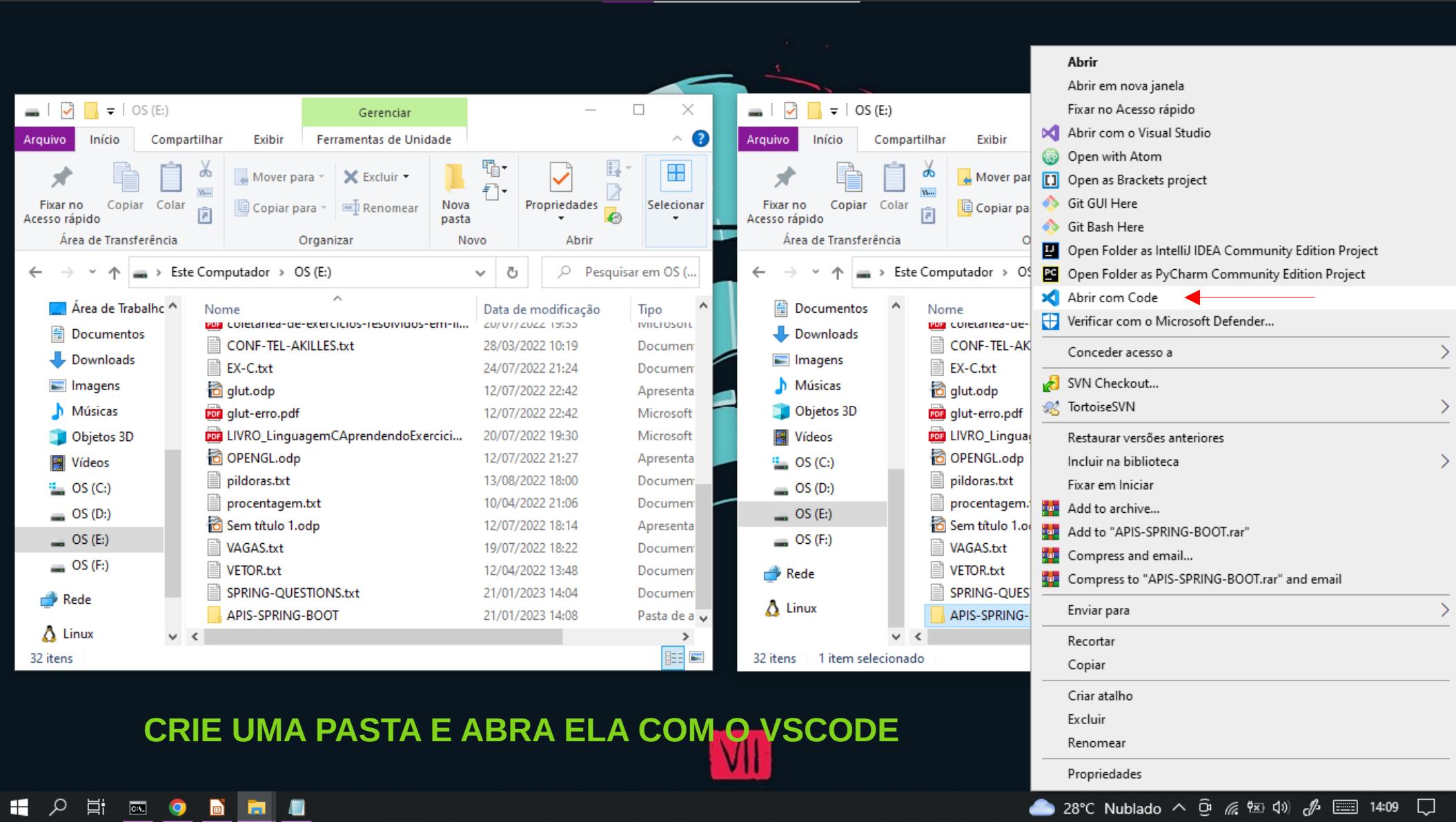
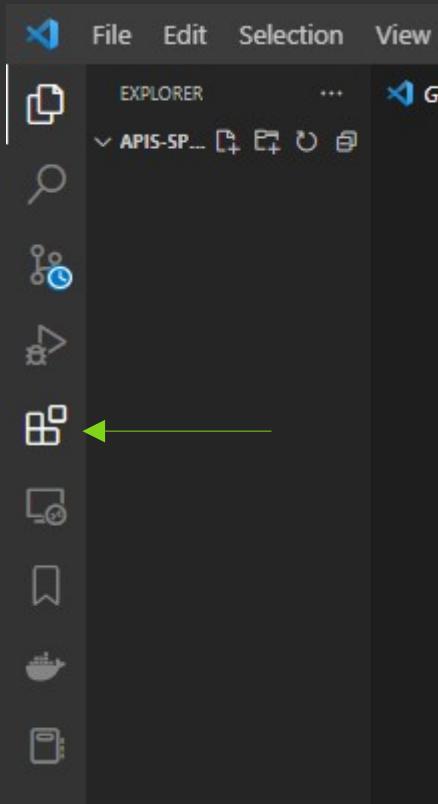


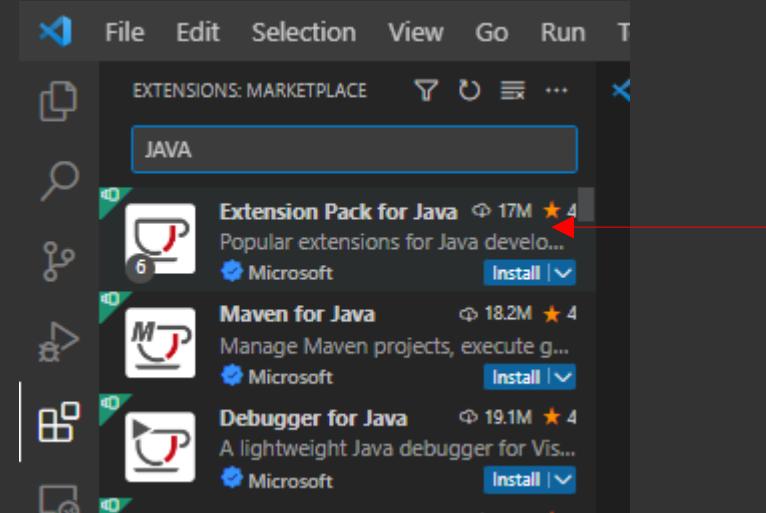
**JAVA SPRING BOOT
2023
NANO**

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)

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



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](#) [Formatters](#)

[Extension Packs](#)

Extension Resources

[Marketplace](#)

[Repository](#)

[License](#)

[Microsoft](#)

More Info

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

Last released [11/29/2022, 02:30:00](#)

File Edit Selection View Go Run Terminal Help

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

EXTENSIONS: MARKETPLACE JAVA

Install New JDK × Get Started ×

Install New JDK

Adoptium's Temurin Others

Version

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

JVM

- hotspot

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

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

Reload Window Having trouble?

Get Started with Java Development

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

Get your runtime ready
The Extension Pack for Java requires at least one Java runtime to be installed.
[Install JDK](#)

Explore your project

Launch, debug and test

Extensions for additional tools and frameworks

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

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`)

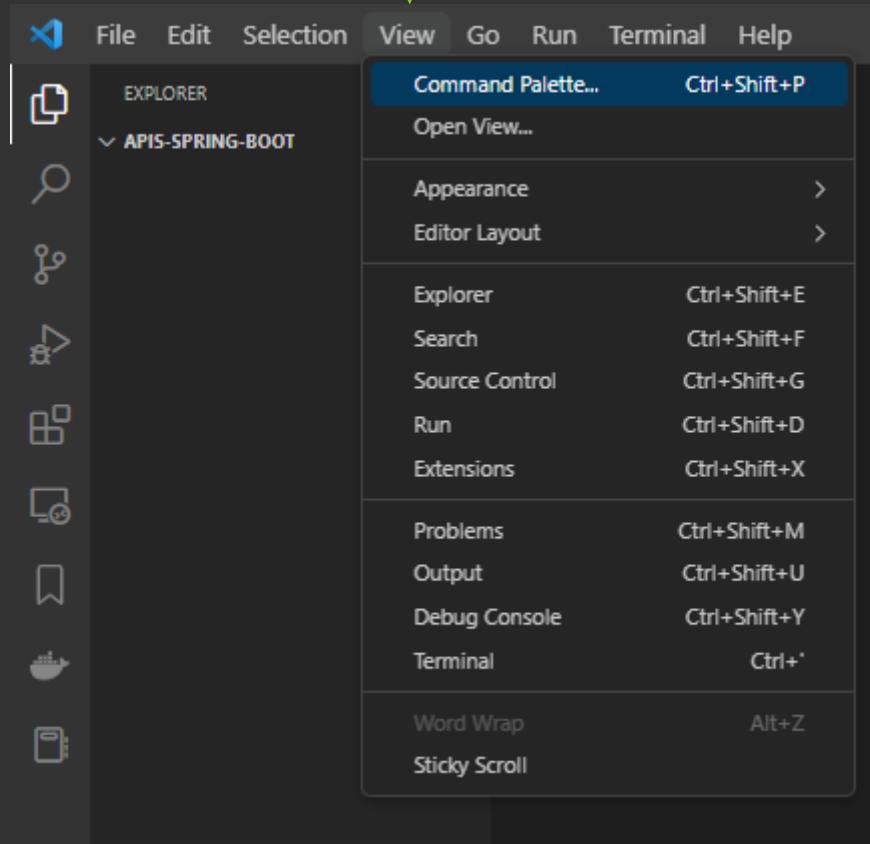
Spring Initializr Java

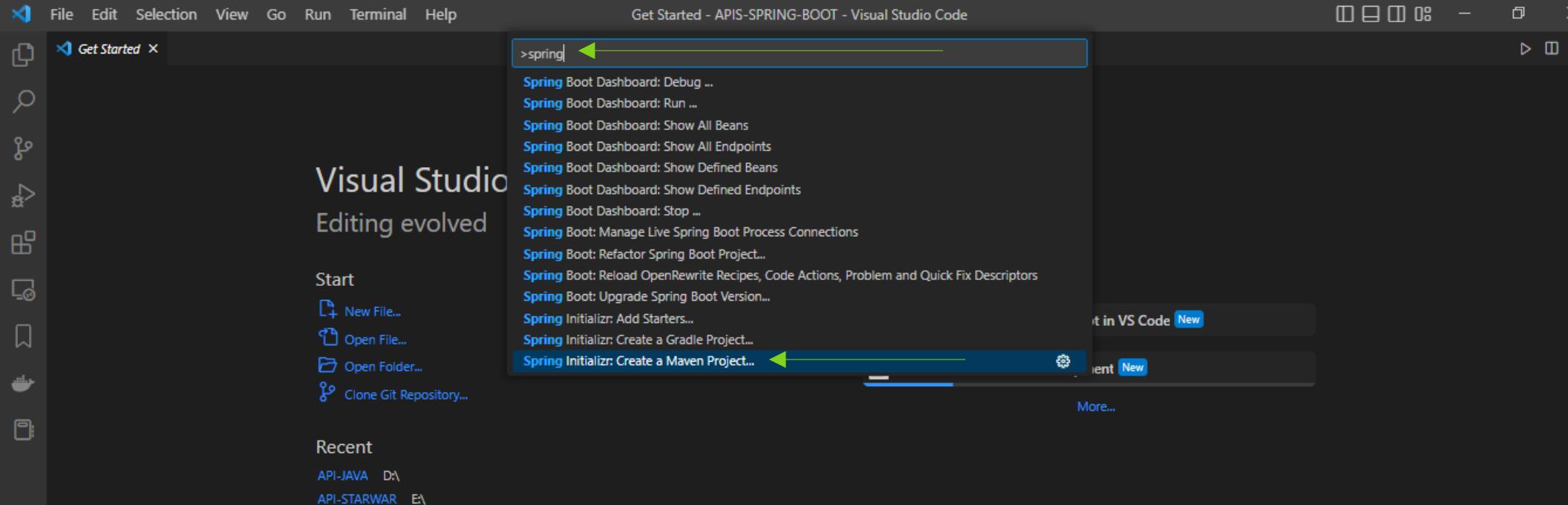
The [Spring Initializr Java Support](#) extension provides support for generating quickstart Spring Boot Java projects with Spring Initializr API.

Spring Boot Dashboard

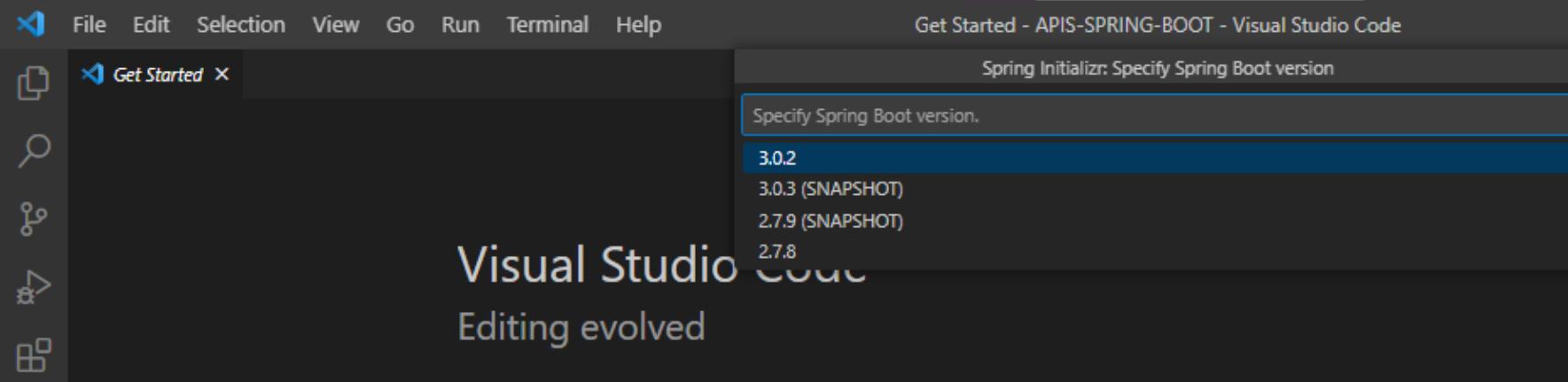
25°C Nublado Go Live 14:22

CRIANDO O PROJETO

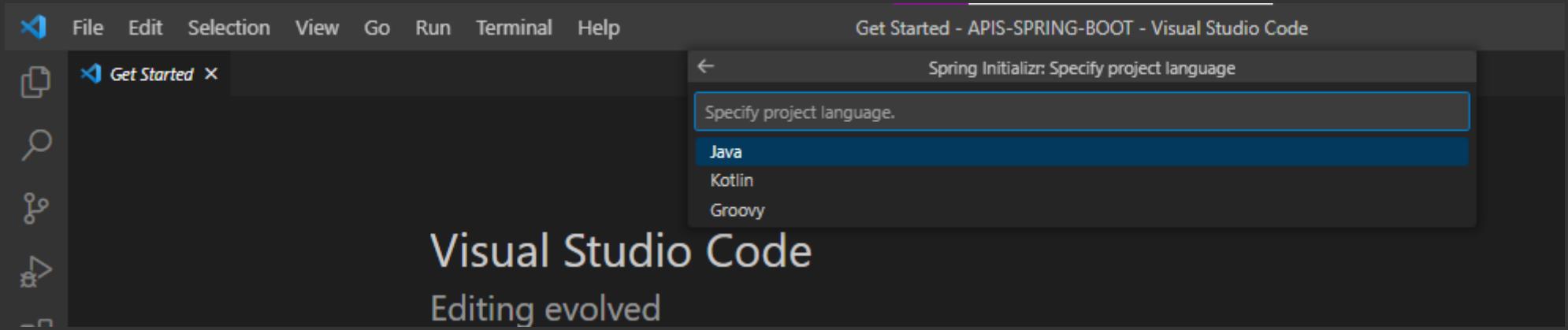




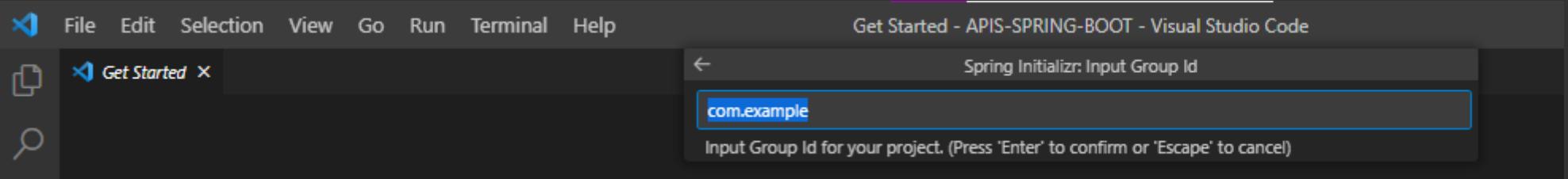
Digite spring e adicione spring initializr do maven
Essa que esta marcada ai



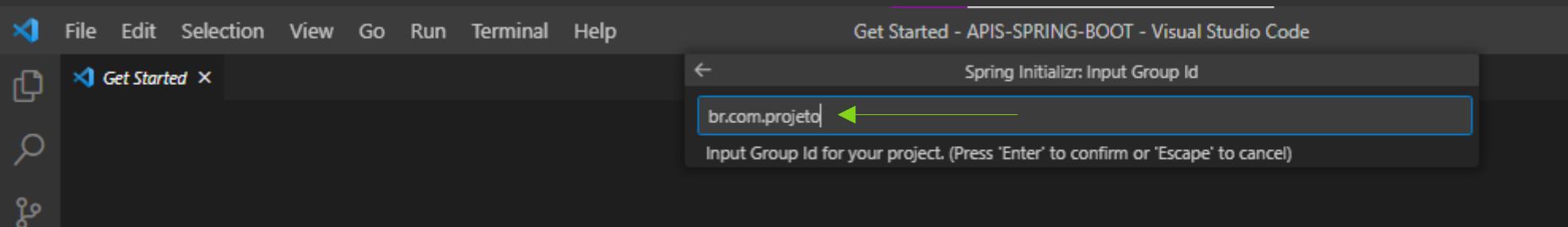
Escolha sua versão nada impede de troca-la mais tarde



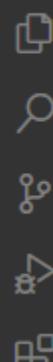
Escolha a linguagem do projeto



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



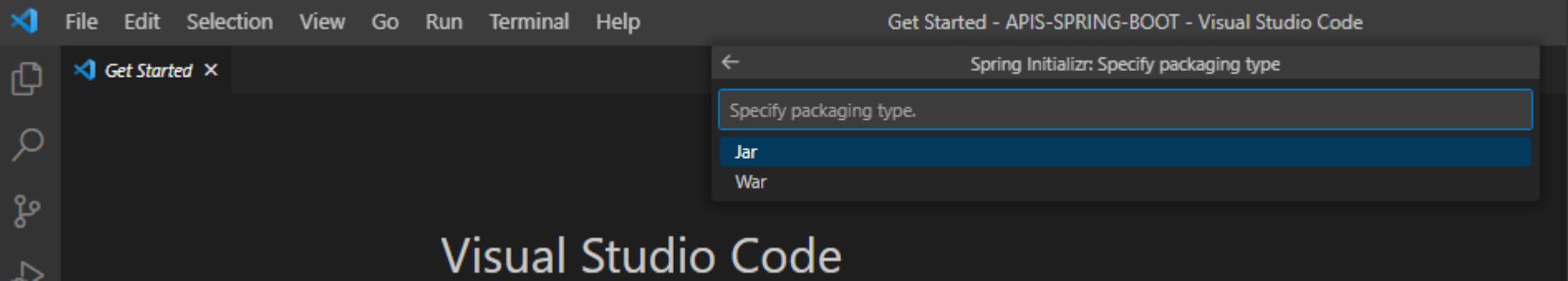
Spring Initializr: Input Artifact Id

api

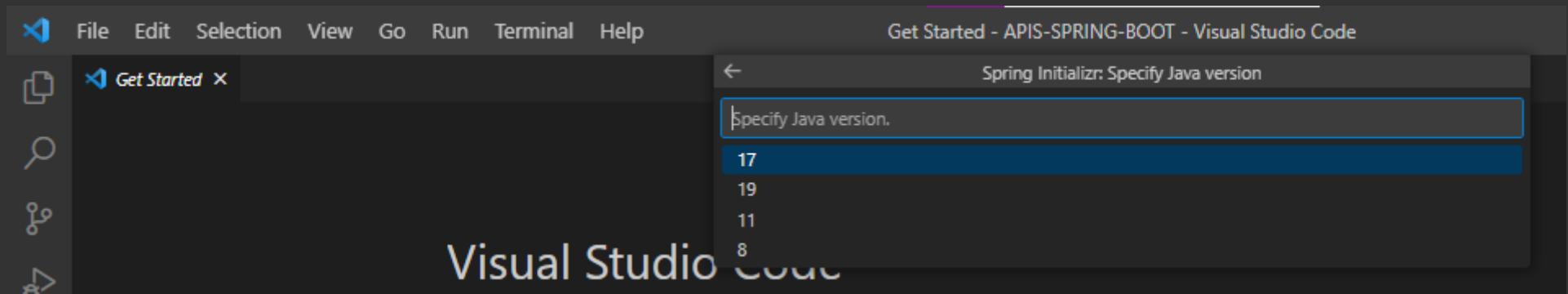
Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

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

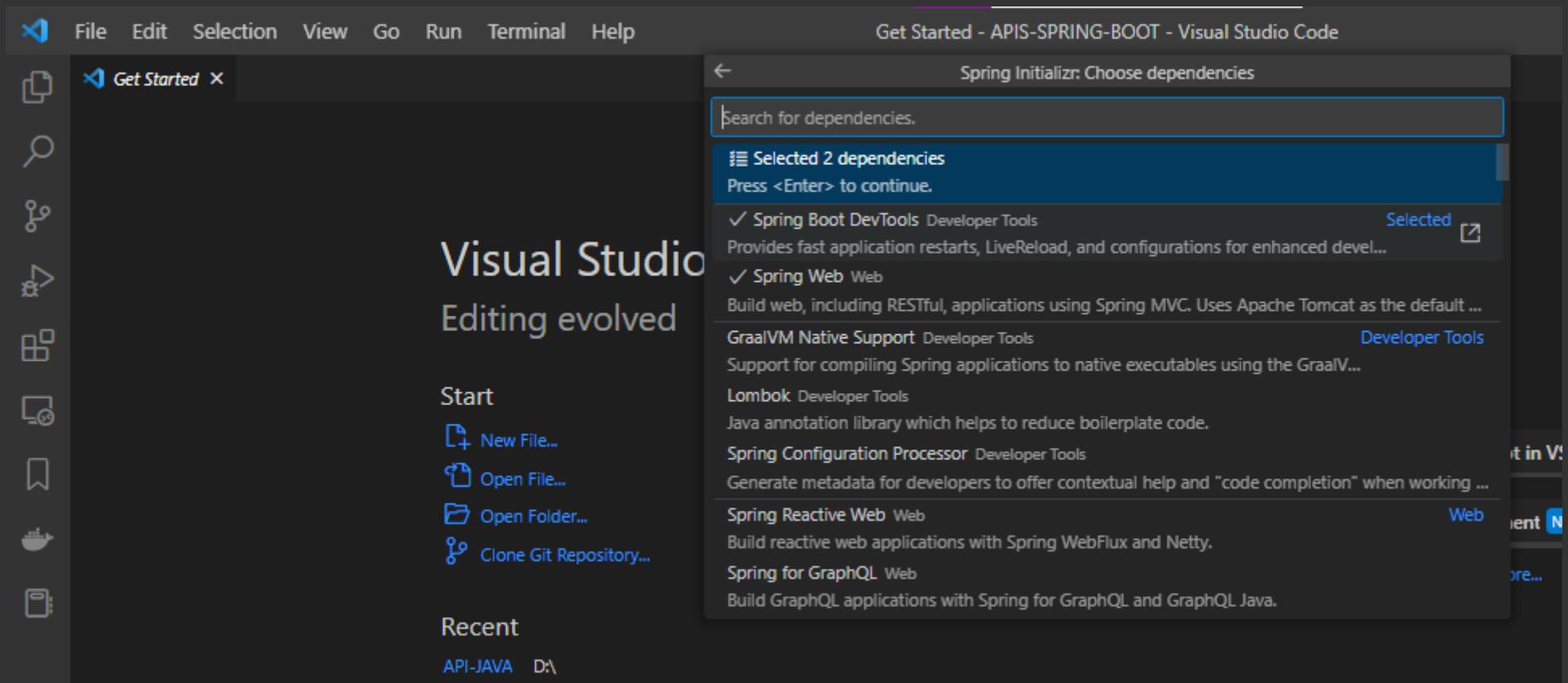
Build GraphQL applications with Spring for GraphQL and GraphQL Java.

it in VS Code New

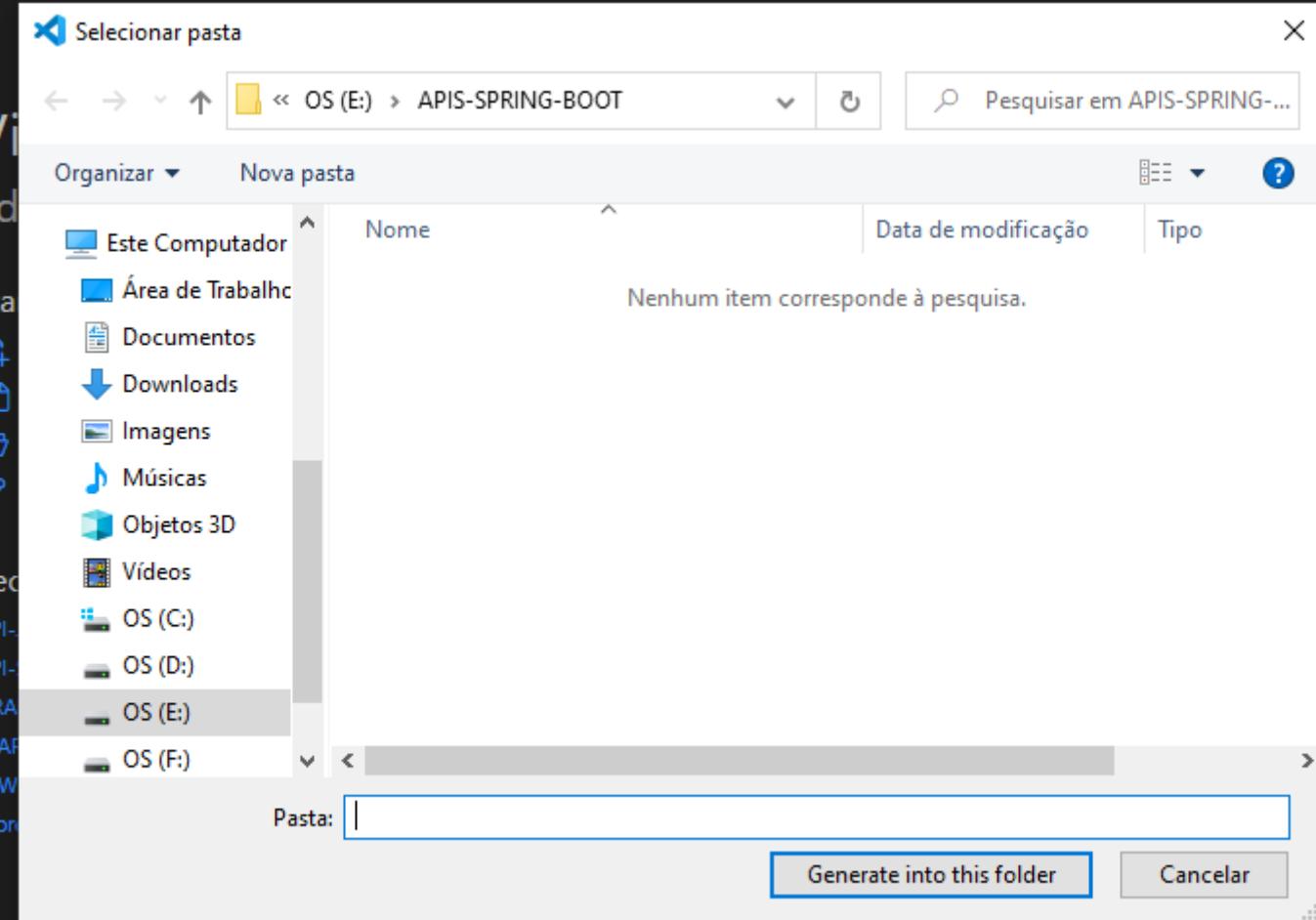
ent New

ore...

