

ApiApplication.java

Repositorio.java

Con

Mensagem.java

New Request

New Request

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

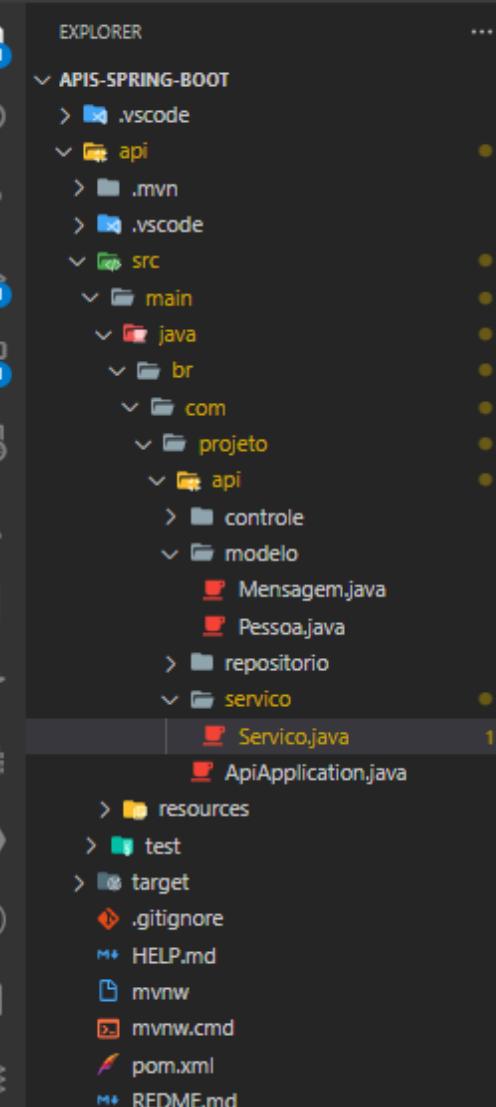
org.springframework.beans.factory.annotation.Autowired

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.

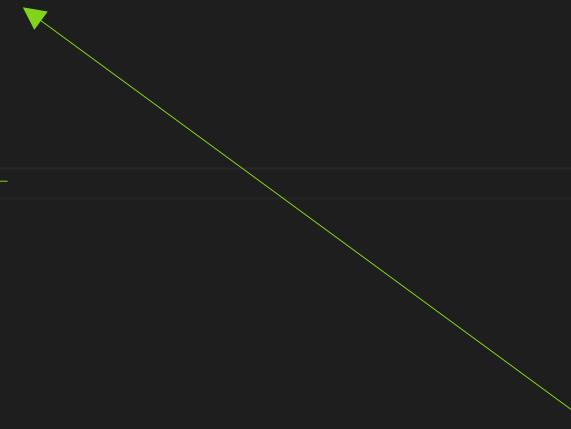
Autowired Constructors

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