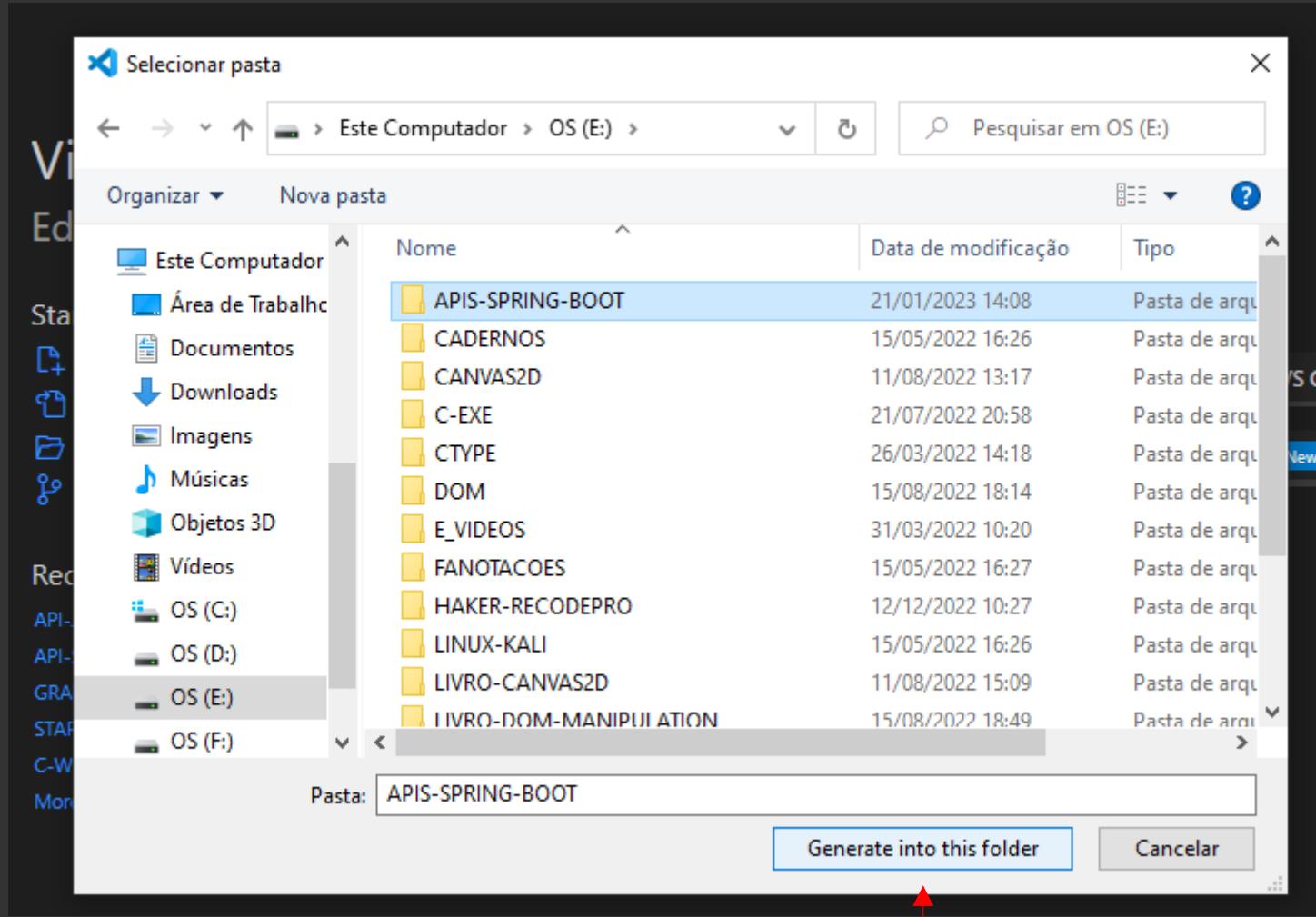
Adicione agora as dependencias do projeto
O que são dependencias? São funções que o spring usa

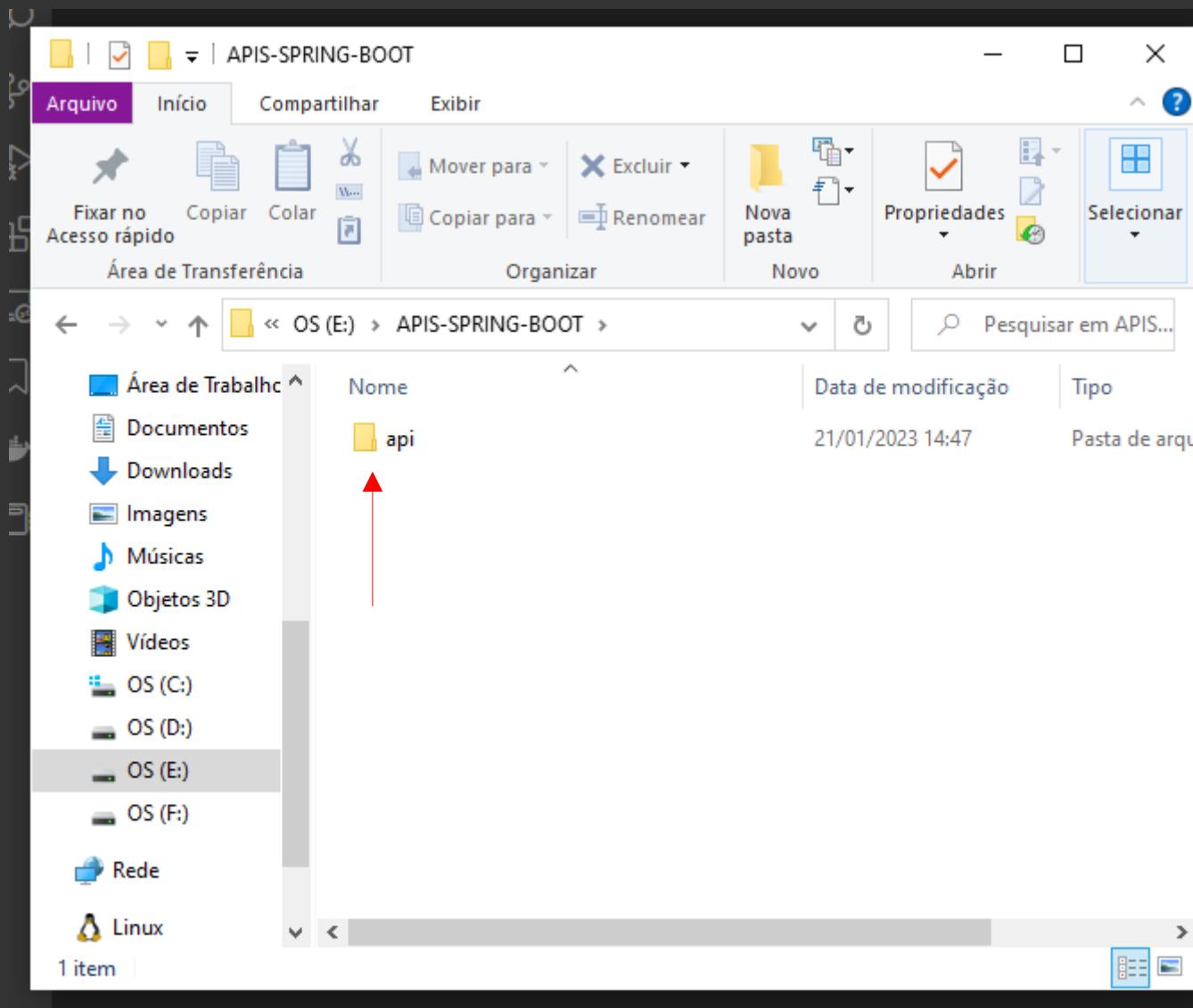


**Click para selecionar spring boot dev tools e spring web
Por fim de enter para selecionar**

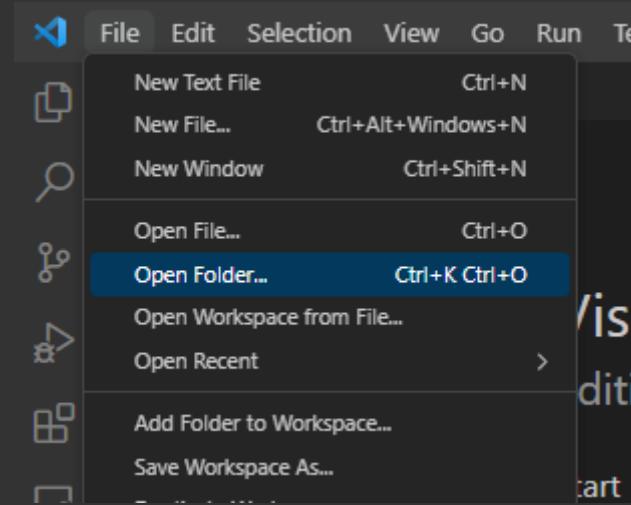


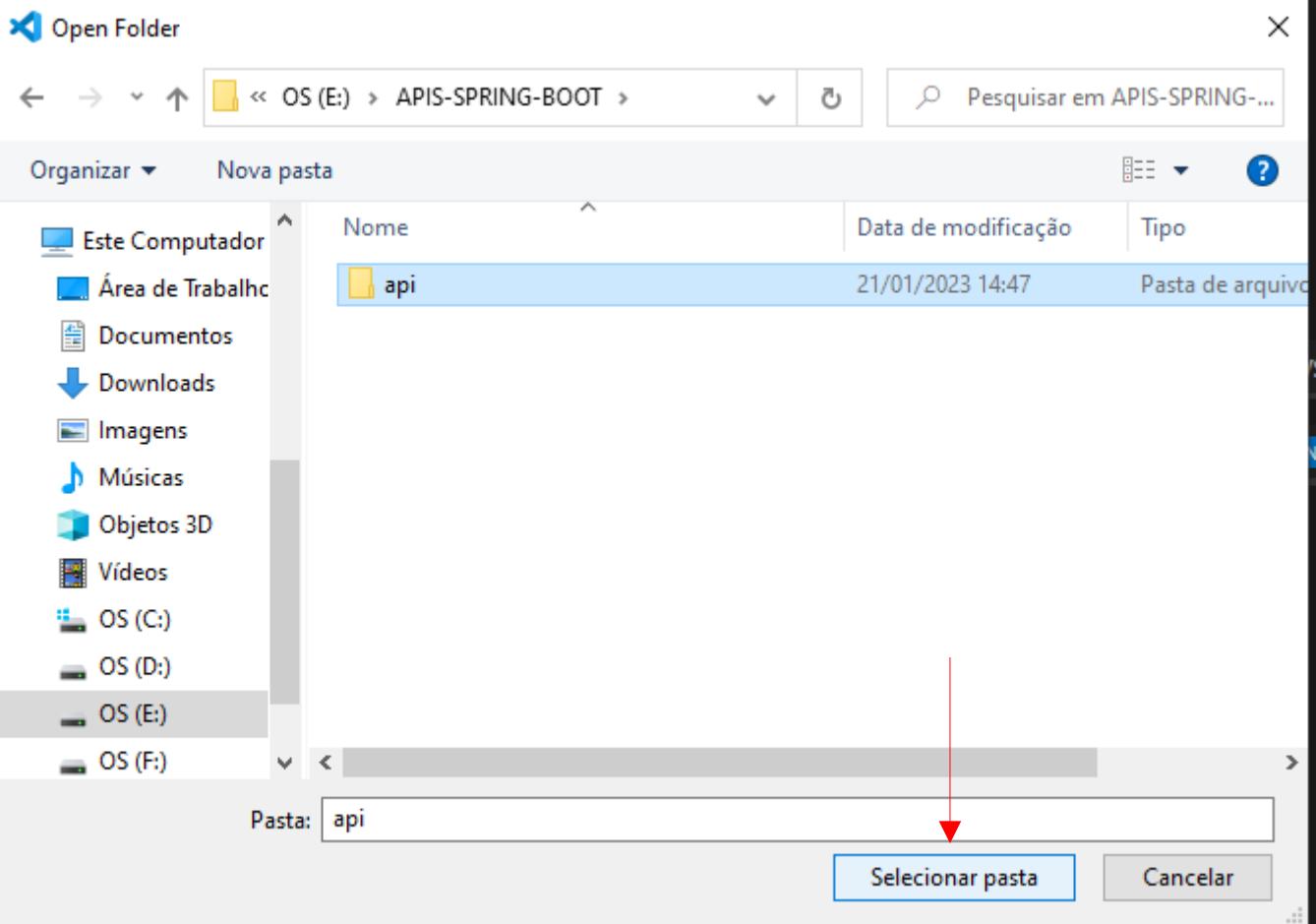
Ele abre para você escolher onde vai salvar o projeto
Você pode escolher qualquer pasta para salvar seu projeto





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...](#)

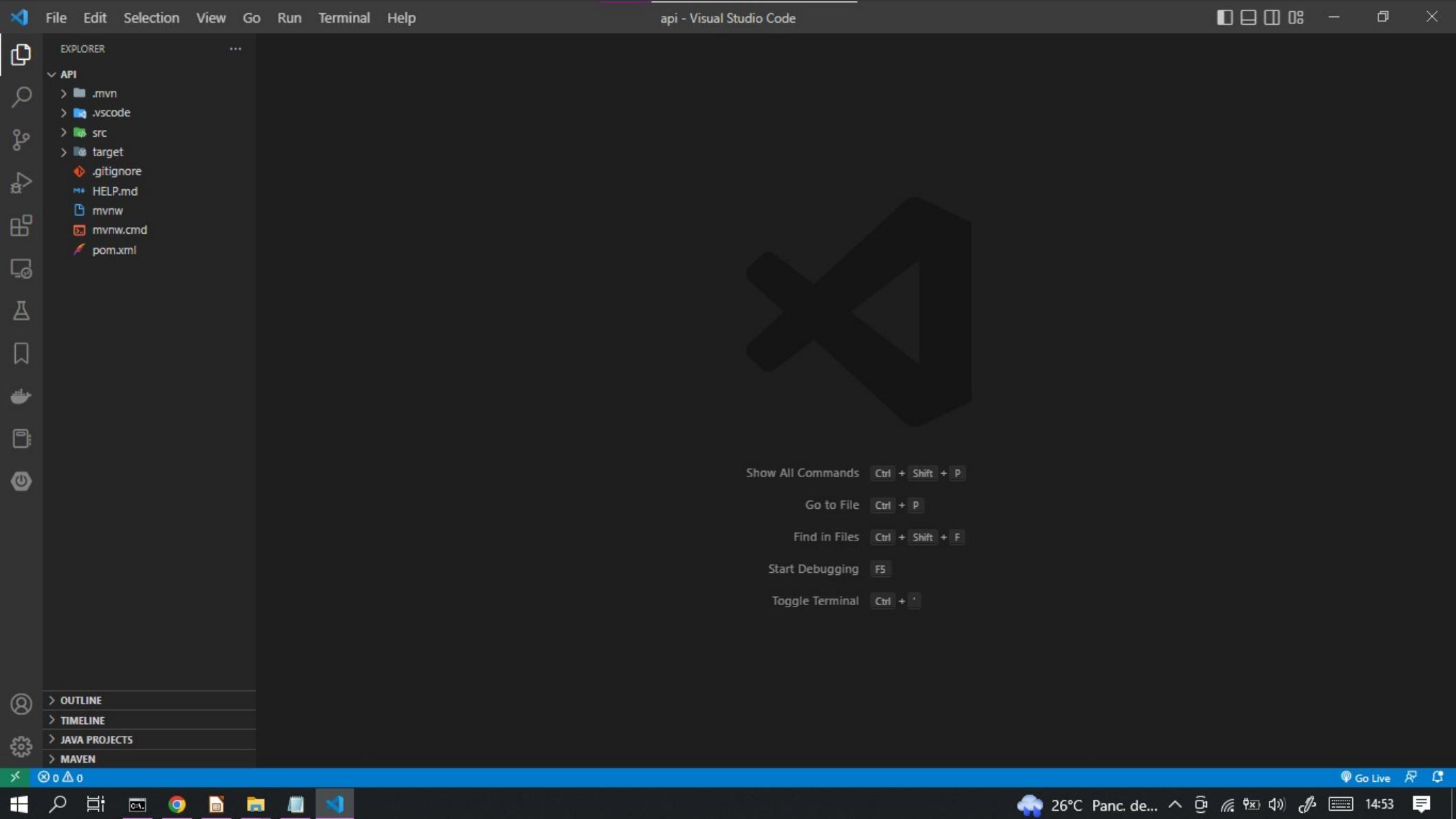
Walkthroughs

- Getting Started with Spring Boot in VS Code

- Get Started with Java Development

[More...](#)

Espere todos os arquivos serem baixados



File Edit Selection View Go Run Terminal Help

api - Visual Studio Code

EXPLORER

...

API

> .mvn
> .vscode
> src
> target
 |.gitignore
 | HELP.md
 | mvnw
 | mvnw.cmd
 | pom.xml

OUTLINE

TIMELINE

JAVA PROJECTS

MAVEN

0 0 0 0

Show All Commands **Ctrl + Shift + P**

Go to File **Ctrl + P**

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

Start Debugging **F5**

Toggle Terminal **Ctrl + `**

26°C Panc. de... 14:53

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 X

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

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

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



File Edit Selection View Go Run Terminal Help

ApiApplication.java - api - Visual Studio Code



EXPLORER

...

ApiApplication.java X

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

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



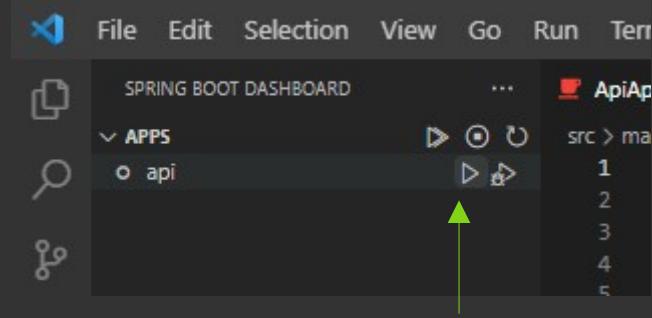
Nessa opção spring boot dashboard

A screenshot of a code editor showing a Spring Boot application structure. The left sidebar shows a project tree with 'SPRING BOOT DASHBOARD' at the root, expanded to show 'APPS' and 'api'. The main editor area shows the 'ApiApplication.java' file:

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

The status bar at the bottom of the editor shows the path: 'C:\Users\...'. A green arrow points from the bottom left towards the power icon in the bottom right corner of the editor window.

E rodamos o projeto
Tanto faz qualquer uma das escolhas



EXECUTE O PROJETO PARA FAZER UM TESTE

SPRING BOOT DASHBOARD

➤ APPS

o api

ApiApplication.java

root.xml

dom ym

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
3     <modelVersion>4.0.0</modelVersion>
4     <parent>
5       <groupId>org.springframework.boot</groupId>
6       <artifactId>spring-boot-starter-parent</artifactId>
7       <version>3.0.2</version>
8       <relativePath/> 
9     </parent>
10  </project>
```

PROBLEMS **OUTPUT** **TERMINAL**

TERMINAL

```
2023-01-21T15:10:28.415-03:00 INFO 968 --- [ restartedMain] br.com.projeto.api.ApiApplication : Starting ApiApplication using Java 17.0.5 with PID 968 (E:\APIS-SPRING-BOOT\api\target\classes started by brito in E:\APIS-SPRING-BOOT\api)
2023-01-21T15:10:28.424-03:00 INFO 968 --- [ restartedMain] br.com.projeto.api.ApiApplication : No active profile set, falling back to 1 default profile: "default"
2023-01-21T15:10:28.920-03:00 INFO 968 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-01-21T15:10:28.924-03:00 INFO 968 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-21T15:10:28.937-03:00 INFO 968 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.5]
2023-01-21T15:10:28.991-03:00 INFO 968 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-01-21T15:10:28.992-03:00 INFO 968 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2023-01-21T15:10:29.197-03:00 INFO 968 --- [ restartedMain] o.s.b.d.Optional.ofSupplier : 
2023-01-21T15:10:29.250-03:00 INFO 968 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : context path ''
2023-01-21T15:10:29.269-03:00 INFO 968 --- [ restartedMain] br.com.projeto.api.ApiApplication : Application running for 37.181)
```

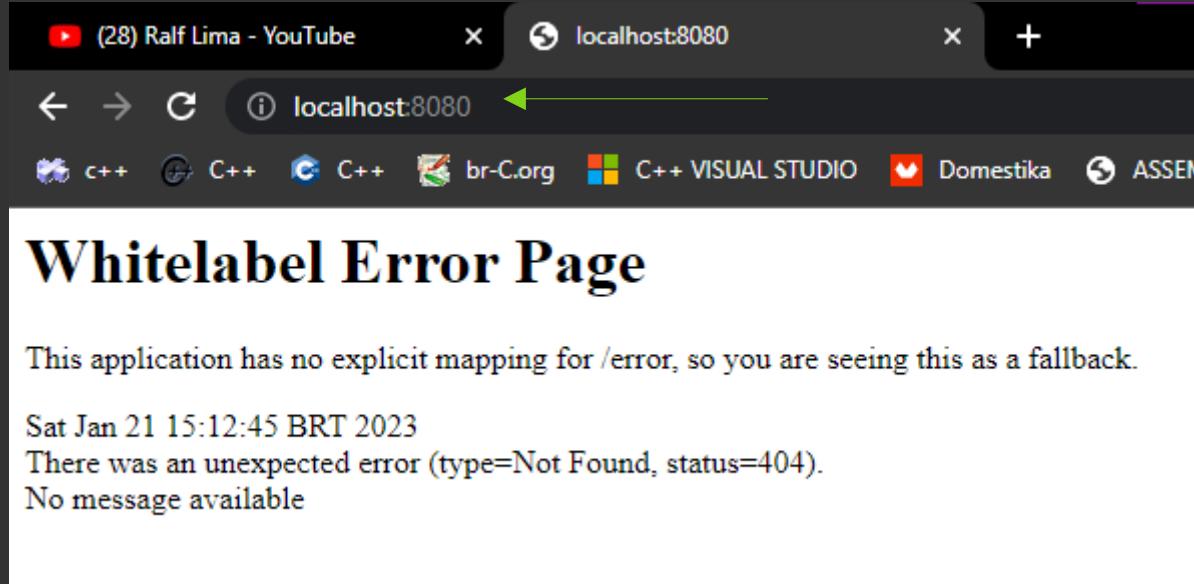
Run: ApiApplication + × ⌂ ⌂ DEBUG CONSOLE

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



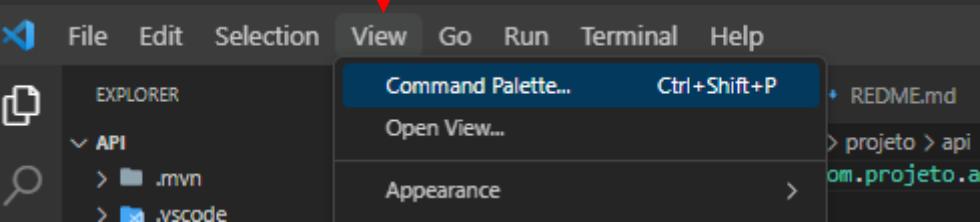
Chuva por parar





ERRO BOM SIGNIFICA QUE ESTA RODANDO

IMPLEMENTANDO O CONTROLLER PARA CORRIGIR O ERRO



ApiApplication.java - api - Visual Studio Code

a X M4 >pref Preferences: Configure Runtime Arguments
ge br.com Preferences: File Icon Theme
t org.spr Preferences: Keymaps
t org.spr Preferences: Language Extensions
ngBootApp Preferences: Open Accessibility Settings
c class A Preferences: Open Default Keyboard Shortcuts (JSON)
un | Debug Preferences: Open Default Settings (JSON)
public sta Preferences: Open Keyboard Shortcuts
Spring 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
Quick Open Previous Editor From History



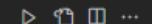
EXPLORER

...

ApiApplication.java

Settings

REDME.md



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

Desabilite essa opção

OUTLINE
TIMELINE
JAVA PROJECTS
MAVEN

Spring Boot-ApiApplication<api> (api)

Search settings

User Workspace

Turn on Settings Sync

Commonly Used

- > 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

Explorer

Auto Reveal

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

true

Auto Reveal Exclude

Configure glob patterns for excluding files and folders from being revealed and selected in the Explorer when they are opened. Read more about glob patterns [here](#).

**/node_modules

**/bower_components

[Add Pattern](#)

Compact Folders

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



Chuva por parar

Go Live

15:22

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

A screenshot of the Visual Studio Code interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "Controle.java - api - Visual Studio Code". The Explorer sidebar on the left lists project files and folders: .mvn, .vscode, src (with main, java, br, com, projeto, api, controle), resources, test, target, .gitignore, HELP.md, mvnw, mvnw.cmd, pom.xml, and README.md. The current file is Controle.java, which contains the following code:

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

A code completion dropdown menu is open at the bottom of the code editor, listing suggestions for 'Controle':

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

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

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

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

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



EXPLORER

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

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

@RestController

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

org.springframework.web.bind.annotation.RestController

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

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

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

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

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

Project Structure:

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

Controle.java - api - Visual Studio Code

Controle.java

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

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

A screenshot of the Visual Studio Code interface. The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "Controle.java - api - Visual Studio Code". The Explorer sidebar on the left shows a project structure under "API": ".mvn", ".vscode", "src" (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 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 (Left):** Shows the project structure:
 - API folder
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - REDME.md
 - Code Editor (Right):** The file Controle.java is 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 }
```

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

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

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

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

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

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

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

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

EXPLORER

API

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

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

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

RESOLVI O ERRO EXECUTANDO POR AQUI RUN

PROBLEMS OUTPUT TERMINAL

TERMINAL

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

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

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

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

Spring Boot-ApiApplication<api> (api)

26°C Nublado



File Edit Selection View Go Run Terminal Help

Controle.java - api - Visual Studio Code



EXPLORER

...

ApiApplication.java

Controle.java X

REDME.md



API

> .mvn

> .vscode

< src

< main

< java

< br

< com

< projeto

< api

< controle

< Controle.java

< ApiApplication.java

> resources

> test

> target

< .gitignore

< HELP.md

< mvnw

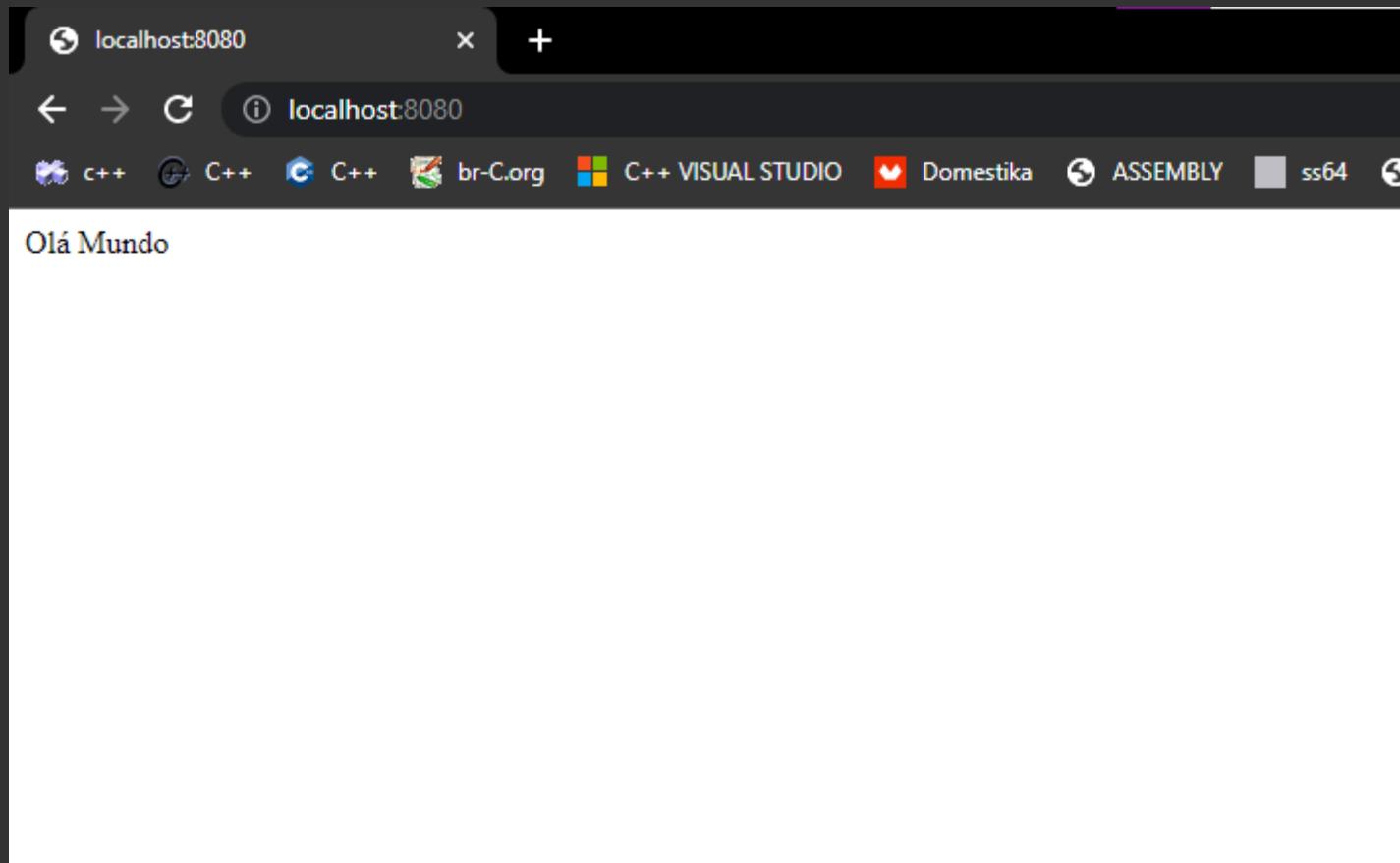
< mvnw.cmd

< pom.xml

< REDME.md

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

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



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

Resolvi adicionando o / na url

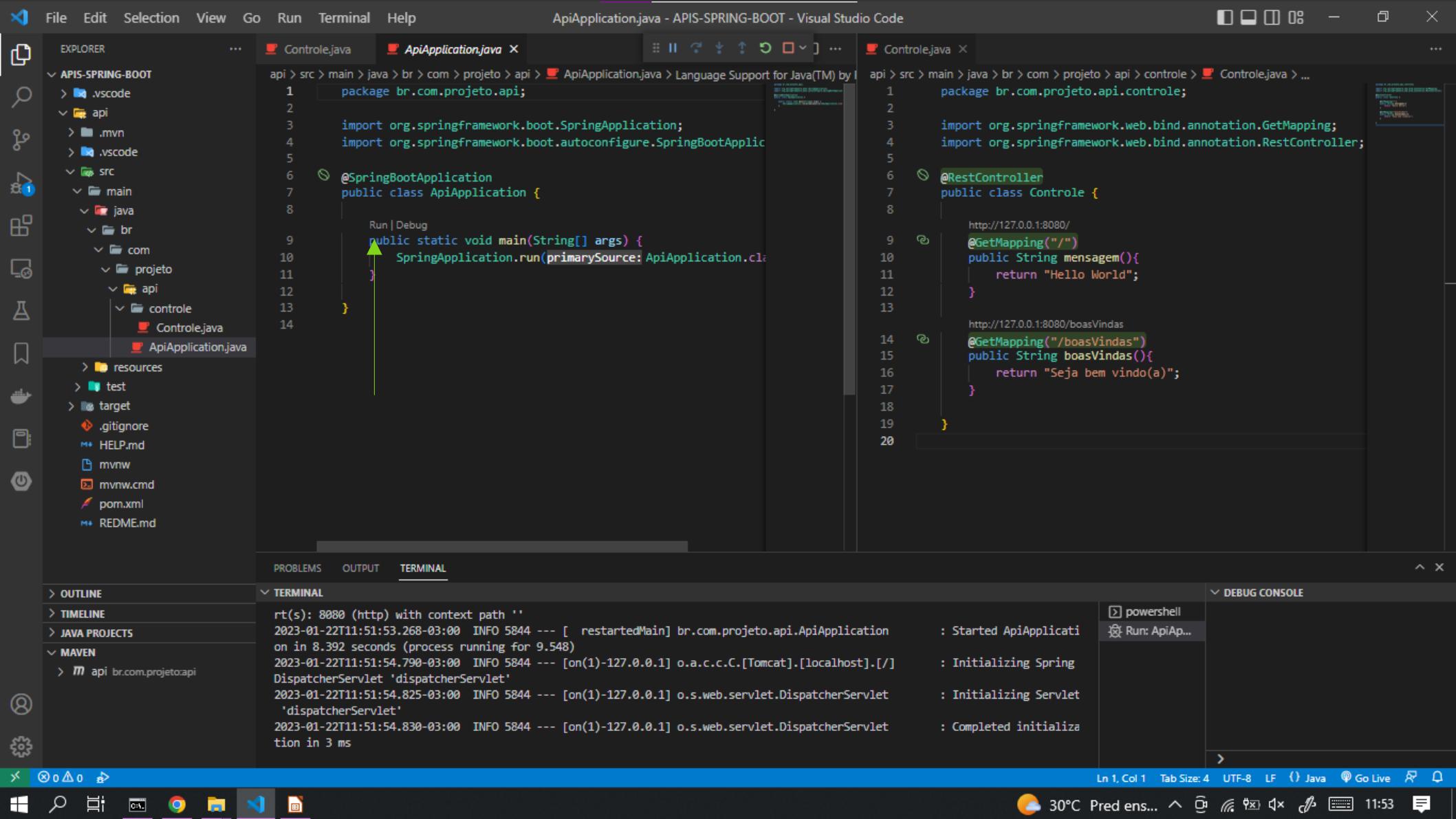
#6 **@PathVariable**

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

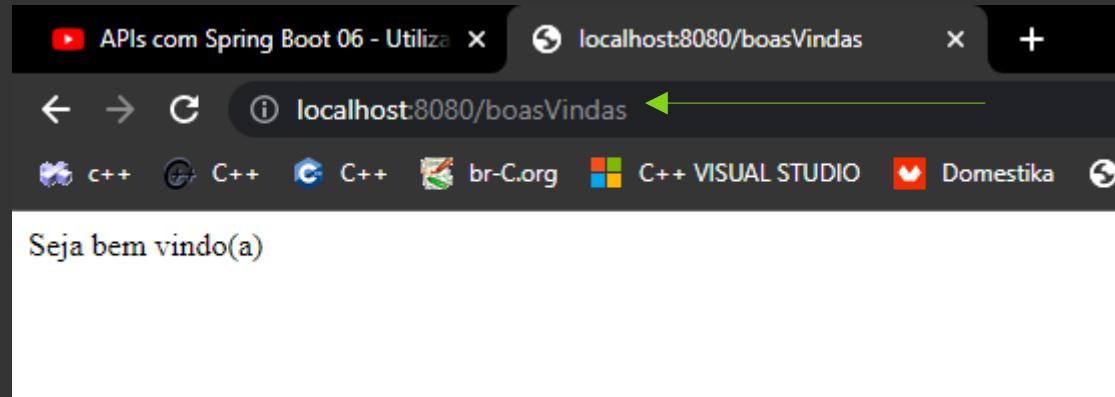
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "src/main/java" folder contains "br.com.projeto.api" which further contains "controle" and "Controle.java". Other files like "ApiApplication.java", ".gitignore", and "pom.xml" are also listed.
- Code Editor:** Displays the "Controle.java"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

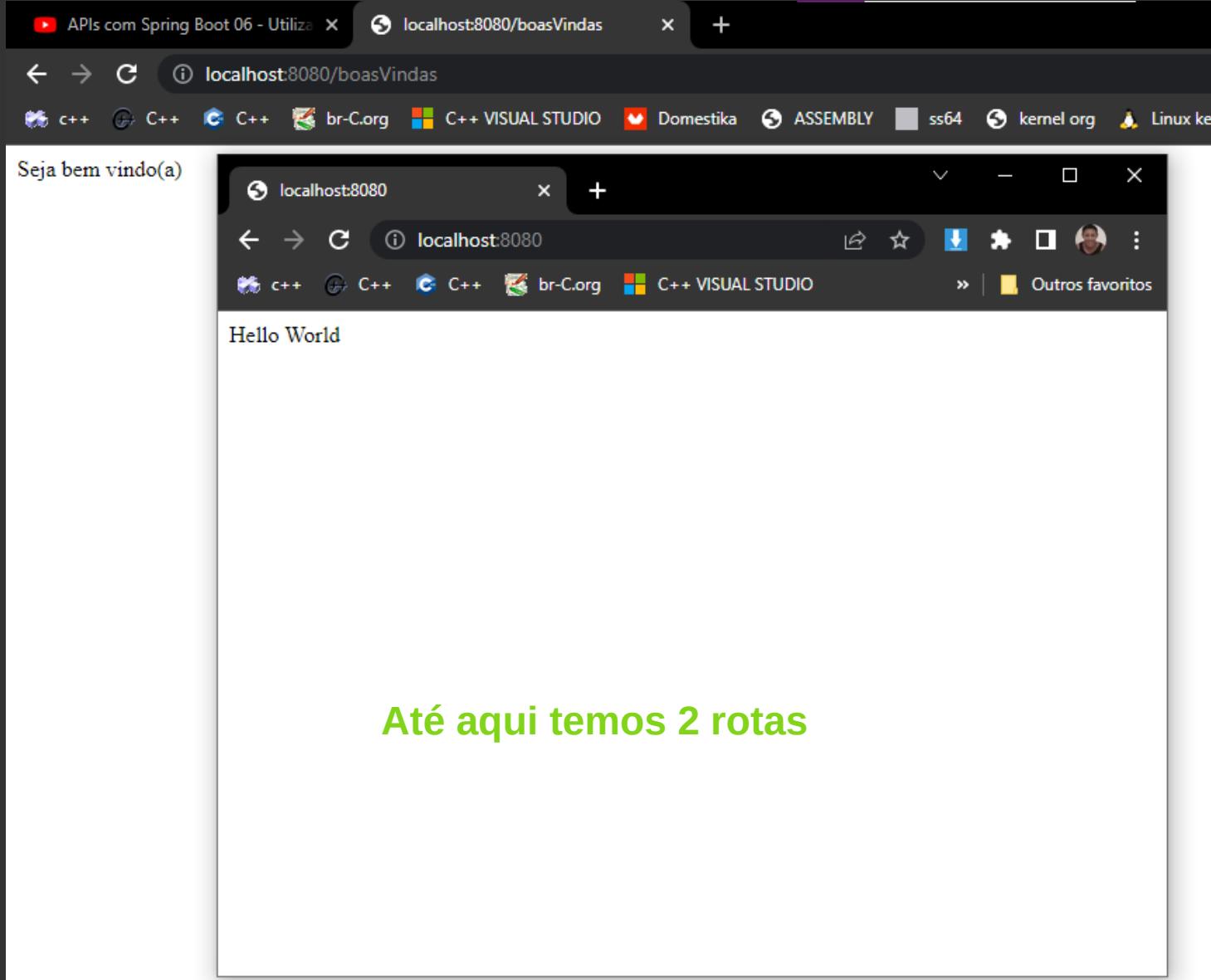
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controle {
8
9     @GetMapping("/")
10    public String mensagem(){
11        return "Hello World";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

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



rota







EXPLORER

APIS-SPRING-BOOT

> .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 Server with a ServerEndpointValue 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 X

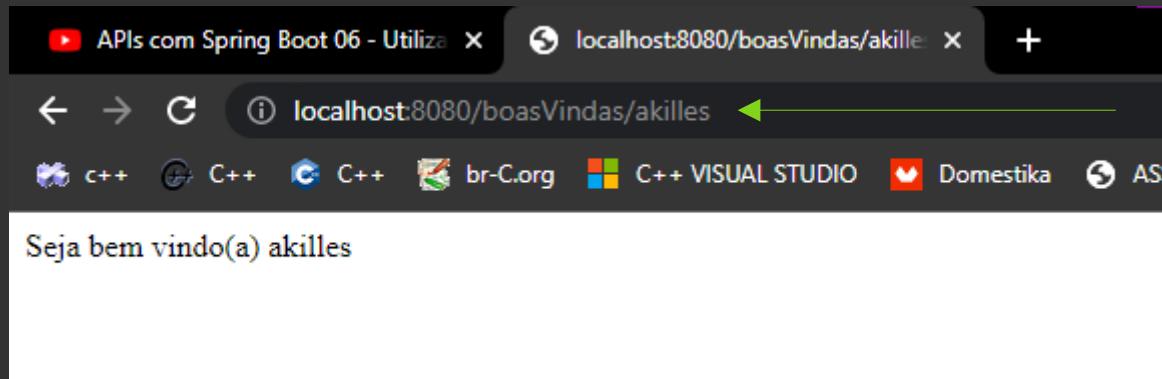
api > src > main > java > br > com > projeto > api > controle > Controle.java > ...

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

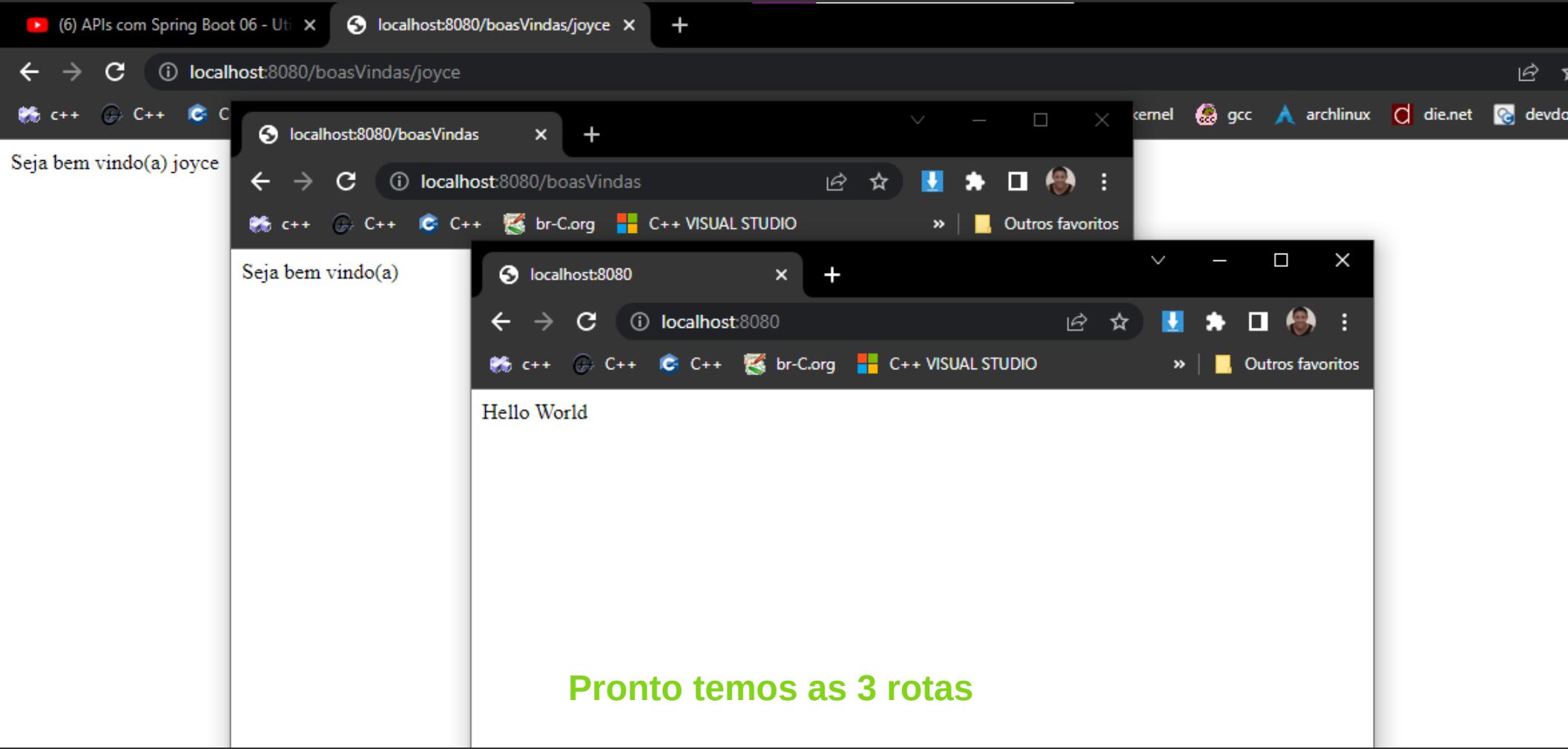
APIS-SPRING-BOOT

- .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

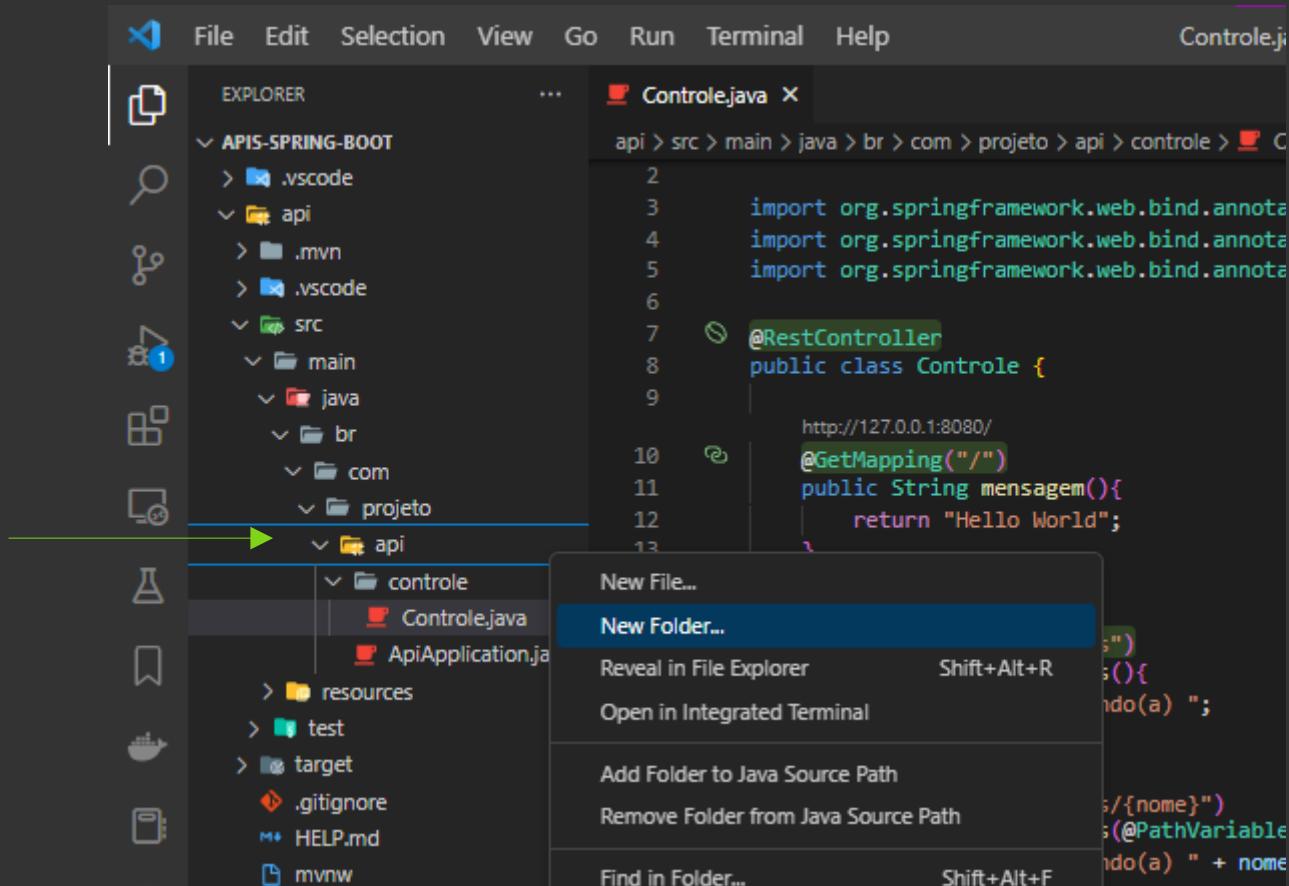
```
2
3     import org.springframework.web.bind.annotation.GetMapping;
4     import org.springframework.web.bind.annotation.PathVariable;
5     import org.springframework.web.bind.annotation.RestController;
6
7     @RestController
8     public class Controle {
9
10        @GetMapping("/")
11        public 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**

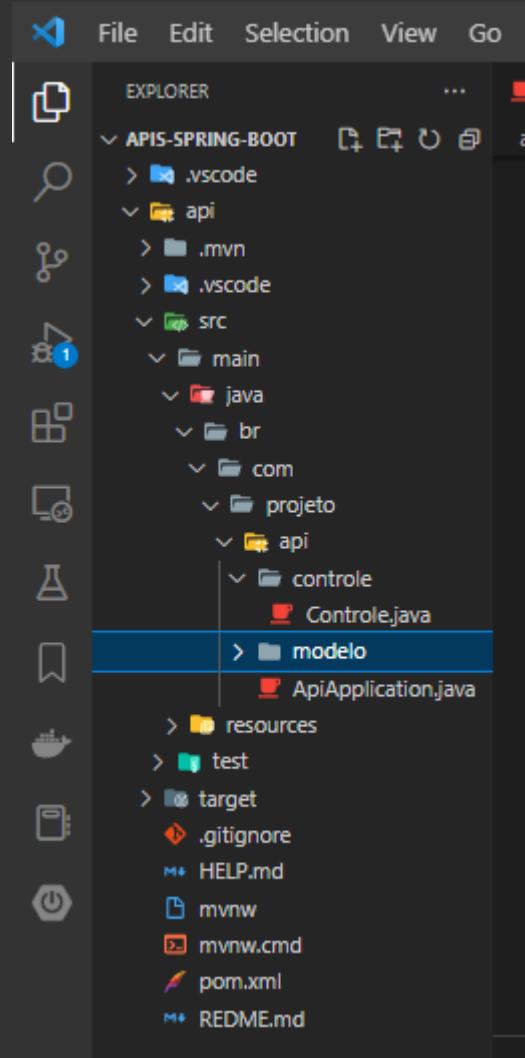


The screenshot shows the VS Code interface with a Java Spring Boot project named "APIS-SPRING-BOOT". The project structure in the Explorer sidebar includes ".vscode", "api", ".mvn", ".vscode", "src", "main", "java", "br", "com", "projeto", and "api". The "api" folder under "src/main/java" is currently selected. In the main editor area, a Java file named "Controle.java" is open, containing the following code:

```
2 import org.springframework.web.bind.annotation.*;
3 import org.springframework.web.bind.annotation.RestController;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7 @RestController
8 public class Controle {
9
10    @RequestMapping("/")
11    @GetMapping("/")
12    public String mensagem(){
13        return "Hello World";
14    }
15}
```

A context menu is open over the "api" folder in the Explorer sidebar, with the "New Folder..." option highlighted.

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.RequestMapping;
6
7     @RestController
8     public class Controle {
9
10        @RequestMapping("/")
11        public String mensagem(){
12            return "Hello World";
13        }
14
15        @RequestMapping("/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



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:** Includes icons for file operations like Open, Save, Find, and Refresh.

Inicialmente crie esses 2 atributos nome e idade

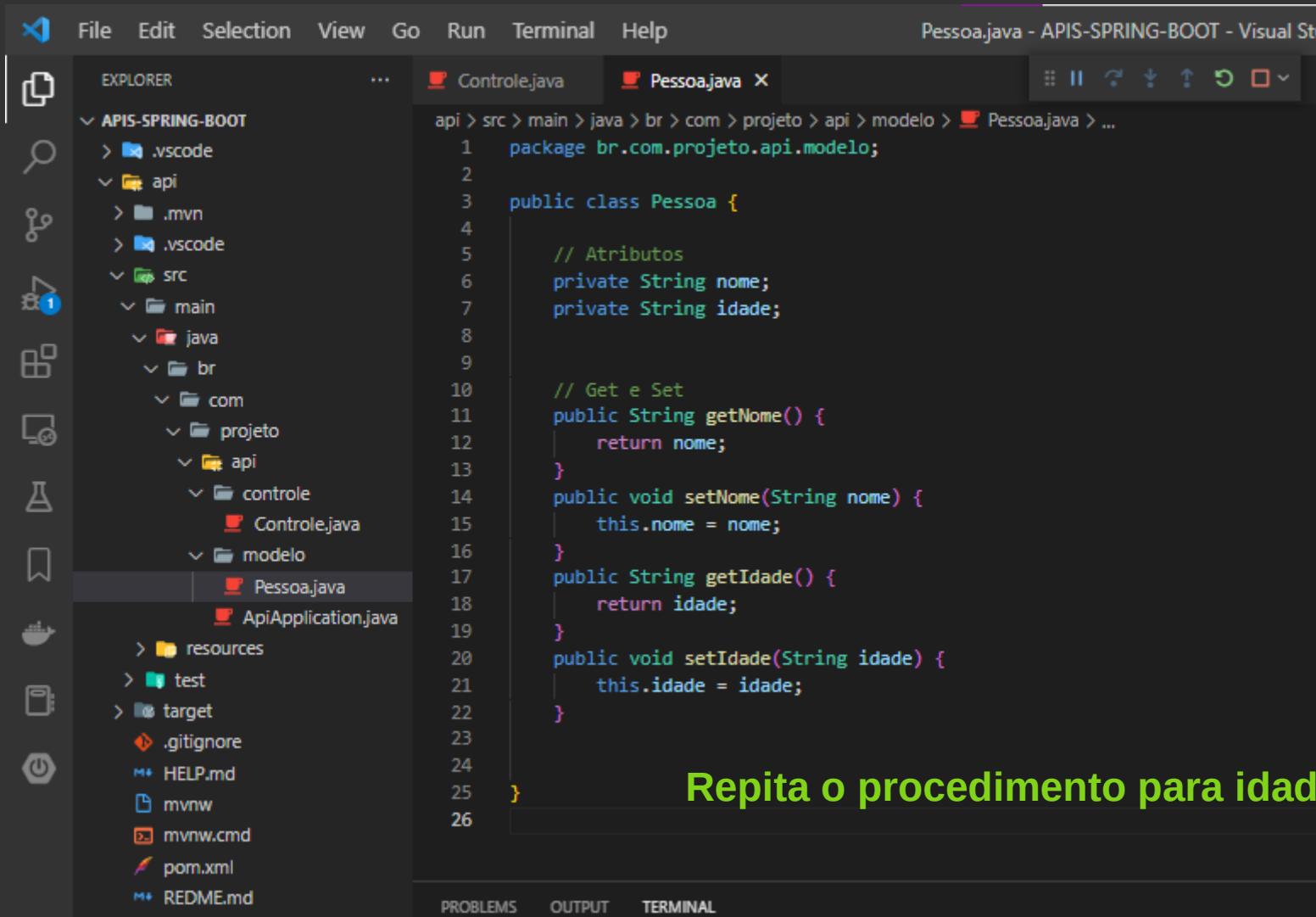
**Lembrando que uma classe em java serve de modelo para construir objetos
atravez do seu construtor**

**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' which has 'java', 'br', 'com', 'projeto', 'api', 'controle', and 'modelo' subfolders. 'Controle.java' and 'Pessoa.java' are listed under 'controle'. 'ApiApplication.java' is listed under 'modelo'.
- Code Editor:** The file 'Pessoa.java' is open. A tooltip is displayed over the line 'private String nome;'. The tooltip includes:
 - Quick Fix...
 - Remove 'nome', keep assignments with side effects
 - More Actions...A dropdown menu is open, listing:
 - Generate Getter and Setter for 'nome'
 - Generate Getter for 'nome'
 - Generate Setter for 'nome'
 - Generate Constructors...
 - Add final modifier for 'nome'The first option, 'Generate Getter and Setter for 'nome'', is highlighted in blue.
- Bottom Status Bar:** Shows 'PROBLEMS 2', 'OUTPUT', and 'TERMINAL'.

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



Repita o procedimento para idade

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

File Edit Selection View Go Run Terminal Help

Pessoajava - APIS-SPRING-BOOT - Visual S

EXPLORER

APIS-SPRING-BOOT

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

Controle.java

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

Pessoajava X

PROBLEMS OUTPUT TERMINAL

Colocados na ordem manualmente

#8

Vinculando modelos e controles

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

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

EXPLORER

APIS-SPRING-BOOT

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

Controle.java 3 Pessoajava

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

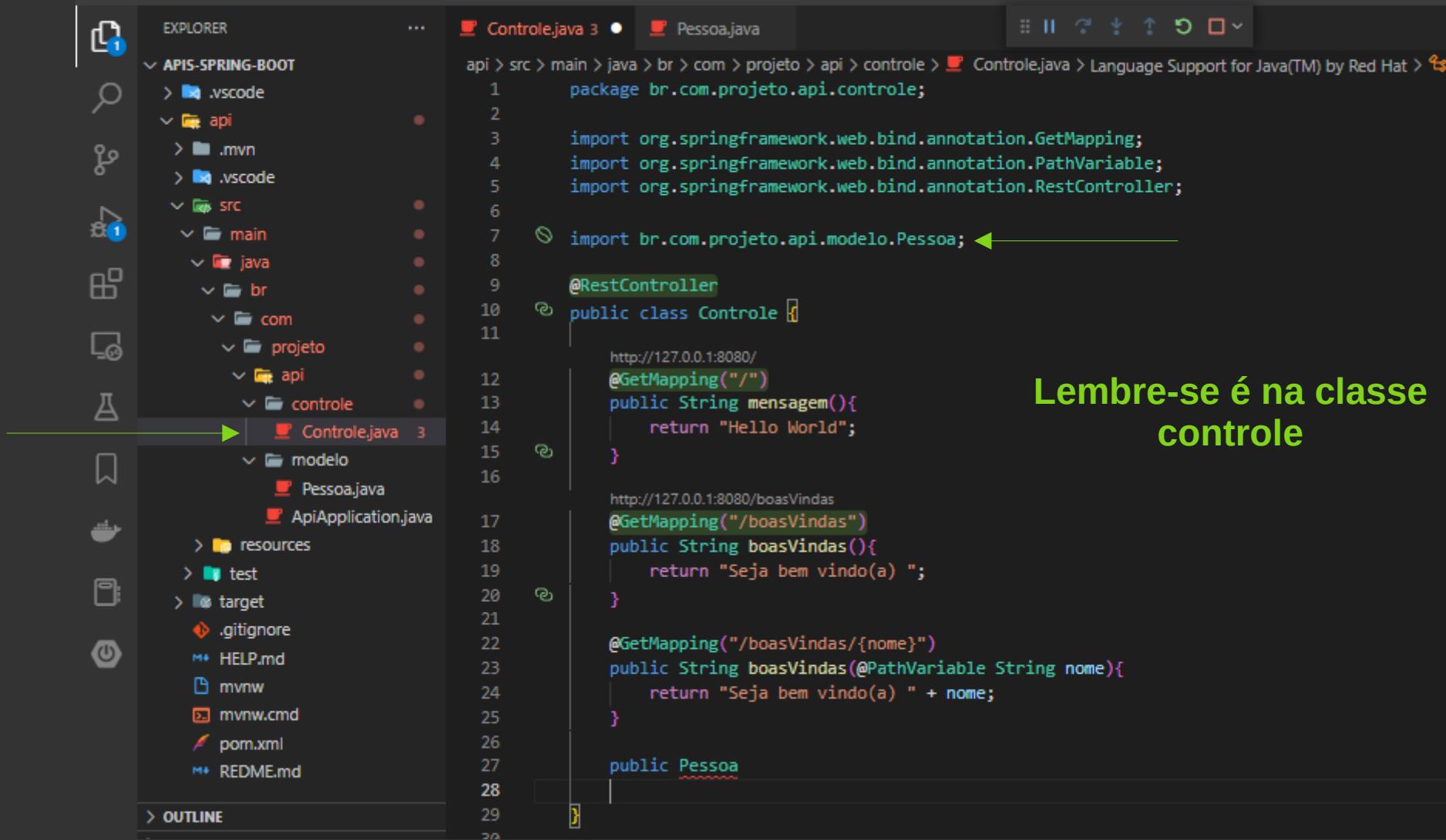
PROBLEMS 3 OUTPUT TERMINAL

Crie a rota e importe Pessoa e adicione seu importe

br.com.projeto.api.modelo.Pessoa

Pessoajava

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



Lembre-se é na classe controle



File Edit Selection View Go Run Terminal Help

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



EXPLORER

APIS-SPRING-BOOT

.vscode

api

.mvn

.vscode

src

main

java

br

com

projeto

api

controle

Controle.java 1

modelo

Pessoa.java

ApiApplication.java

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

README.md

Controle.java 1 X

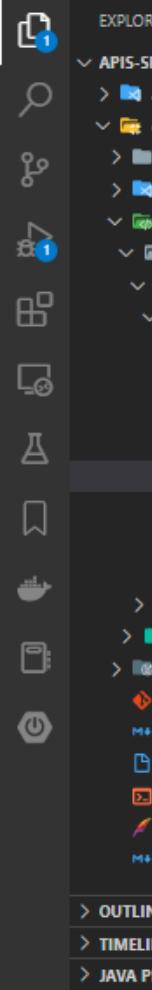
Pessoa.java

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

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

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

O nome da rota é pessoa



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 x
n.RequestAttribute

Annotation to bind a method parameter to a request attribute.

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

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

EXPLORER

APIS-SPRING-BOOT

- > .vscode
- < api
- > .mvn
- > .vscode
- < src
- < main
- < java
- < br
- < com
- < projeto
- < api
- < controle
- Controle.java 3
- < 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 }
```

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.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT". The "src/main/java" folder contains "br.com.projeto.api.control" which has "Controle.java" selected. Other files include "Pessoa.java" and "ApiApplication.java".
- Code Editor:** Displays the "Controle.java" file content:

```
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    }
```
- Annotations:** Three green arrows point from the text "Pessoa é o tipo de dado que espero" to the word "Pessoa" in the code editor.
- Text Annotations:**
 - "Pessoa é o tipo de dado que espero" is located in the upper right area.
 - "Nome desse objeto do tipo Pessoa p" is located in the middle right area.
 - "O que retornamos" is located at the bottom right area.

**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    }
```

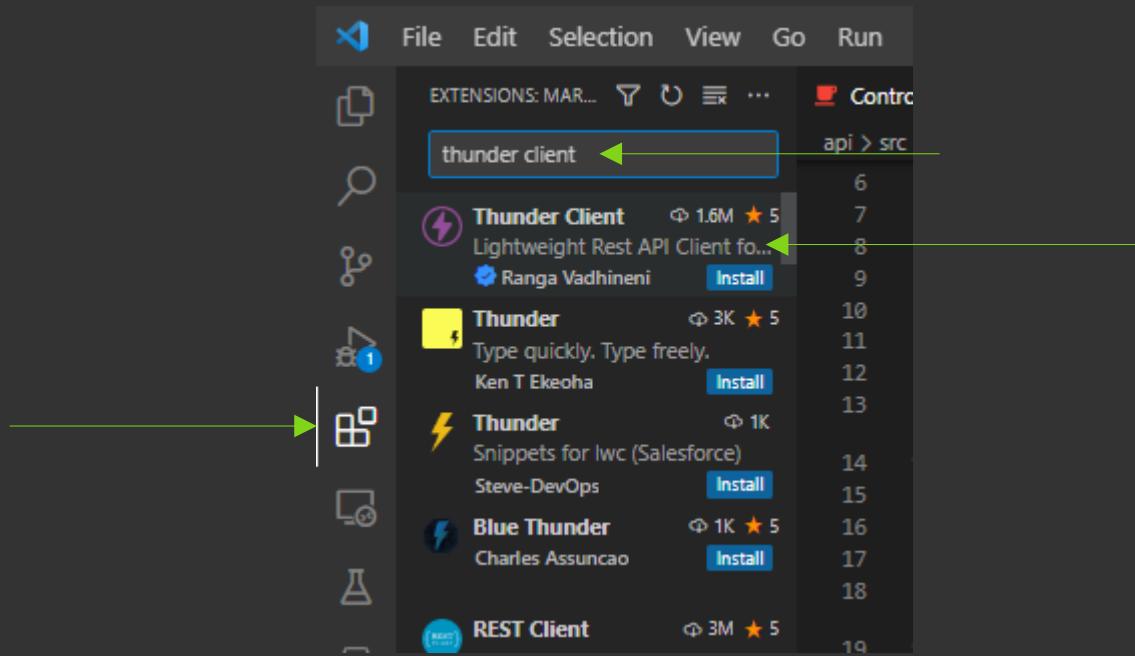
PROBLEMS OUTPUT TERMINAL

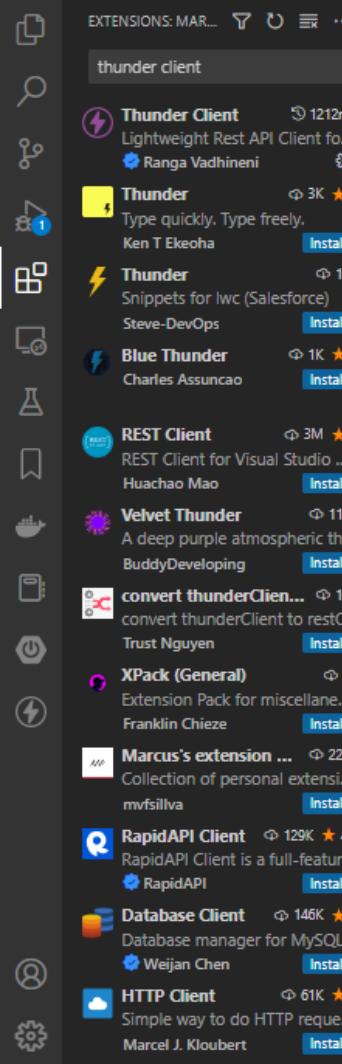
Por fim criamos a rota /pessoa

#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





Extension: Thunder Client X Pessoajava

Thunder Client v2.3.4

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

Lightweight Rest API Client for VS Code

Disable Uninstall ⚙️

This extension is enabled globally.

Pronto instalamos

Details Feature Contributions Changelog Runtime Status

Thunder Client

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

- Voted as #10 Product of the day on Product Hunt
- Website - www.thunderclient.com
- Follow Twitter for updates - twitter.com/thunder_client
- Support: github.com/rangav/thunder-client-support

Story behind Thunder Client

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

Usage

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

TERMINAL

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

Categories

Programming Languages Snippets Testing

Extension Resources

Marketplace Repository License Ranga Vadhineni

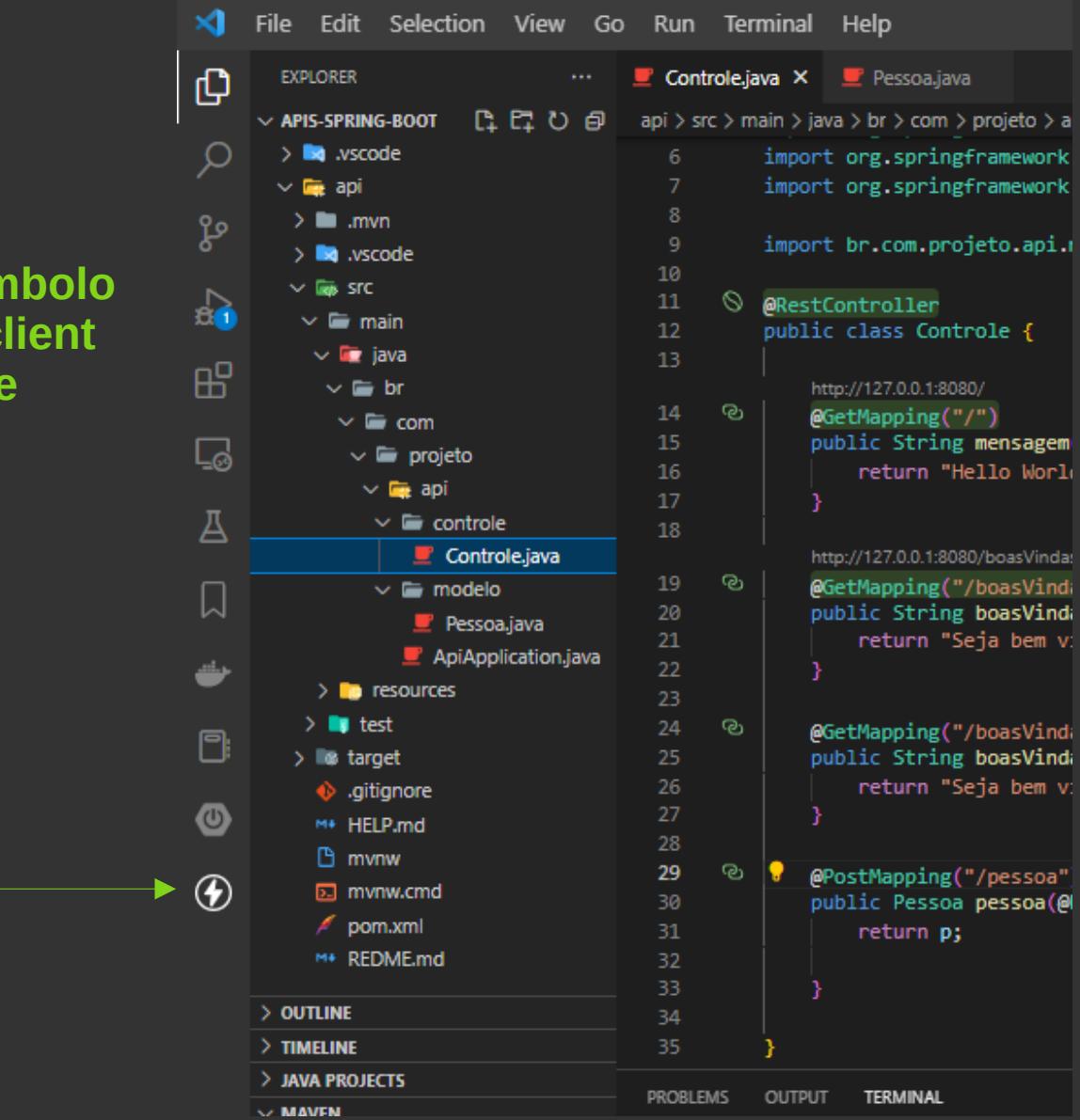
More Info

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

DEBUG CONSOLE

- powershell
- Run: ApiAp...

Repare no simbolo
do thunder client
Click nele

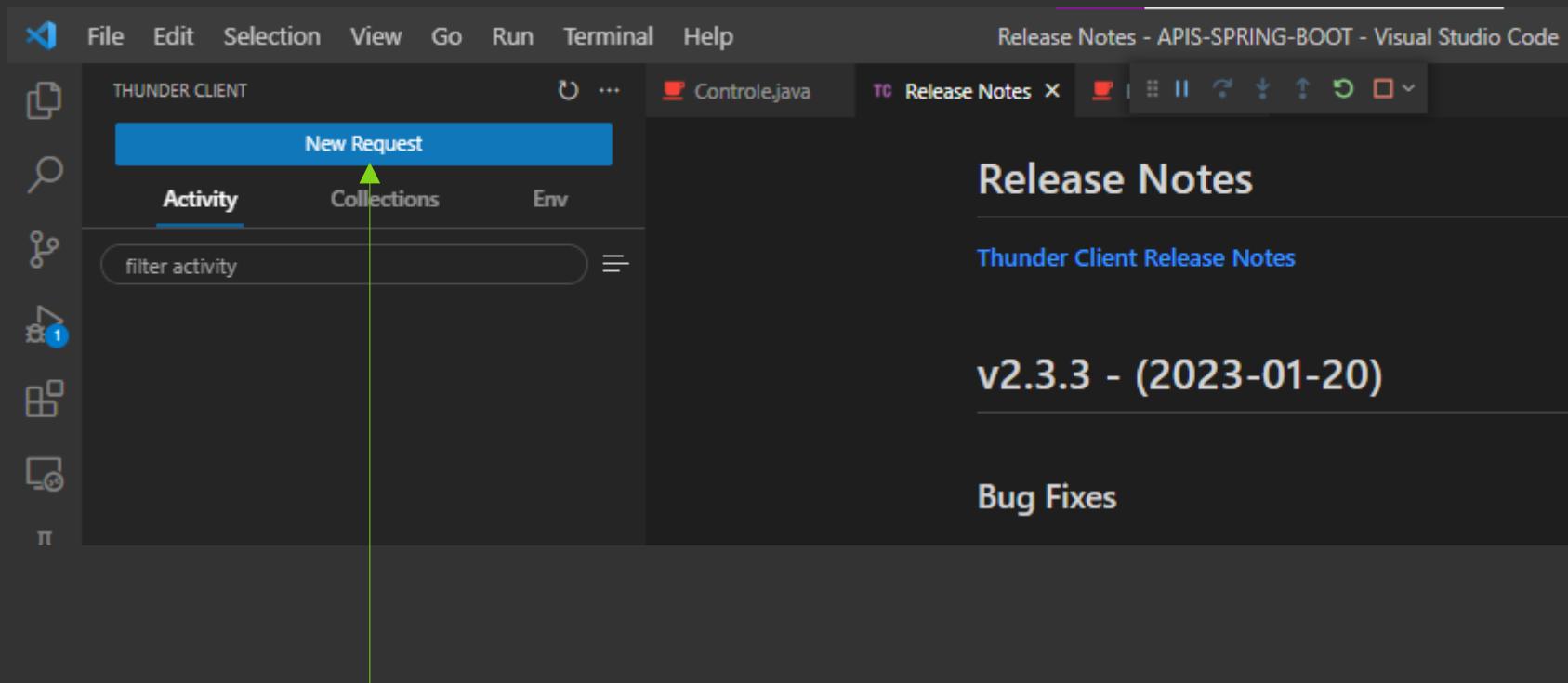


The screenshot shows the Visual Studio Code interface with the following details:

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

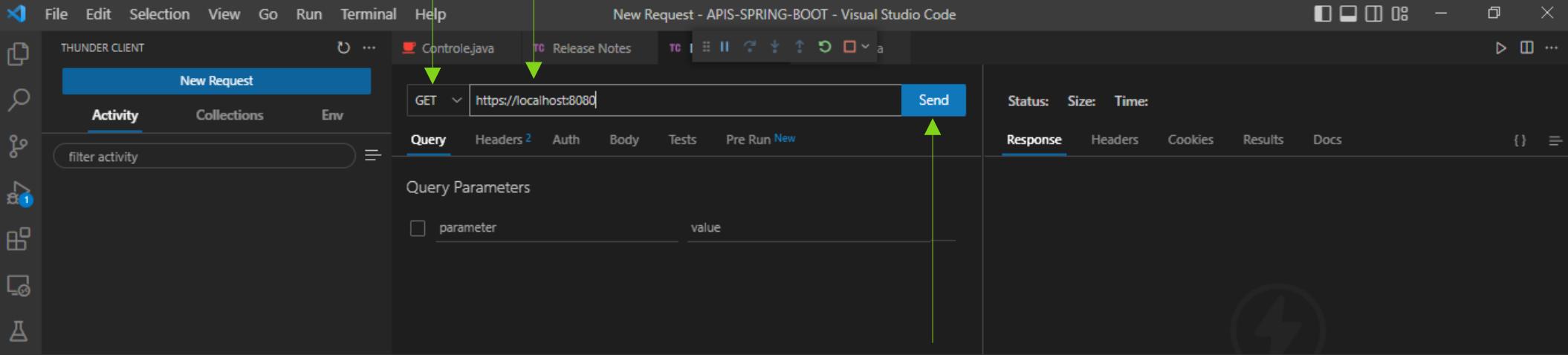
TERMINAL

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

DEBUG CONSOLE

powershell Run: ApiAp...

31°C Parc ens... 13:09

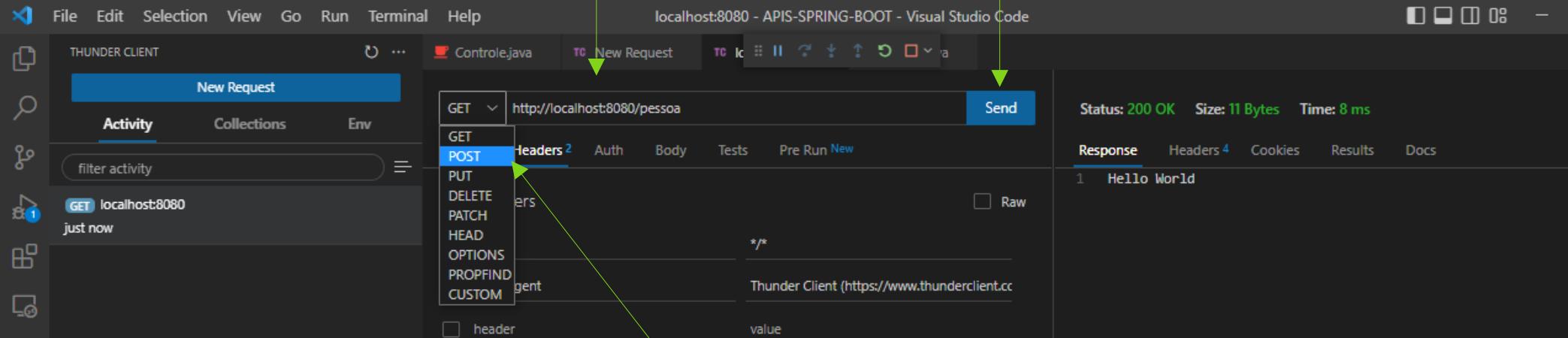


Fazemos um teste na rota hello world digite a url com o metodo get e click em send

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

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

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



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

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

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080 just now

Controle.java New Request

POST http://localhost:8080/pessoa Send

Headers 2 Query Auth Body Tests Pre Run New

Http Headers Raw

Accept */*

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

header value

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

Response Headers 4 Cookies Results Docs

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

Deu esse erro pois essa rota espera uma informação via body

PROBLEMS OUTPUT TERMINAL

TERMINAL

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

DEBUG CONSOLE

powershell Run: ApiApp...

31°C Parc ens... 13:16

Click em body

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

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

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

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

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

No body crie seu objeto json
E depois click em send

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080 just now

ApiApplication.java localhost:8080

POST http://localhost:8080/pessoa Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

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 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
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].[/]
dispatcherServlet '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
[]
```

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

powershell
Run: ApiAp...

#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...
nespinozacr [Install](#)
- MySQL Statement Scr...** v41K ★ 5
Easy mysql statement running with s...
Jared Black [Install](#)
- ES7 JavaScript/Node/...** v38K ★ 5
Simple extension for Node, javascript...
abrahamwilliam007 [Install](#)
- SQLite & MySQL Snipp...** v27K ★ 5
A snippet for MySQL and SQLite, to ...
Rohit Chouhan [Install](#)
- MySQL Tool** v2K ★ 4.5
MySQL management tool is a unifie...
wycliffe Pepela [Install](#)
- MySQL Inline Decorator...** v1K ★ 5
Ilyas Akin [Install](#)
- Go MySQL Highlighter** v5K ★ 5
Syntax highlights MySQL strings in G...
Stirling Marketing Group [Install](#)

Extension: MySQL X



MySQL v6.1.3

Weijan Chen | 767,140 | ★★★★★(191)

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

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

This extension is enabled globally.

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

Database Client

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

The online document is migrate to doc.database-client.com.

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

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

PROBLEMS OUTPUT TERMINAL

TERMINAL

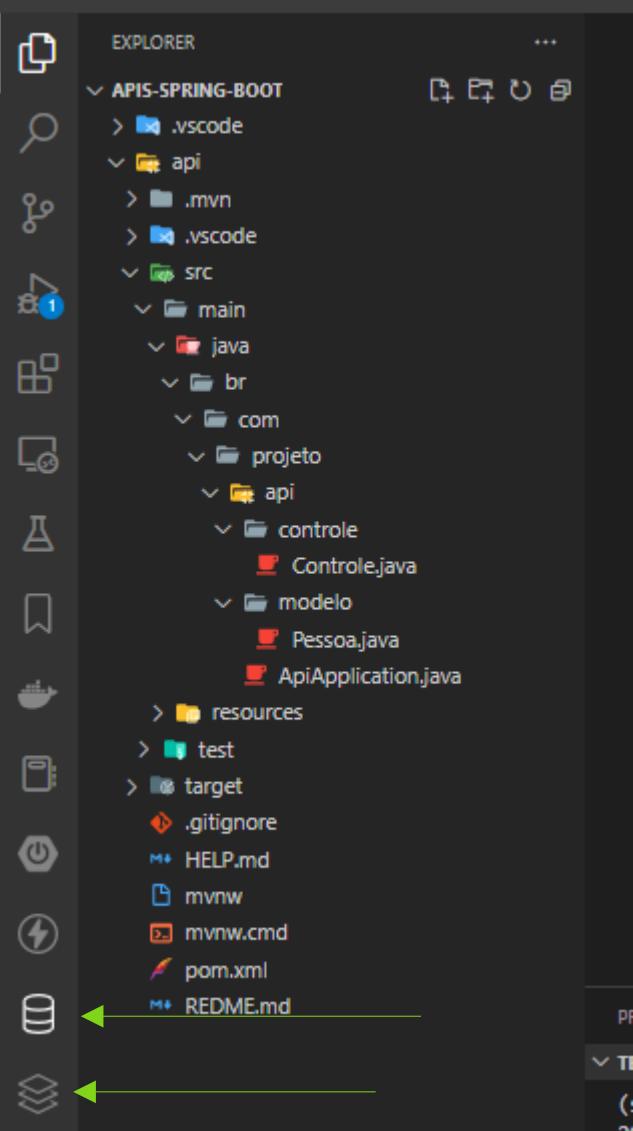
```
(s): 8080 (http) with context path ""
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [ restartedMain] br.com.projeto.api.ApiApplication : Started ApiApp...
2023-01-22T13:41:41.901-03:00 INFO 10316 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Sp...
2023-01-22T13:41:41.901-03:00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

DEBUG CONSOLE

powershell

Run: ApiAp...

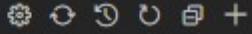
File Edit Selection View Go Run Terminal



Ela cria 2 atalhos click no database



DATABASE



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



Create Connection

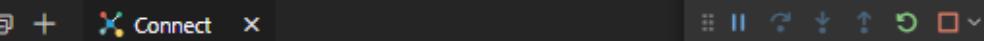


PROBLE

TERMIN

(s): 8

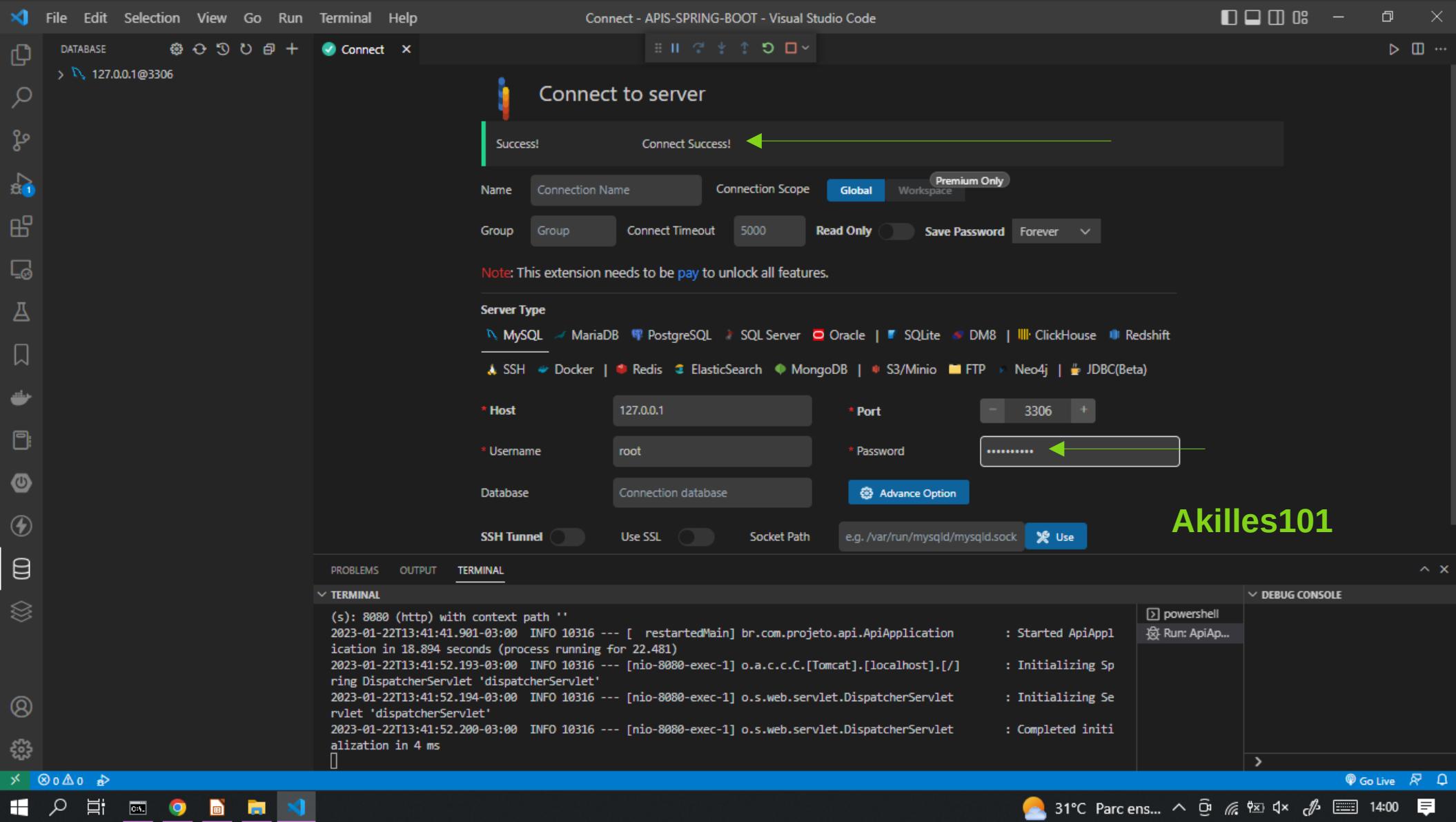




Deixei os dados como
estão só mudei a senha
Depois click em connect

Akilles101

A senha é a verdadeira do mysql



Aqui estão as bases de dados do mysql

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

DATABASE 127.0.0.1@3306 + -

Connect to server

Success! Connect Success!

Name Connection Name Connection Scope Global Premium Only

Group Group Connect Timeout 5000 Read Only Save Password Forever

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

Server Type

MySQL MariaDB PostgreSQL SQL Server Oracle SQLite DM8 ClickHouse Redshift

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

* Host 127.0.0.1 * Port 3306

* Username root * Password

Database Connection database Advance Option

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

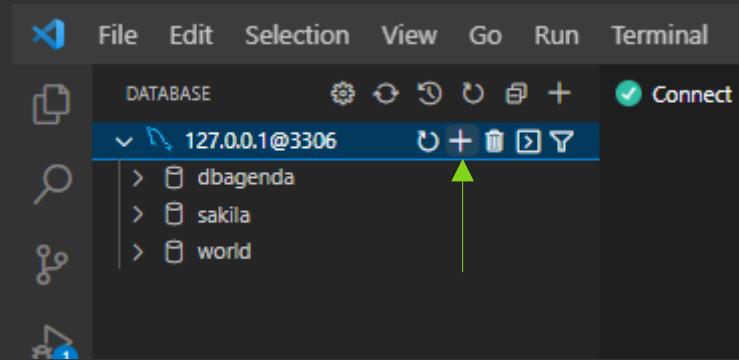
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

TERMINAL

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

powershell Run: ApiAp...

31°C Parc ens... Go Live 14:01



Click no sinal de + para criar o banco de dados

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

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

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

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

Depois de dar nome api_spring click em execute

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

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

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

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

Data

CREATE DATABASE api_spring
DEFAULT CHARACTER SET = 'utf8mb4'

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

CREATE DATABASE api_spring DEFAULT CHARACTER SET = 'utf8mb4'

AffectedRows : 1

Veja que o banco de dados foi criado

PROBLEMS OUTPUT TERMINAL

TERMINAL

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

DEBUG CONSOLE

powershell
Run: ApiAp...

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

31°C Parc ens... 14:05

#11
IMPLEMENTANDO DEPENDENCIAS

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

EXPLORER

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

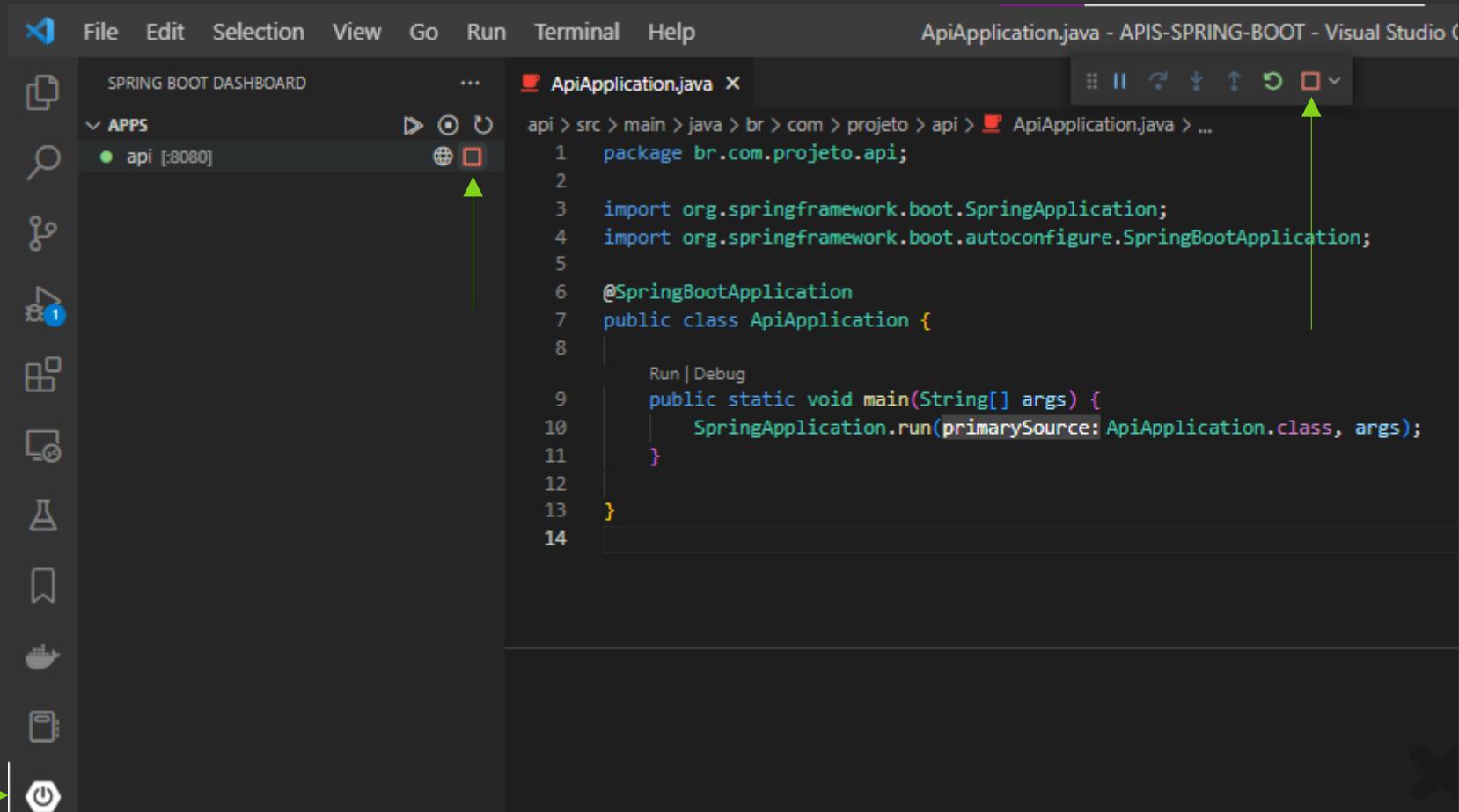
ApiApplication.java X

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

PROBLEMS OUTPUT TERMINAL

TERMINAL

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



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

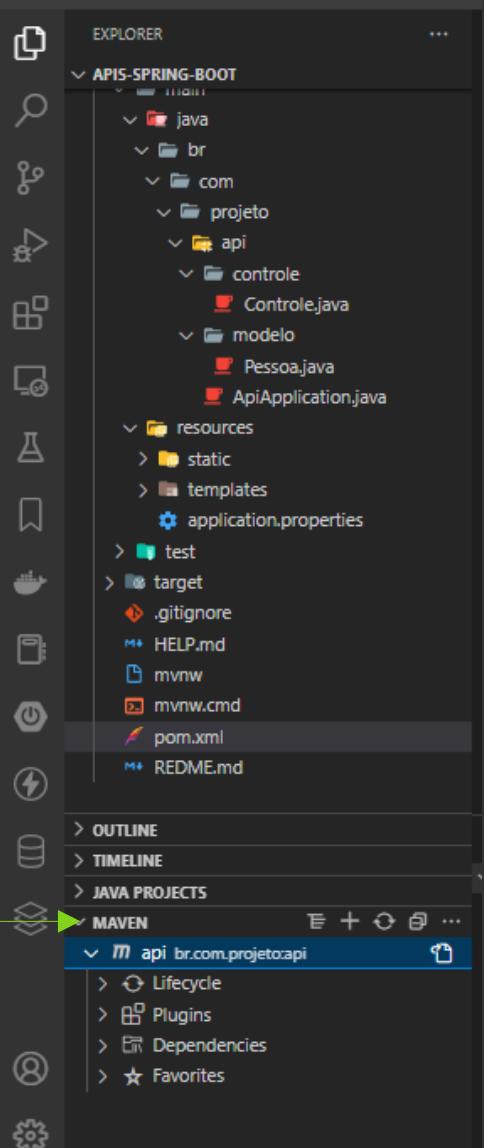
The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** pom.xml - APIS-SPRING-BOOT - Visual Studio Code.
- Left Sidebar (Explorer):** Shows the project structure under "APIS-SPRING-BOOT".
 - Root: .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
- Middle Area (Code Editor):** The "pom.xml" file is open, displaying its XML content. The code includes dependencies for Spring Boot Starter Actuator and Spring Boot itself.
- Right Area (Preview):** A preview pane on the right shows the generated Java code for the API.
- Bottom Navigation:** PROBLEMS, OUTPUT, TERMINAL (selected), DEBUG CONSOLE.
- Bottom Status Bar:** port(s): 8080 (http) with context path '' 2023-01-23T11:23:14.611-03:00 INFO 6540 --- [restartedMain] br.com.projeto.api.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].[/] g DispatcherServlet 'dispatcherServlet' 2023-01-23T11:23:16.063-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet et 'dispatcherServlet' 2023-01-23T11:23:16.087-03:00 INFO 6540 --- [on(2)-127.0.0.1] o.s.web.servlet.DispatcherServlet zation in 2 ms PS E:\APIS-SPRING-BOOT>
- Bottom Icons:** Standard VS Code icons for file operations.

Text Overlay: Todas as dependencias ficam no arquivo pom.xml

ADICIONANDO AS DEPENDÊNCIAS

File Edit Selection View Go Run

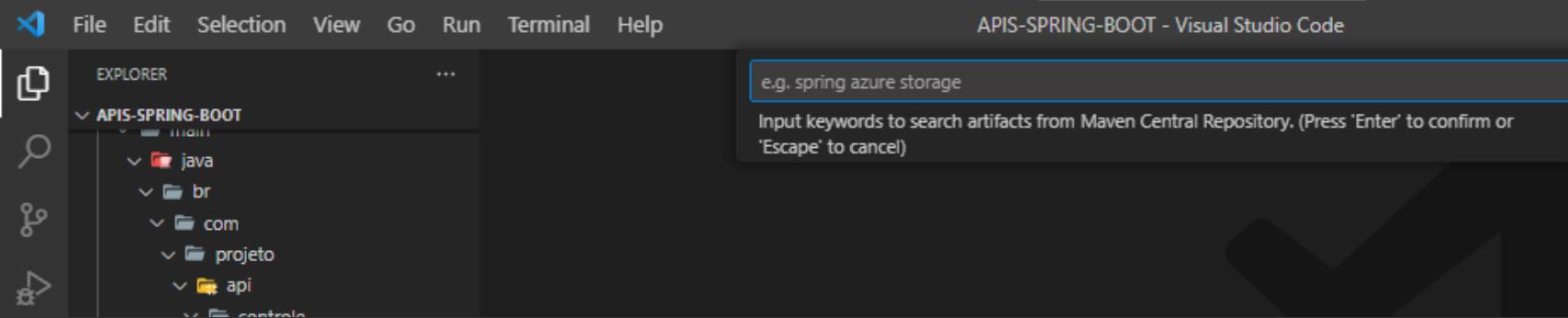


O QUE É O MAVEN?

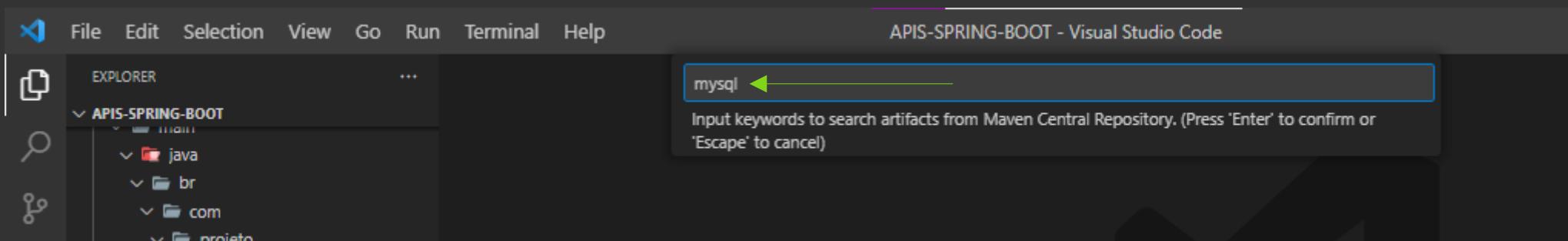
Responsavel por gerenciar os pacotes do spring

Click no sinal de +

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

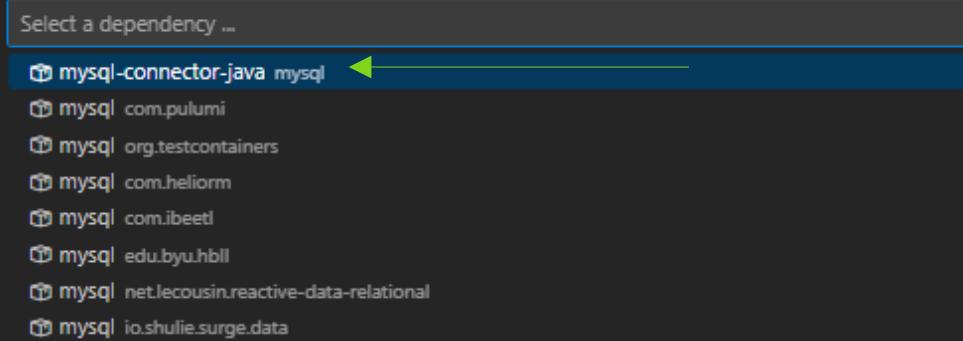
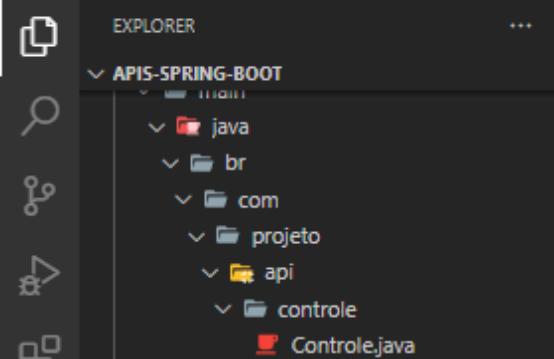


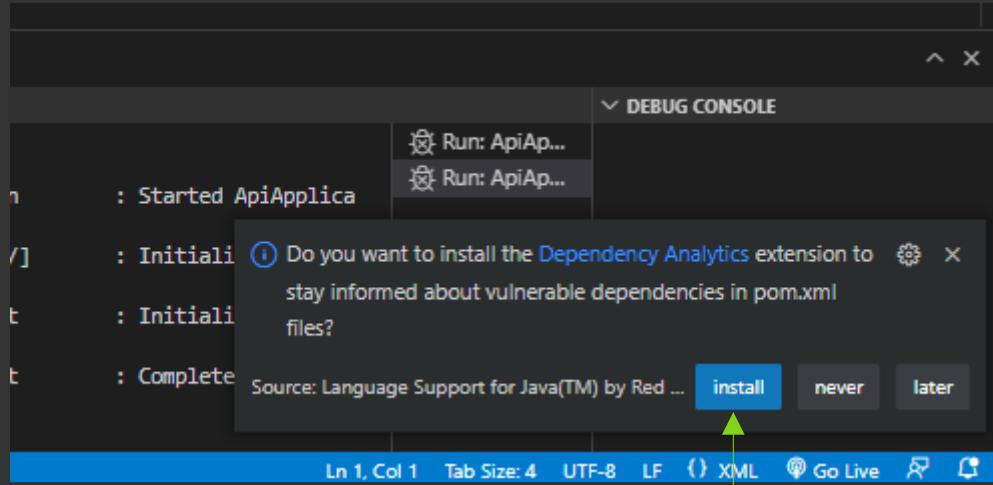
Digite mysql e tecle enter



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

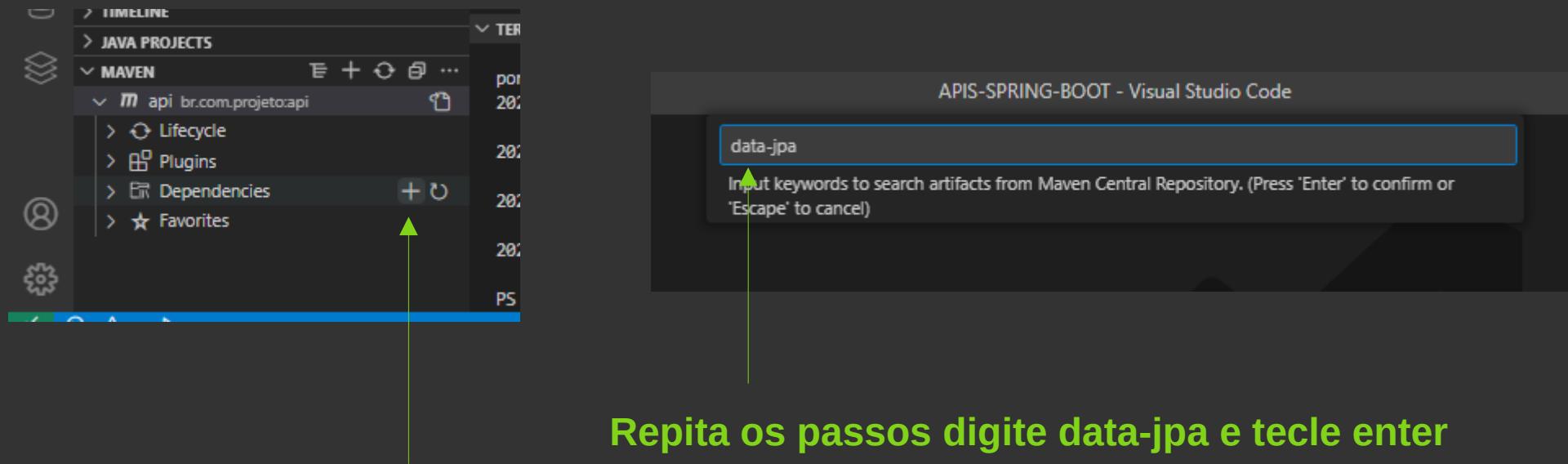
```
port(s): 8080 (http) with context path ''
2023-01-23T11:23:14.611-03:00  INFO 6540 --- [ restartedMain] br.com.projeto.api.ApiApplication
: Started ApiApplication in 11.71 seconds (process running for 14.239)
2023-01-23T11:23:15.963-03:00  INFO 6540 --- [on(2)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/]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

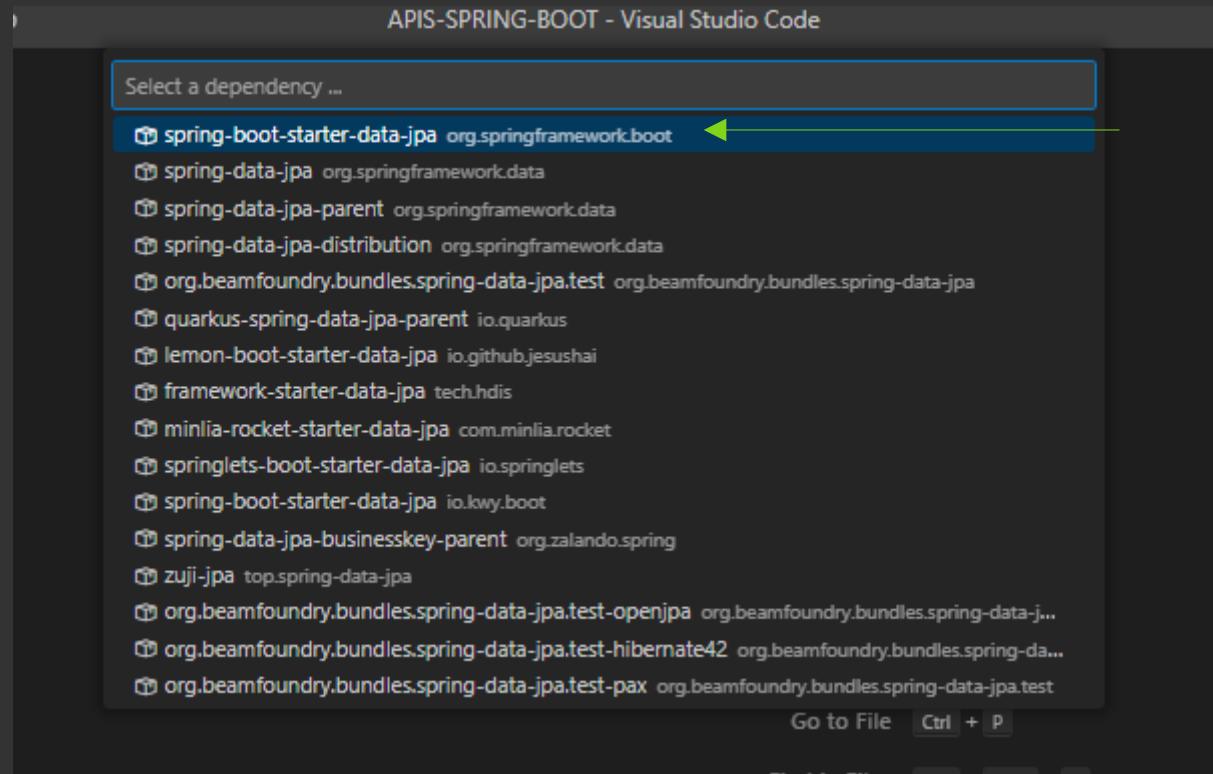
Run: ApiAp...

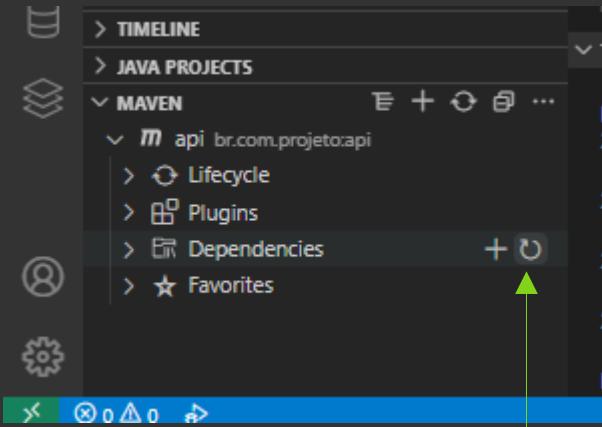
DEBUG CONSOLE

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



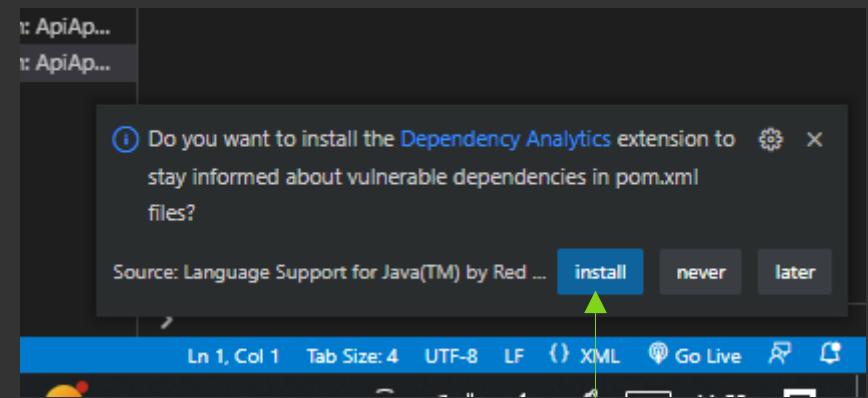
Repita os passos digite **data-jpa** e tecle enter





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>
```

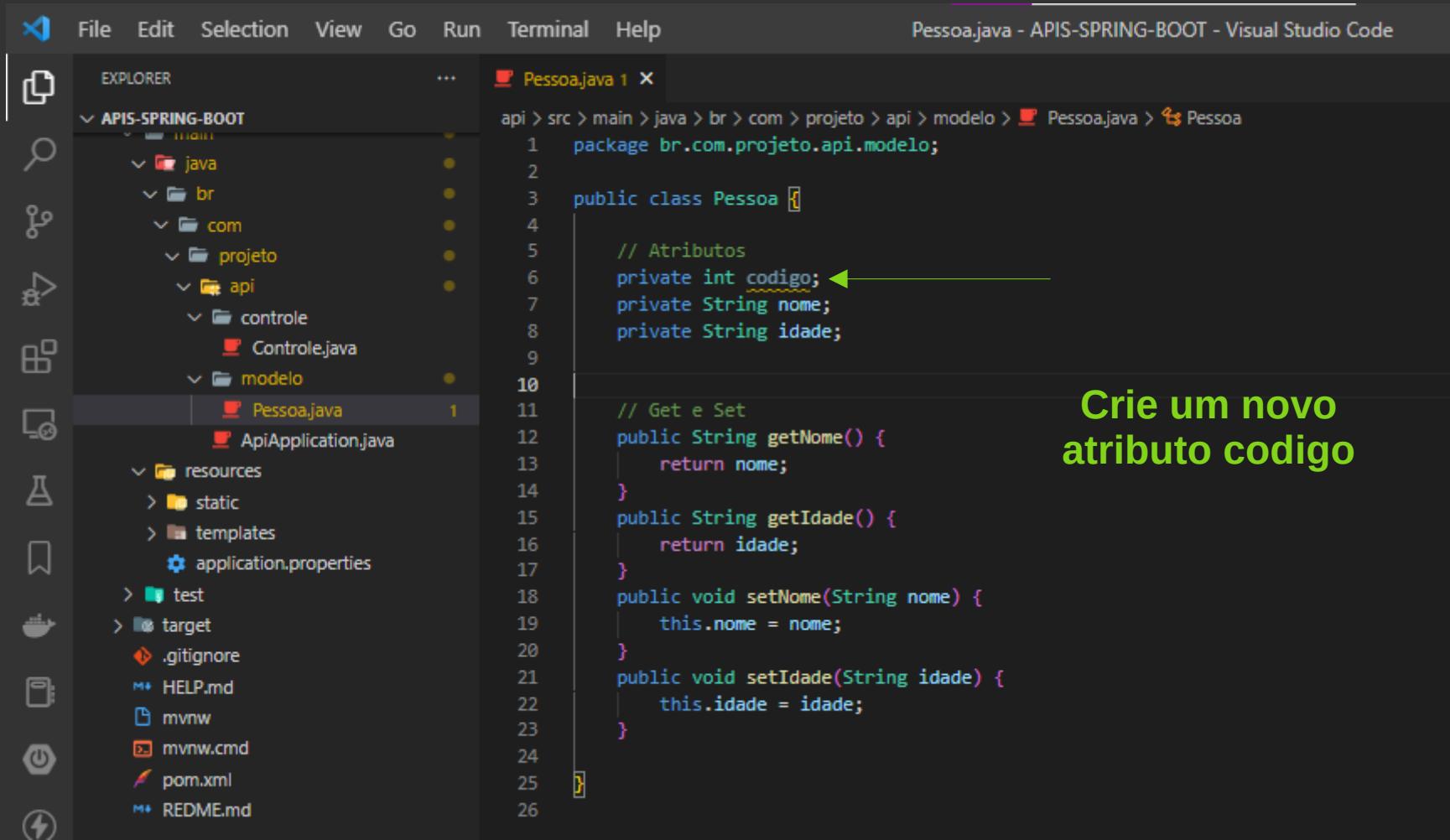
Adicionando assim temos ela instalada

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' (selected)
 - Generate Getter for 'codigo'
 - Generate Setter for 'codigo'
 - Generate Constructors...
 - Add final modifier for 'codigo'

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

Run Terminal Help

Pessoa.java - APIS-SPR

...

Pessoa.java X

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

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

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

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

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

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

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

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

Adicione a anotation e seu import

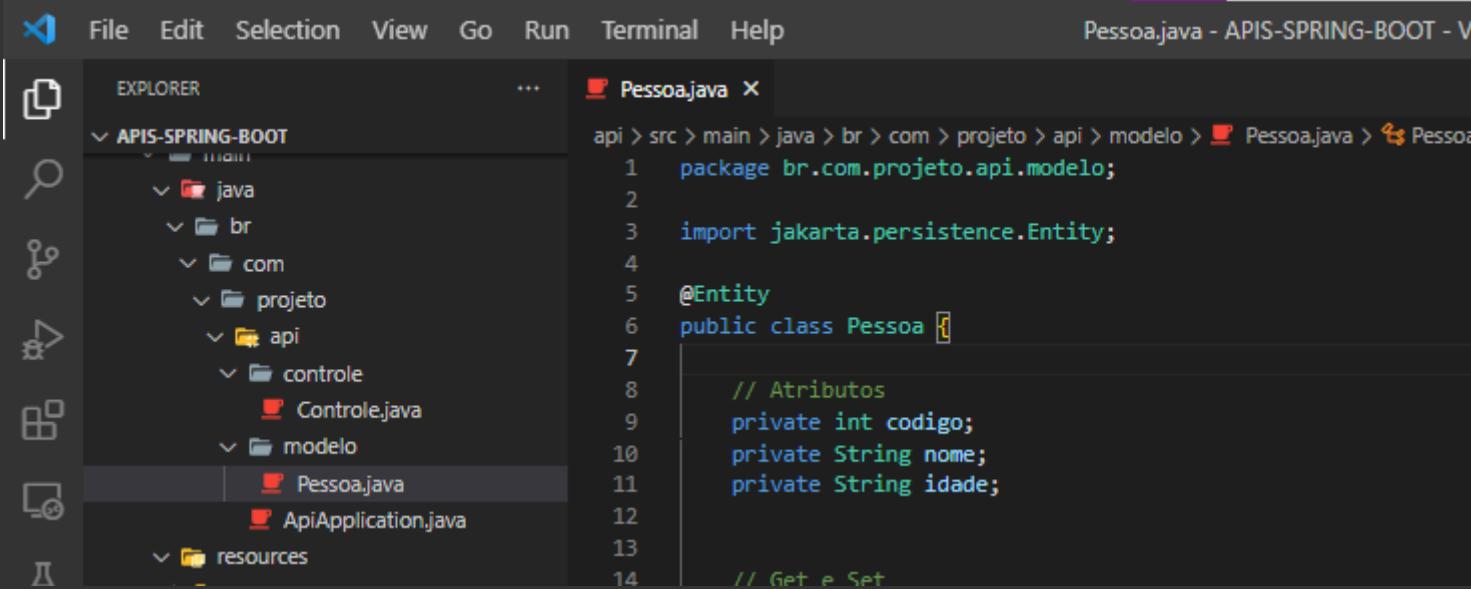
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoajava - APIS-SPRIN
- Explorer:** Shows the project structure under APIS-SPRING-BOOT:
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - Controle.java
 - modelo
 - Pessoajava
 - ApiApplication.java
 - resources
- Editor:** Displays the content of Pessoajava.java.

```
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 }
```

A code editor sidebar on the left shows the current file being edited: Pessoajava.java.

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:** The file "Pessoajava.java" is open, displaying the following code:

```
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(name = "pessoas")". A yellow box highlights the opening brace of the class definition.

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

Vamos adicionar a chave primaria

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



EXPLORER

APIS-SPRING-BOOT

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

Pessoa.java 1

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

jakarta.persistence.Id

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

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Pessoajava - API-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under "APIS-SPRING-BOOT":
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle (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 codigo = Generated - jakarta.annotation
15 private String nome = Generated - javax.annotation.processing
16 private String idade = Generated - org.hibernate.annotations
17     + GeneratedColumn - org.hibernate.annotations
18     + GeneratedColumn - org.hibernate.annotations.Dialect
19     + GeneratedColumns - org.hibernate.annotations.Dialect
20     + GeneratedValue - jakarta.persistence
21     return GeneratorType - jakarta.persistence.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`.

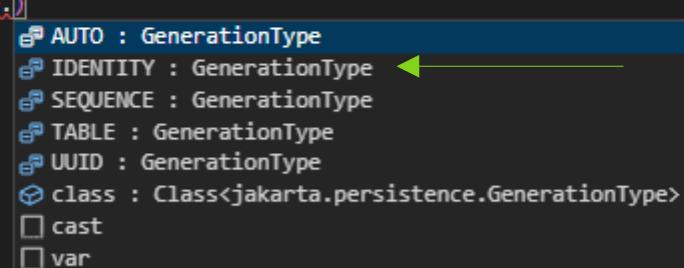
Pessoa.java ●

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

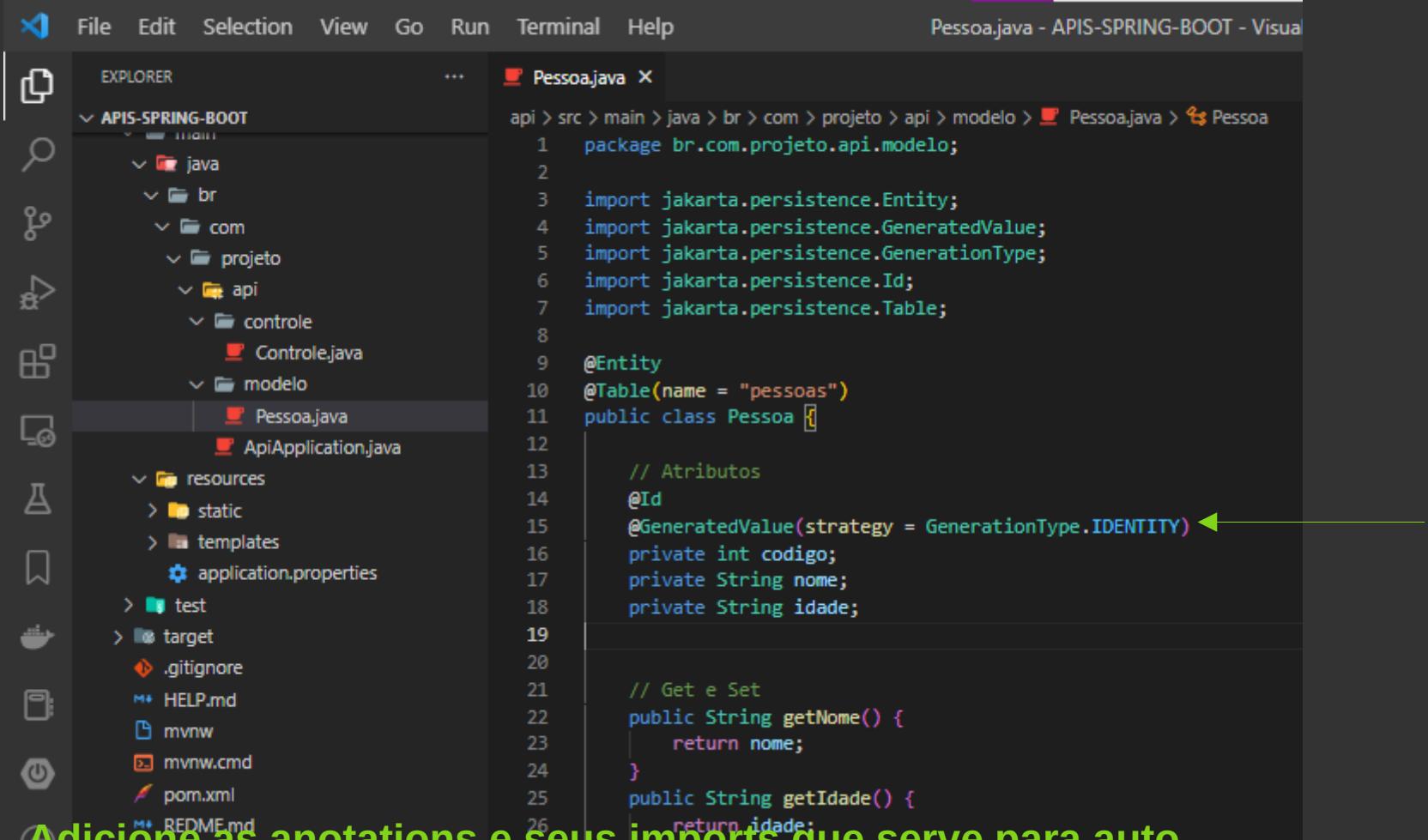
```
1 package br.com.projeto.api.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue; ←
5 import jakarta.persistence.Id;
6 import jakarta.persistence.Table;
7
8 @Entity
9 @Table(name = "pessoas")
10 public class Pessoa {
11
12     // Atributos
13     @Id
14    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

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 Sidebar:** Shows the project structure:
 - API-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - resources
 - static
 - templates
 - application.properties
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - README.md

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

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

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

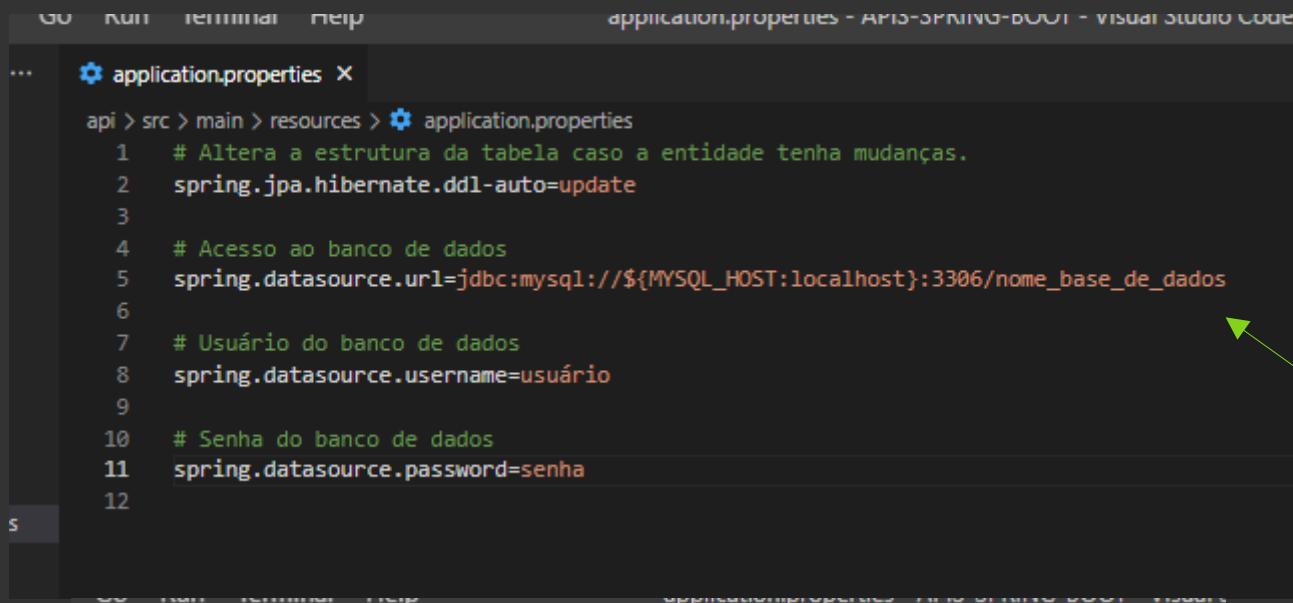
Se o arquivo pom.xml marcar um erro feche o vscode e abra novamente

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

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

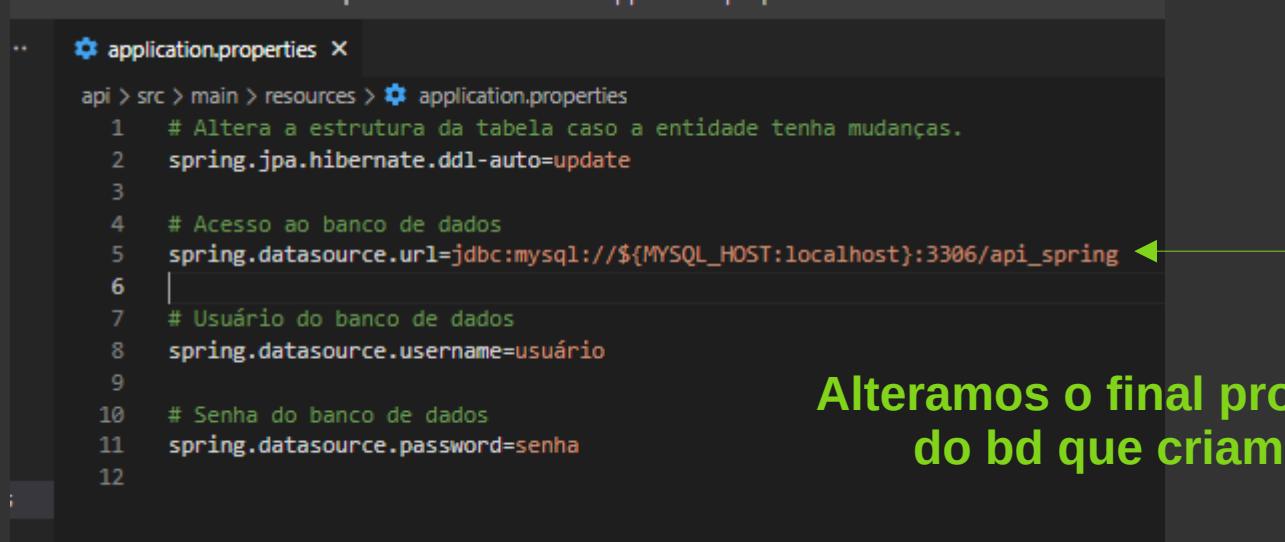
Annotations are present in the bottom right corner of the code editor:

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



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

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



application.properties X

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

Alteramos o final pro nome
do bd que criamos

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

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

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

es

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

Connect X

Connect to server

Name Connection Name Connection Scope Global Premium Only

Group Group Connect Timeout 5000 Read Only Save Password Forever

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

Server Type

MySQL MariaDB PostgreSQL SQL Server Oracle SQLite DM8 ClickHouse Redshift

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

* Host 127.0.0.1 * Port 3306

* Username root * Password

Database Connection database Advance Option

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

Save Connect Close

Desses dados que
estou falando

Senha:
Akilles101

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

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

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

Akilles101

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

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

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

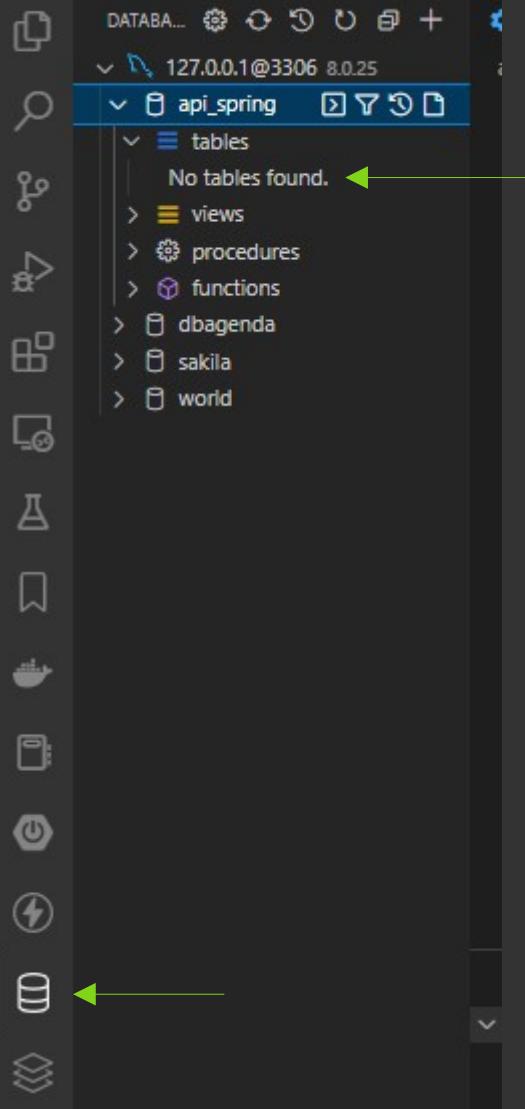
The screenshot shows the Visual Studio Code interface with the following details:

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

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

Salve os dados

File Edit Selection View Go

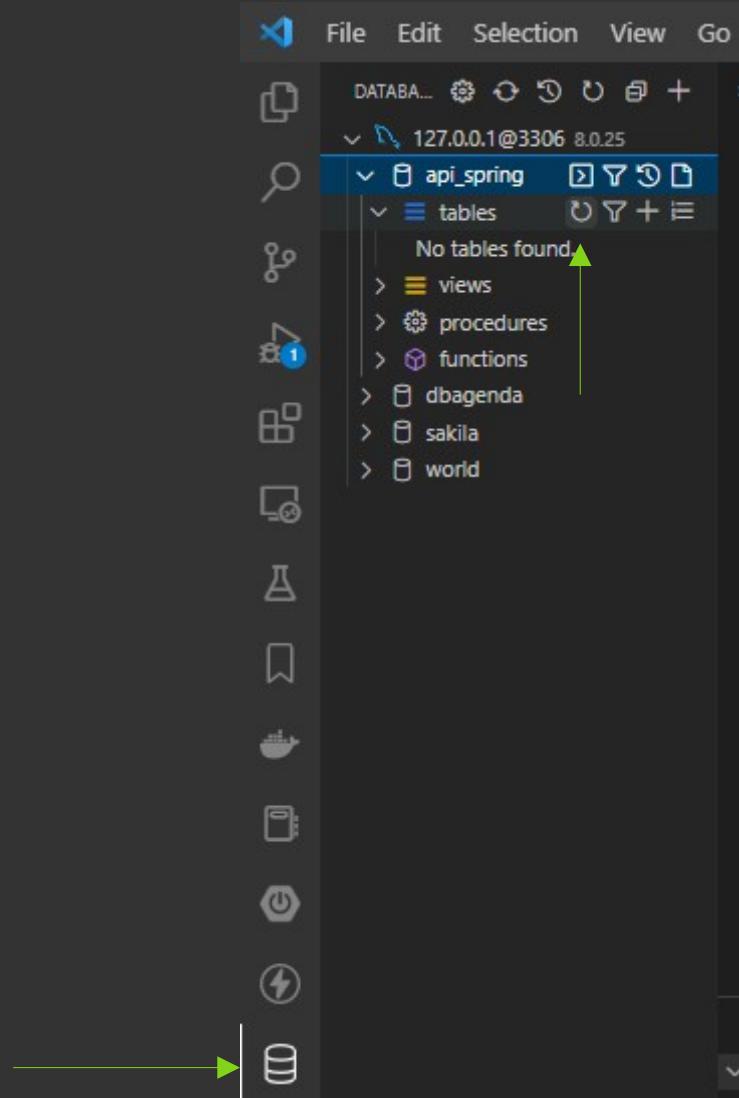


Abra o bad pra ver que ainda não existem tabelas

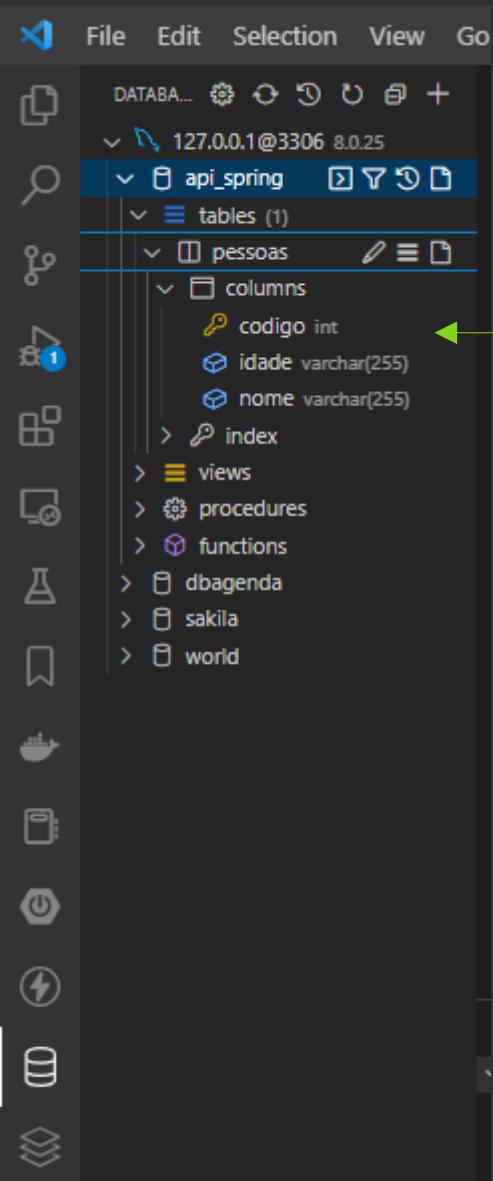
The screenshot shows the Visual Studio Code interface with the following details:

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

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



Depois de executar
atualize refresh



Os dados foram criados

#14
Criando repositorio

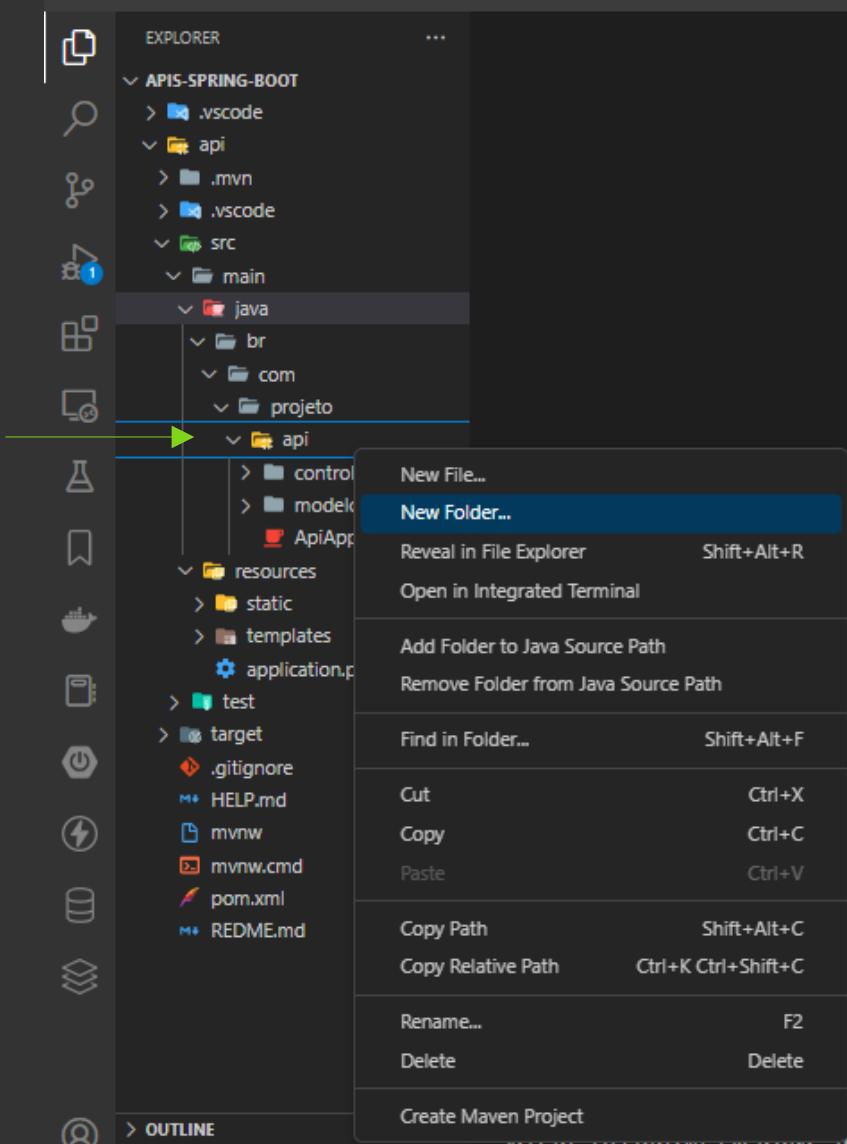
o que é um repositorio?

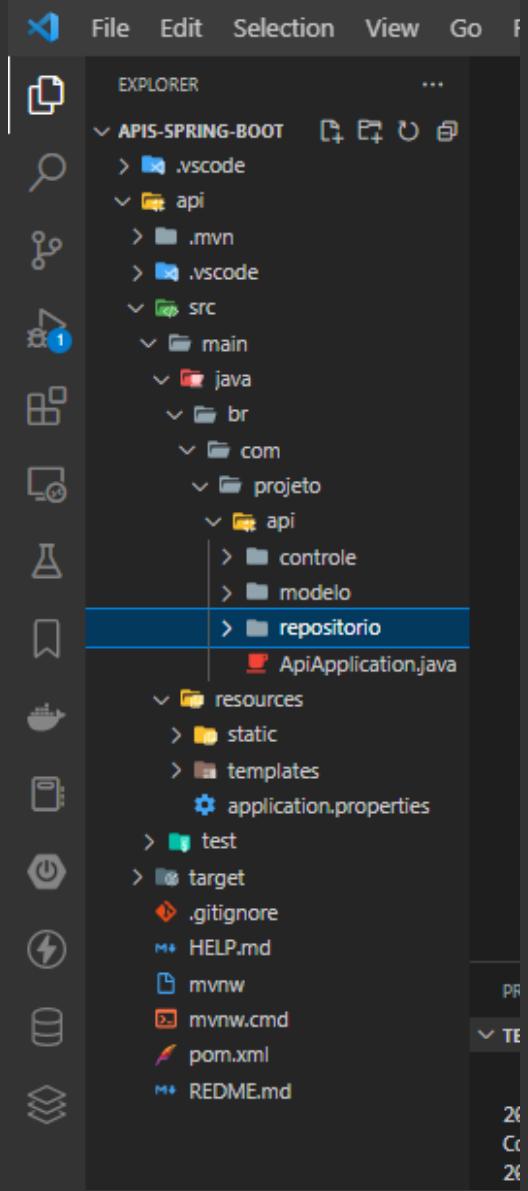
Uma camada de persistencia

O que é uma camada de persistencia?

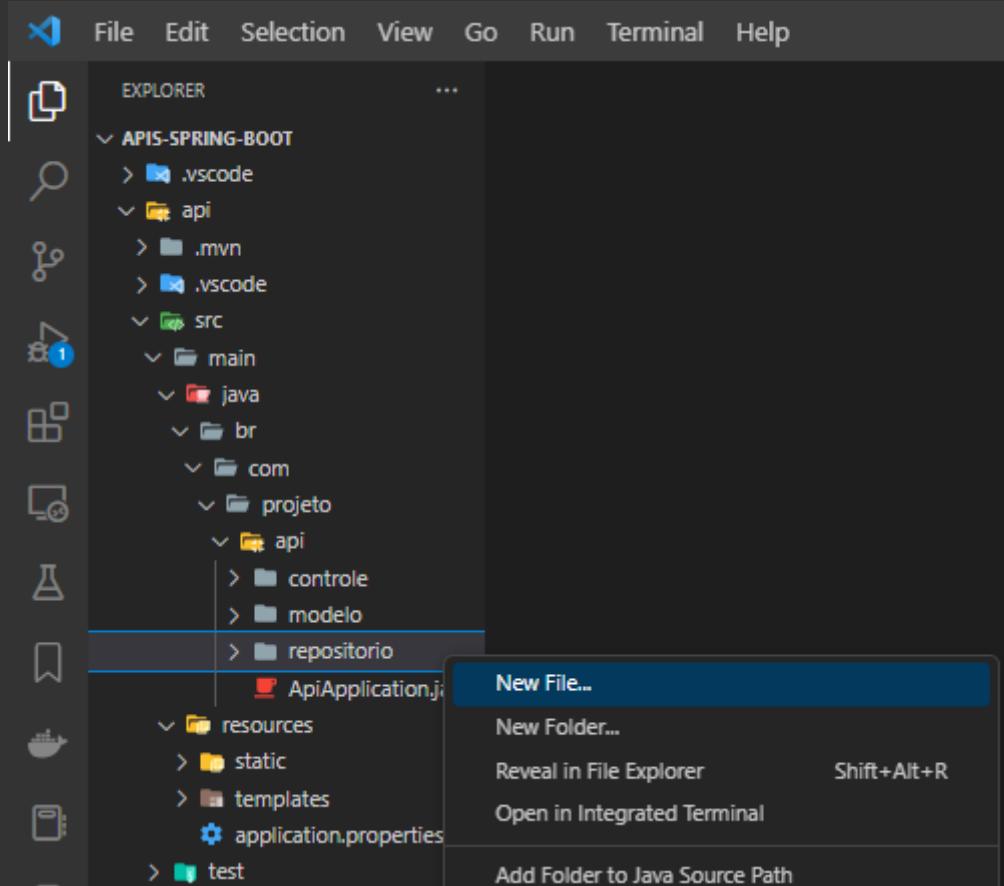
Parte do projeto onde fazemos as ações do banco de dados

Os chamados cruds





Crie a pasta
repositorio



File Edit Selection View Go Run Terminal Help

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

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

Repositorio.java

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

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 and java folders), and java (with br, com, projeto, and api subfolders).
- 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 interface with the following details:

- File Bar:** File Edit Selection View Go Run Terminal Help
- Title Bar:** Repositorio.java - APIS-SPRING-BOOT
- Explorer Panel (Left):** Shows the project structure:
 - APIS-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - Editor Panel (Right):** Displays the code for `Repositorio.java`:

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

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

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

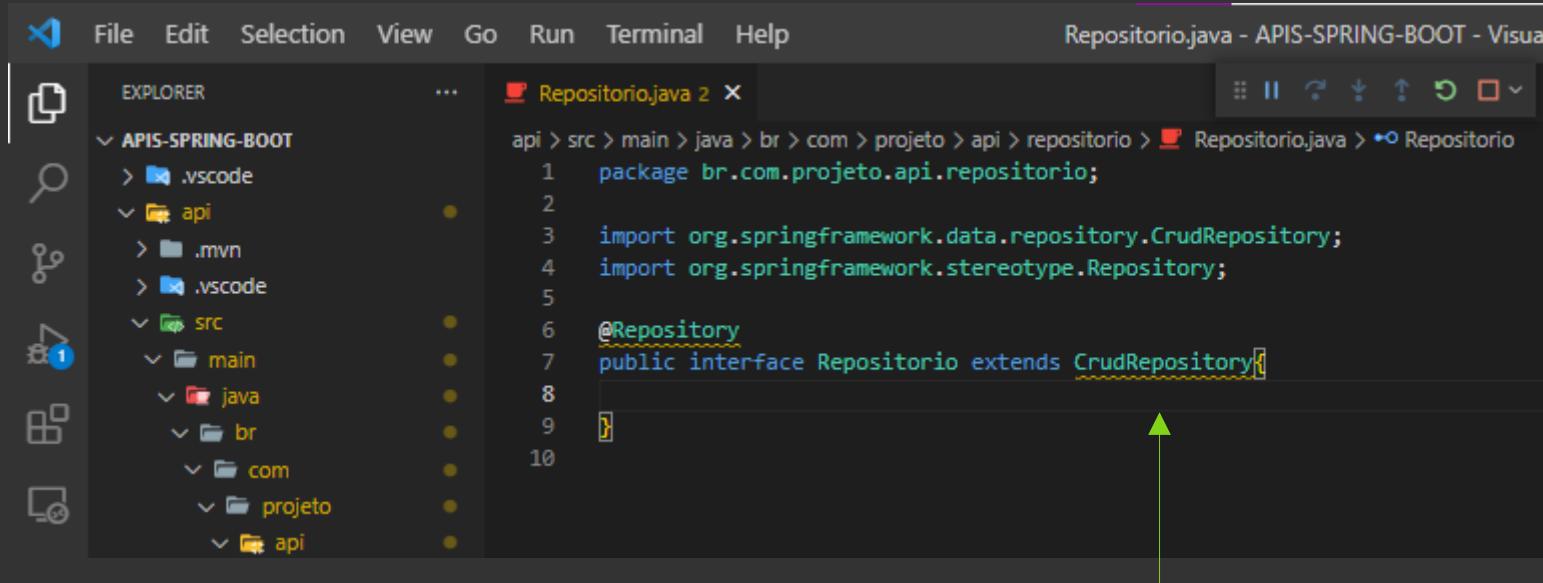
```
api > src > main > java > br > com > projeto > api > repositorio > Repository.java > Repository
1 package br.com.projeto.api.repositorio;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public interface Repository extends Crud[
```
- IntelliJ IDEA Completion Screenshot:**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

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

• Author:

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

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



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

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

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

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

Primeiro adicionamos o modelo Pessoa e importamos

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

- File 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" namespace.
- 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 Title:** • Repositorio.java - APIS-SPRING-BOOT - Visual Studio
- Editor Content:** A Java code file named `Repositorio.java`. The code defines an interface `Repositorio` that extends `CrudRepository<Pessoa, Long>`. The code is as follows:

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

The Explorer pane on the left shows the project structure:

- APIS-SPRING-BOOT
- ↳ .vscode
- ↳ api
- ↳ .mvn
- ↳ .vscode
- ↳ src
 - ↳ main
 - ↳ java
 - ↳ br
 - ↳ com
 - ↳ projeto
 - ↳ api
 - ↳ controle
 - ↳ modelo

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

File Edit Selection View Go Run Terminal Help

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

EXPLORER

APIS-SPRING-BOOT

- .vscode
- api
- .mvn
- .vscode
- src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - modelo
 - repositorio
- Repositorio.java
- ApiApplication.java
- resources
 - static
 - templates
- application.properties

Repositorio.java

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

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

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

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

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

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

REPOSITORY

OUTLINE

TIMELINE

JAVA PROJECTS

MAVEN

File 0 Δ 1 Spring Boot-ApiApplication<api> (APIS-SPRING-BOOT)

Repository.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>
  </dependencies>

```

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

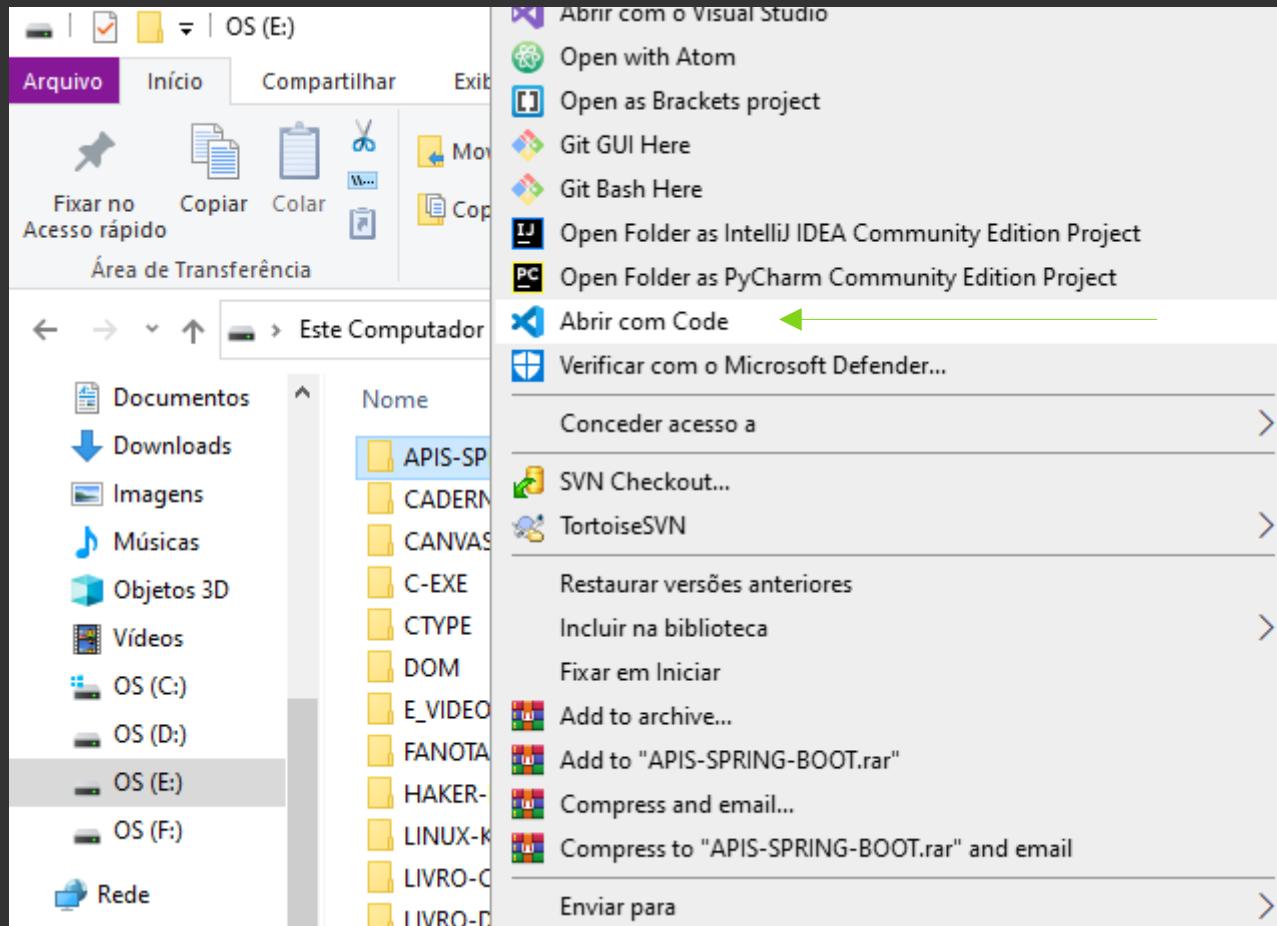
Do you want to install the Dependency Analytics extension to stay informed about vulnerable dependencies in pom.xml files?

Source: Language Support for Java(TM) by Red Hat ... [install](#) [never](#) [later](#)

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

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

Fechei e abri de novo o vscode



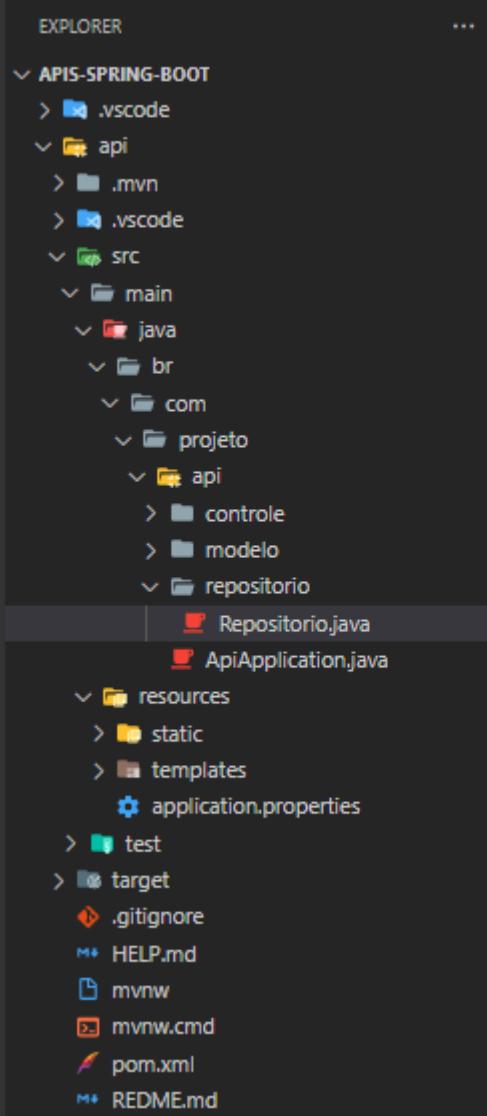
The screenshot shows a Visual Studio Code interface with the following details:

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

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

- Terminal View:** Shows a PowerShell session in the 'TERMINAL' tab. The session starts with a welcome message from Microsoft, followed by a command to restore history, and then displays the standard Windows PowerShell copyright notice.
- Status Bar:** Shows 'Ln 12, Col 1' and other status indicators like 'Spaces: 4', 'UTF-8', 'CRLF', 'Java', 'Go Live', and file icons.

A large green text overlay 'E o erro sumiu voltou o erro' is centered in the terminal area.



REPORATORIO.java X pom.xml

```
api > src > main > java > br > com > projeto > api > repositorio > Reppositorio.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 Reppositorio 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 the Visual Studio Code interface with the following details:

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

```
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

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

@RestController
public class Controle {

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

TERMINAL

Estamos criando um objeto de forma privada

Repositories - org.springframework.data.repository...
RepositoriesPopulatedEvent - org.springframework...
Repositorio - br.com.projeto.api.repositorio
Repository - org.apache.catalina.startup.ClassLo...
Repository - org.aspectj.apache.bcel
Repository - org.aspectj.apache.bcel.util
Repository - org.springframework.boot.actuate.aut...
Repository - org.springframework.data.repository
Repository - org.springframework.stereotype
RepositoryAwareMethodLookup - org.springframework...
RepositoryBeanDefinitionParser - org.springframew...
RepositoryBeanDefinitionRegistrarSupport - org.sp...

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

modelo

repositorio

- Repository.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>

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

```
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import br.com.projeto.api.modelo.Pessoa;
11 import br.com.projeto.api.repositorio.Repositorio;
12
13 @RestController
14 public class Controle {
15
16     @Autowired
17     private Repositorio acao; // 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 }
```

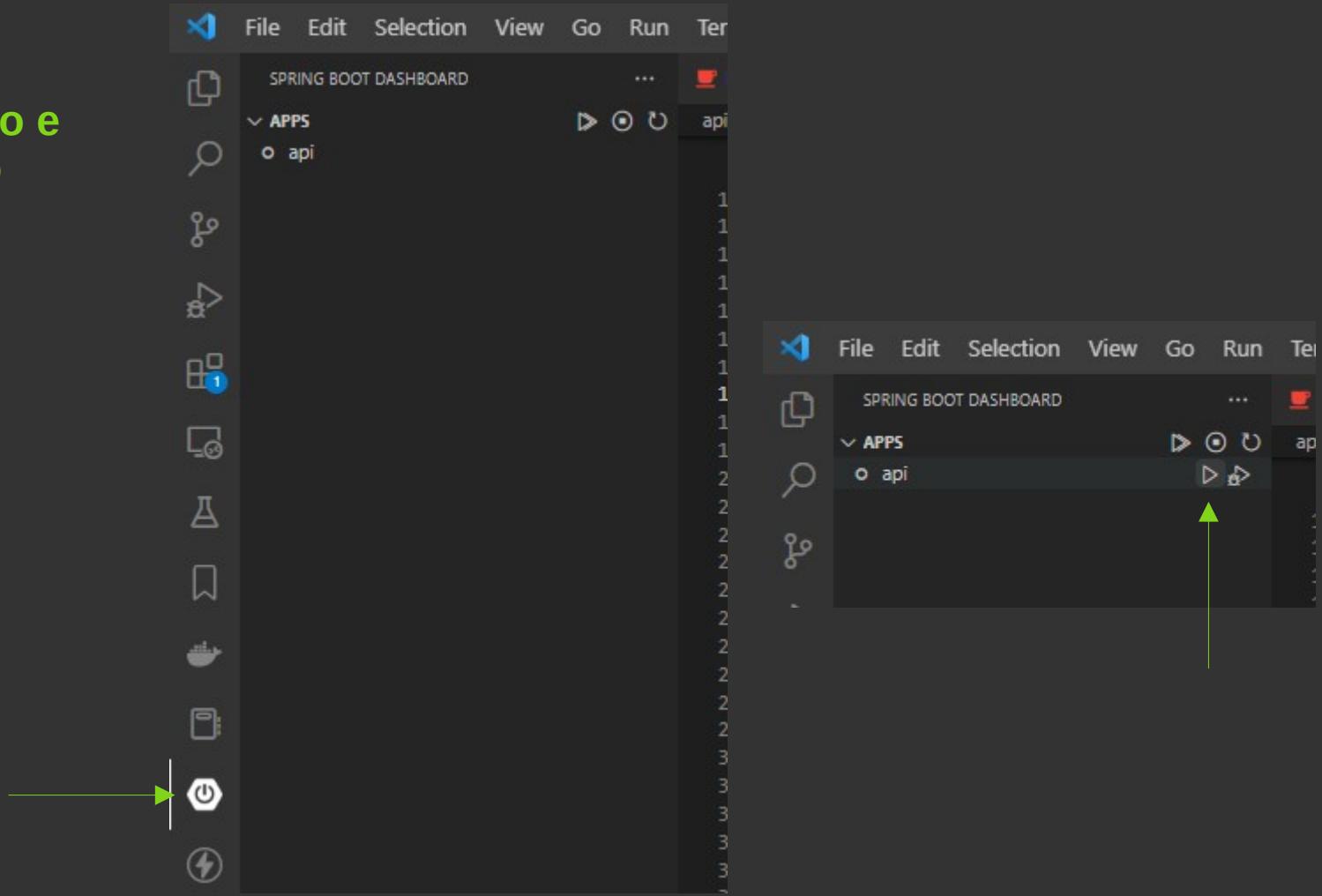
Nesse caso o objeto acao pode chamar esses metodos crud's

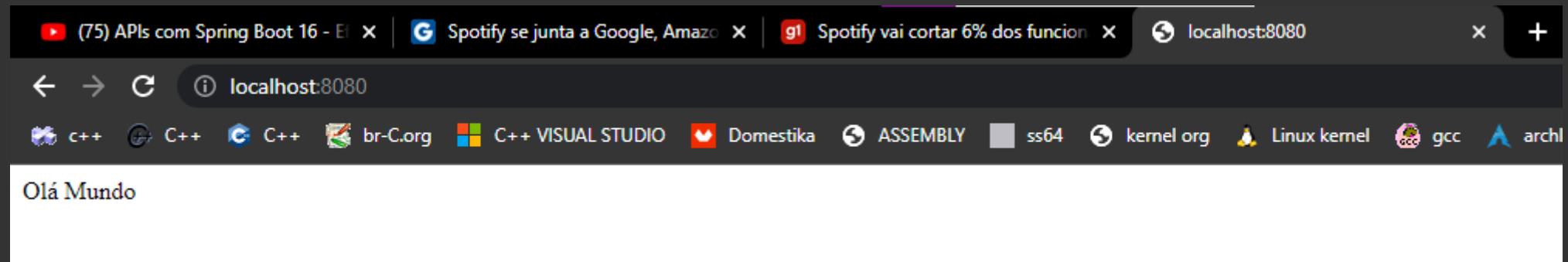
E onde esta a instancia onde esta o new para criar um novo objeto?

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:

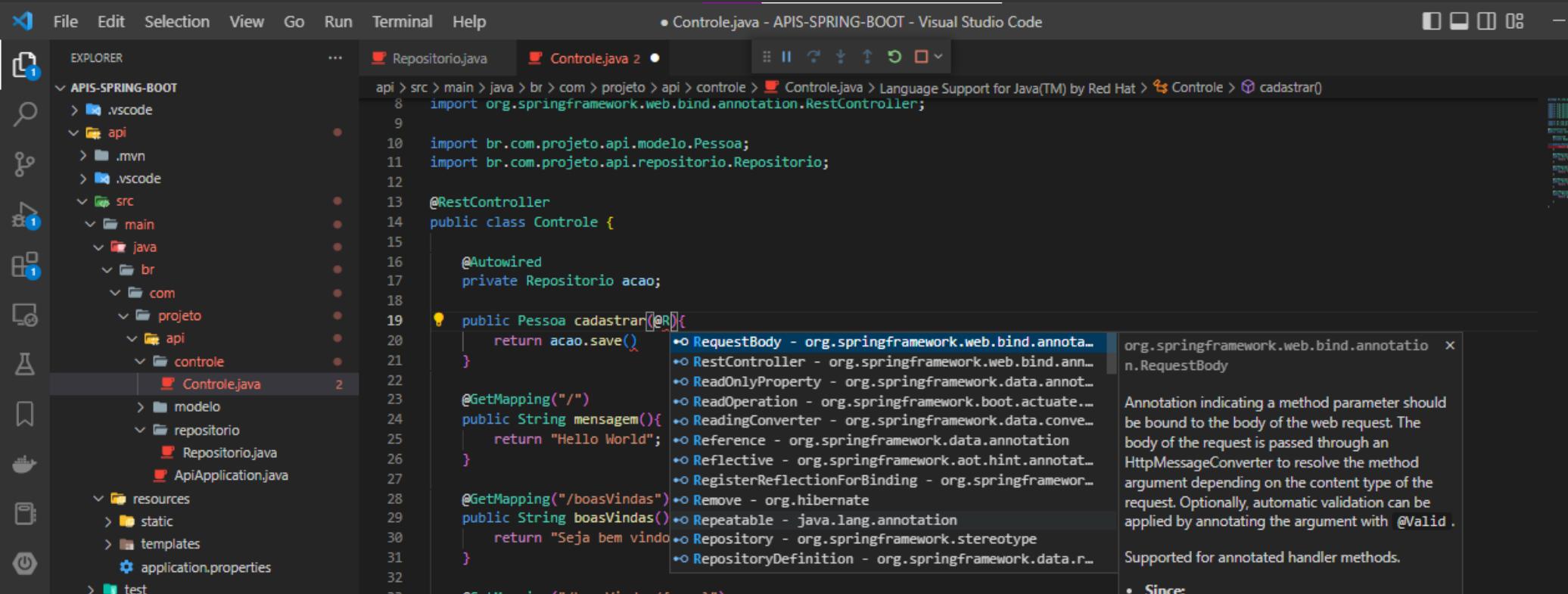
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - API-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
 - API-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - 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 Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure under APIS-SPRING-BOOT, including .vscode, api, src, resources, static, templates, and test folders. Inside src/main/java, there are br, com, projeto, api, and controle packages. The Controle.java file is selected in the list.
- Code Editor:** Displays the Controle.java 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(@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 Hover 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.
- Right Panel:** Shows the Java Language Support for Java(TM) by Red Hat extension in the Marketplace.

File Edit Selection View Go Run Terminal Help

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

EXPLORER

APIS-SPRING-BOOT

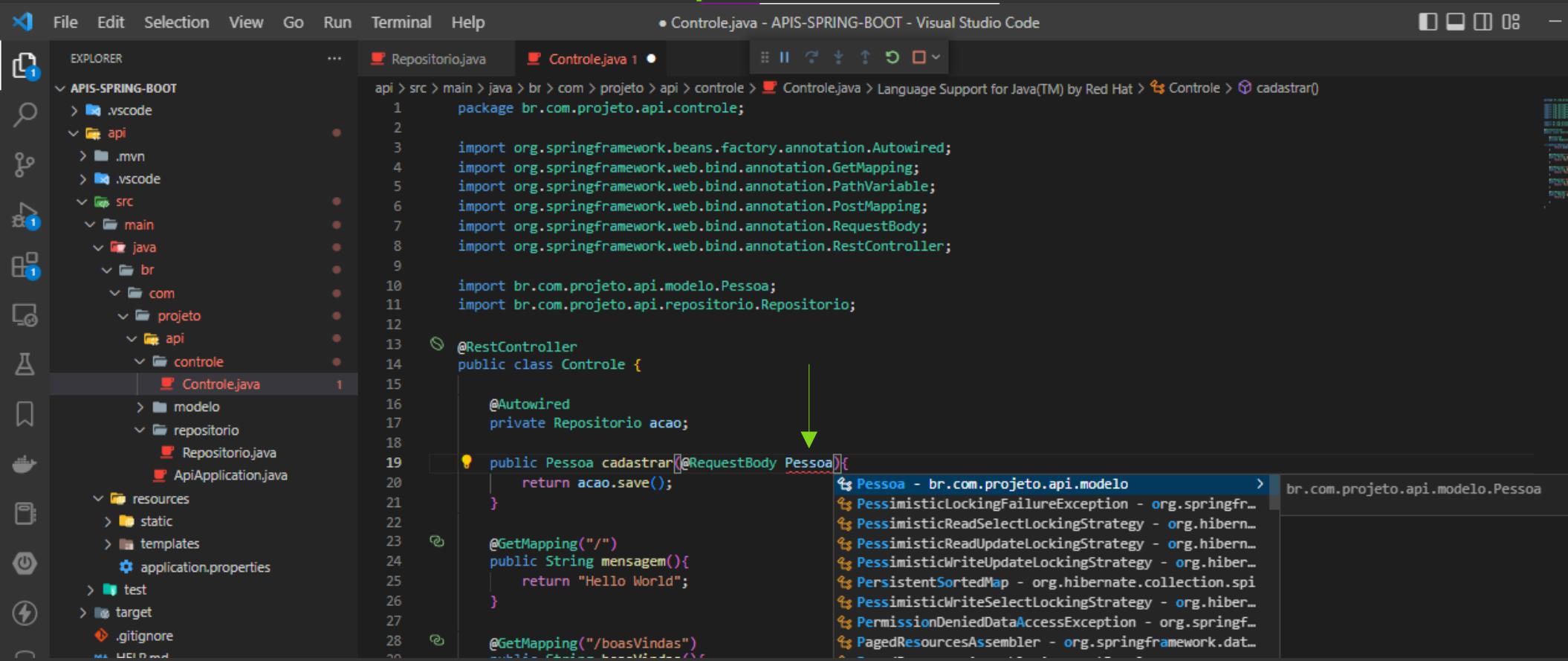
- .vscode
- api
 - .mvn
 - .vscode
- src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - resources
 - static

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
 - Controle.java
 - resources
 - static
- Editor View:** Controle.java 1 X (active tab).

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

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

E damos o nome de obj

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

- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT". The "src" folder contains "main", "java", "br", "com", "projeto", "api", and "controle". Inside "controle", there is a file named "Controle.java".
- Editor (Center):** Displays the code for "Controle.java".

```
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){[{"id": 1, "name": "arrow-up"}]
20         return acao.save(obj);
21     }
22 }
```

A green arrow points upwards from the explanatory text below to the closing brace of the method's body.
- Terminal (Top Right):** Shows the command "Controle.java - APIS-SPRING-BOOT - visual studio".

O save efetua o cadastramento do objeto do tipo Pessoa

File Edit Selection View Go Run Terminal Help

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

EXPLORER

APIS-SPRING-BOOT

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

resources

 - > static
 - > templates
 - > application.properties

test

Repositorio.java Controle.java 1

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

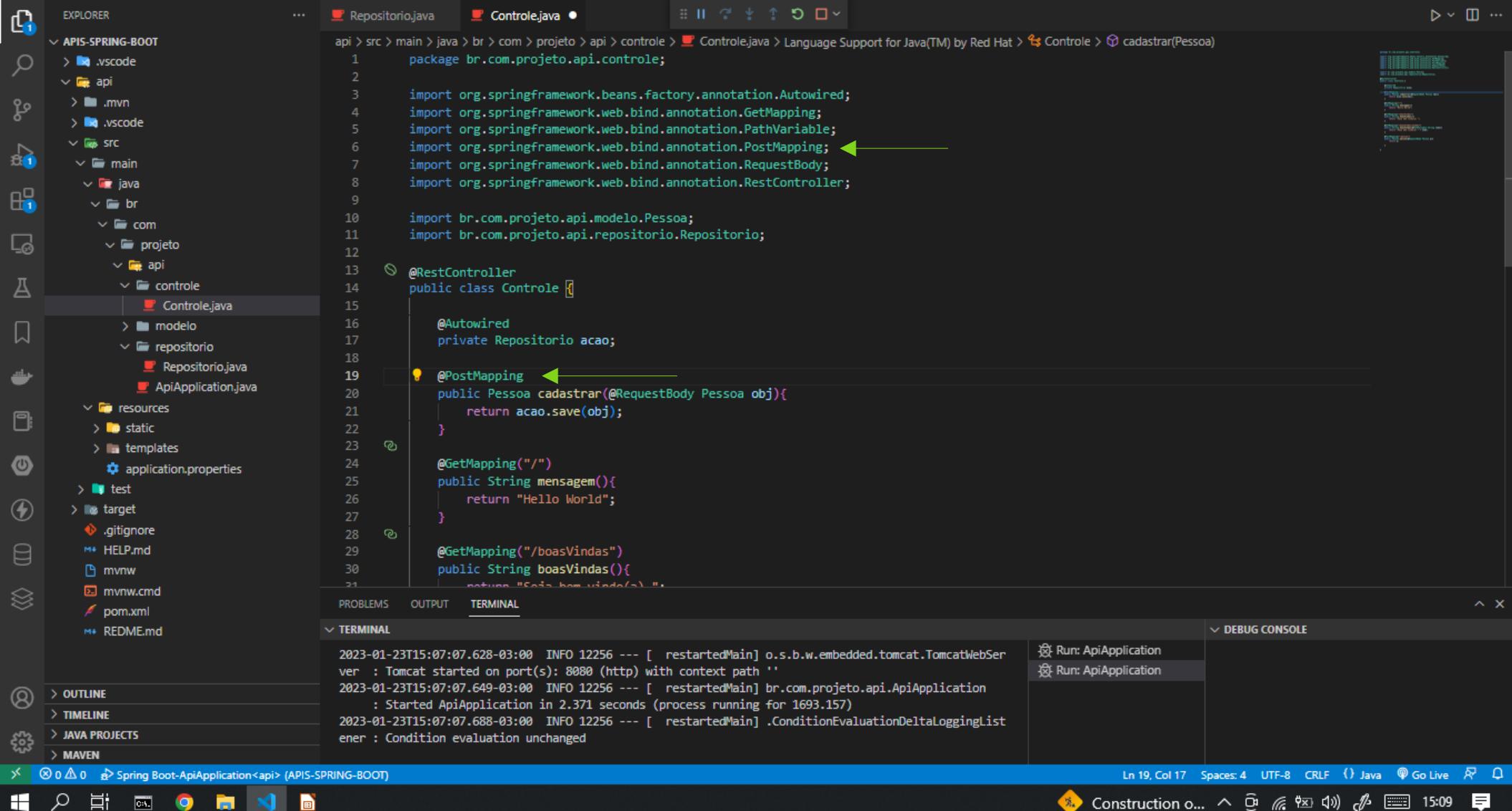
```
1 package br.com.projeto.api.controle;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PathVariable;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import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

Annotation for mapping HTTP POST requests onto specific handler methods.

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

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

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



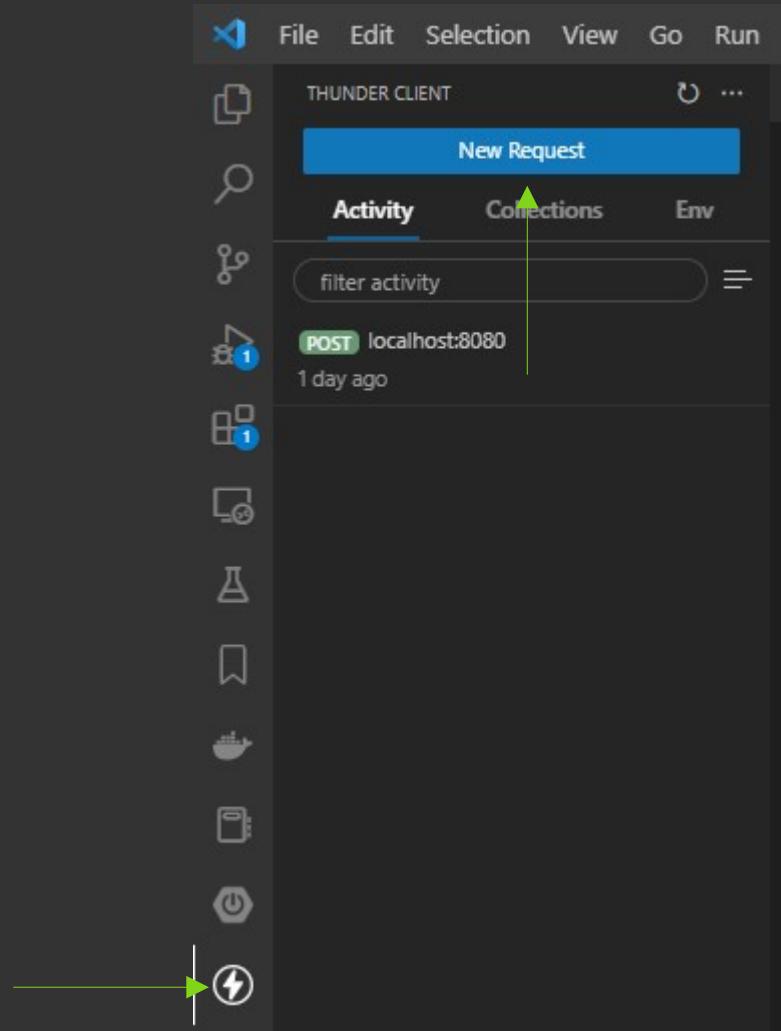
The screenshot shows the Visual Studio Code interface with the following details:

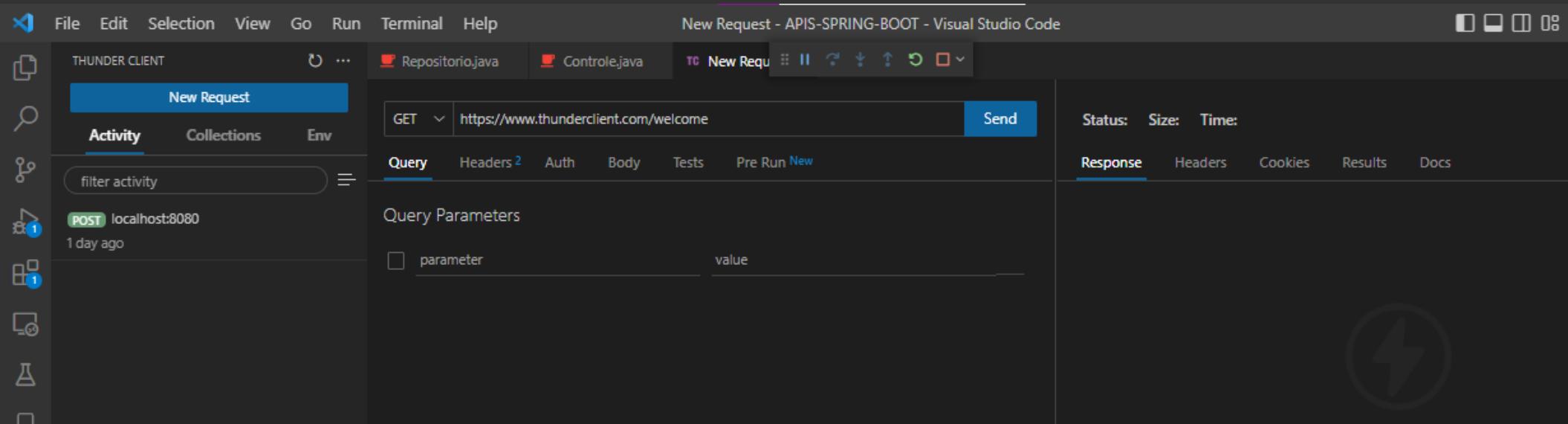
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
 - APIS-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle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

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

Por fim colocamos a rota (“/api”)

Salve abra o thunder client e faça uma nova requisição





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

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

New Request

Activity Collections Env

filter activity

POST localhost:8080 1 day ago

POST http://localhost:8080/api

Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Headers 2 Auth Body Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Format

Send Request Ctrl + Enter

Import Curl Ctrl + U

Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

TERMINAL

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

DEBUG CONSOLE

Run: ApiApplication

Run: ApiApplication

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

Go Live

Chuva chegando

File Edit Selection View Go Run Terminal Help

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

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080/api just now

POST localhost:8080 1 day ago

Repositorio.java Controle.java New Request

POST http://localhost:8080/api Send

Status: Size: Time:

Response Headers Cookies Results Docs

Query Headers 2 Auth Body 1 Tests Pre Run New

Json XML Text Form Form-encode GraphQL Binary

Json Content Format

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

Send Request Ctrl Enter

crie um objeto json

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

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

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

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

New Request

Activity Collections Env

filter activity

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Response Headers Cookies Results Docs

Status: Size: Time:

Json Content Format

Json XML Text Form Form-encode GraphQL Binary

Json Content

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

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8080/api 3 mins ago

POST localhost:8080 1 day ago

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content Format

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

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

Response Headers 4 Cookies Results Docs

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

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

PROBLEMS OUTPUT TERMINAL

TERMINAL

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

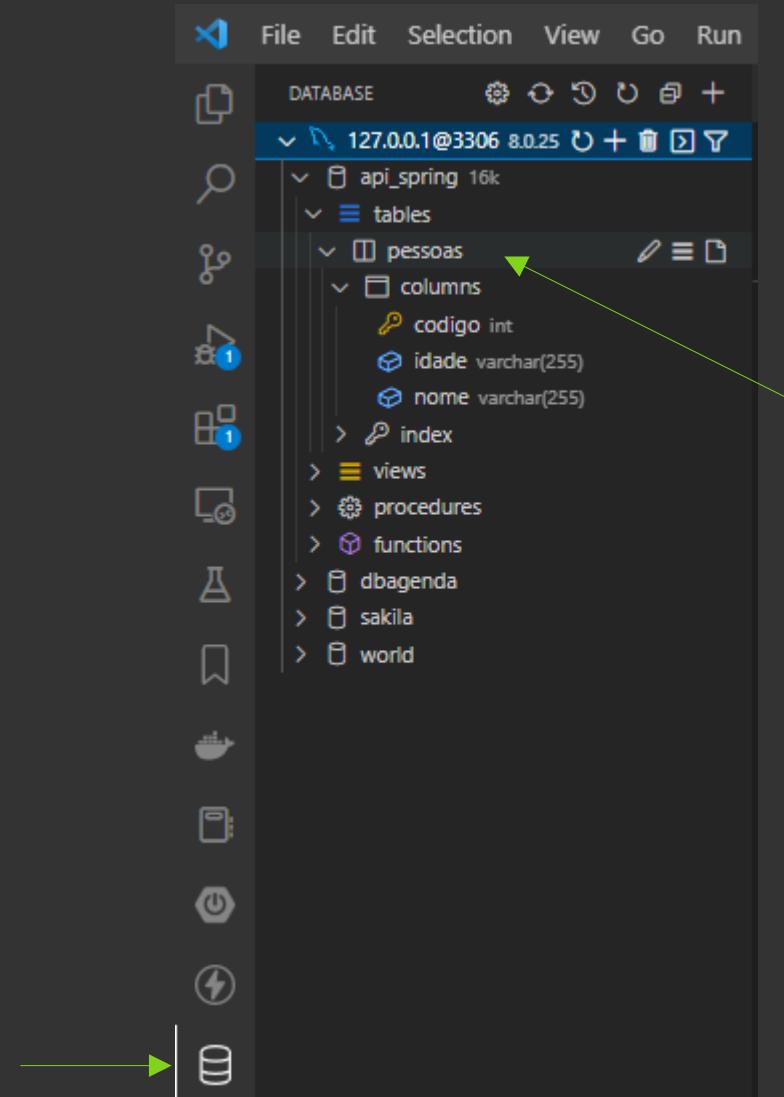
DEBUG CONSOLE

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

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

Chuva chegando

Go Live



Click duplo em pessoas

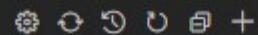


File Edit Selection View Go Run Terminal Help

pessoas - APIS-SPRING-BOOT - Visual Studio Code



DATABASE



127.0.0.1@3306 8.0.25



api_spring 16k



tables (1)



> pessoas



> views



> procedures



> functions



> dbagenda



> sakila



> world



Repository.java

Controle.java

New Requ

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

Input to filter result

Free

1

+

-

?

↑

↓

▶

◀

Cost: 90ms

1

Total 1

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

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

Fazendo mais um

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

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

POST http://localhost:8080/api Send

Query Headers 2 Auth Body 1 Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Json Content Format

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

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

Response Headers 4 Cookies Results Docs

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

File Edit Selection View Go Run Terminal Help

pessoas - APIS-SPRING-BOOT - Visual Studio Code

DATABASE

127.0.0.1@3306 8.0.25

api_spring 16k

tables (1)

pessoas

columns

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

index

views

procedures

functions

dbagenda

sakila

world

Repository.java

Controle.java

New Req

SELECT * FROM pessoas LIMIT 100;

Input to filter result

Free 1

Locked

* codigo idade nome

int varchar(255) varchar(255)

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

Cost: 8ms < 1 > Total 2

#17

Listando os dados com o comando `findAll()`

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

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

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

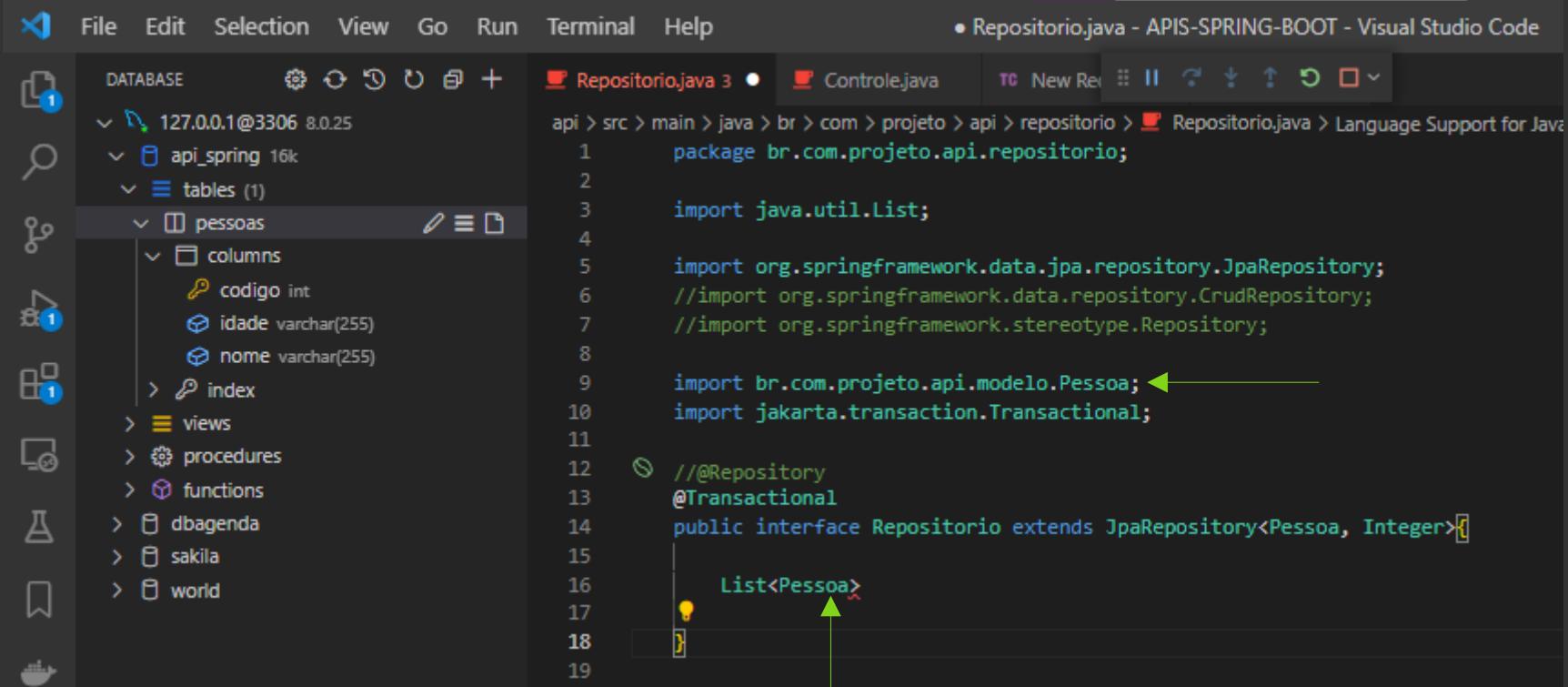
```
• Repositorio.java - APIS-SPRING-BOOT - Visual Studio Code
```

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 a Visual Studio Code interface with a dark theme. On the left is a sidebar titled "DATABASE" showing a connection to "127.0.0.1@3306 8.0.25" and a schema named "api_spring". Inside "api_spring", there is a table named "pessoas" with columns "codigo" (int), "idade" (varchar(255)), and "nome" (varchar(255)). Other 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 }
```

A yellow lightbulb icon is visible at the bottom of the code editor, indicating a potential issue or suggestion.

**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 and other database objects like views, procedures, and functions.
- Code Editor:** The file "Repositorio.java" is open. The code defines a public interface "Repositorio" that extends "JpaRepository<Pessoa, Integer>". It includes a native query 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]: Tomcat started on port(s) :8080 [Tomcat]: Started Application in 3.823 seconds (process running for 3491704)



File Edit Selection View Go Run Terminal Help

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



DATABASE



127.0.0.1@3306 8.0.25

api_spring 16k

tables (1)

pessoas



columns

codigo int

idade varchar(255)

nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world

List porque
vamos importar
uma lista de
pessoas

Repository.java

Controle.java 1

New Rec

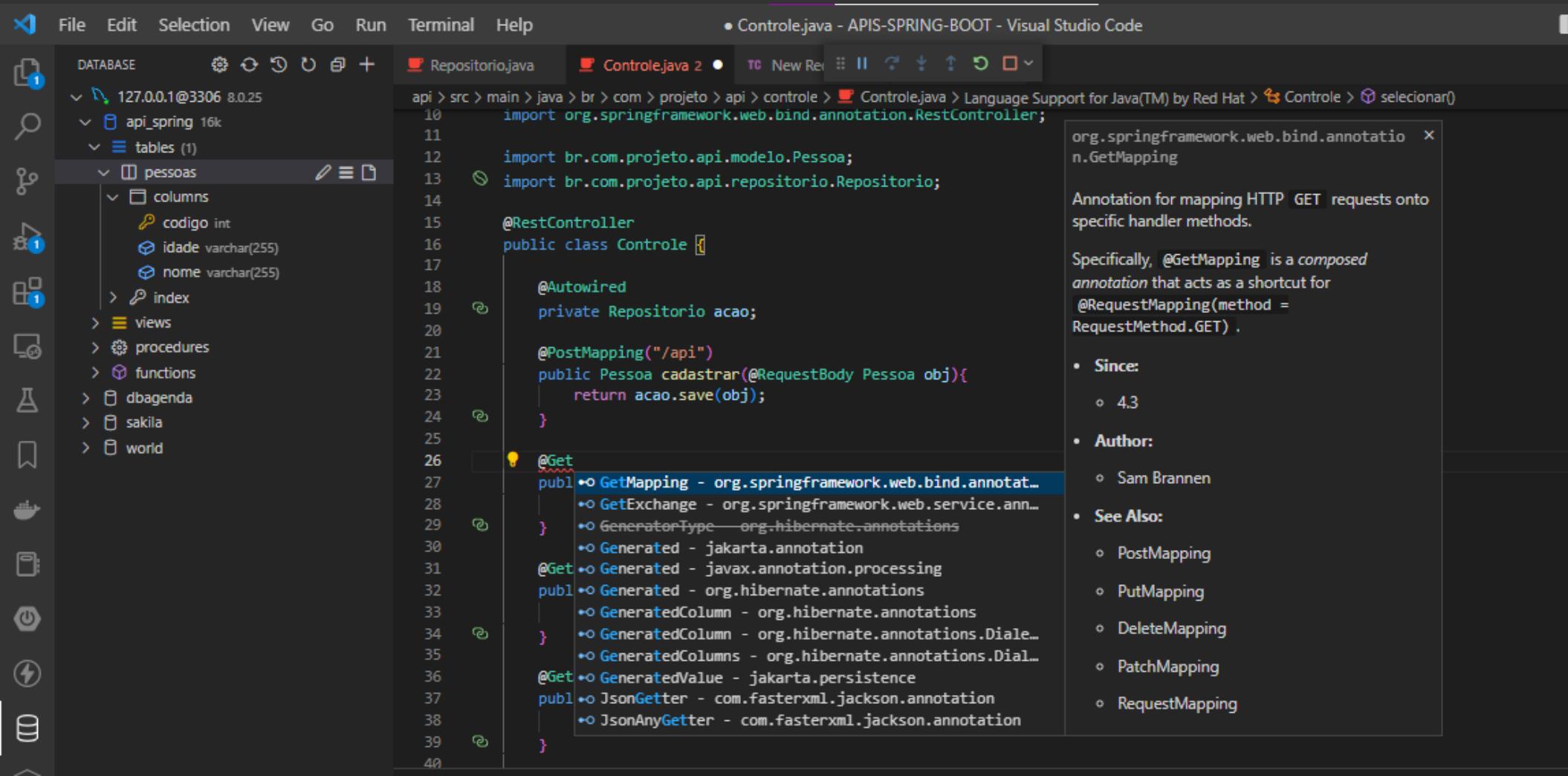


api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java(TM)

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

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

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



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

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

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

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

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

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

@RestController
public class Controle {

    @Autowired
    private Repositorio acao;

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

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



File Edit Selection View Go Run Terminal Help

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



DATABASE



127.0.0.1@3306 8.0.25

api_spring 16k

tables (1)

pessoas



columns

codigo int

idade varchar(255)

nome varchar(255)

index

views

procedures

functions

dbagenda

sakila

world



Só pra dar uma
olhada de como
esta até aqui

Repository.java

Controle.java X

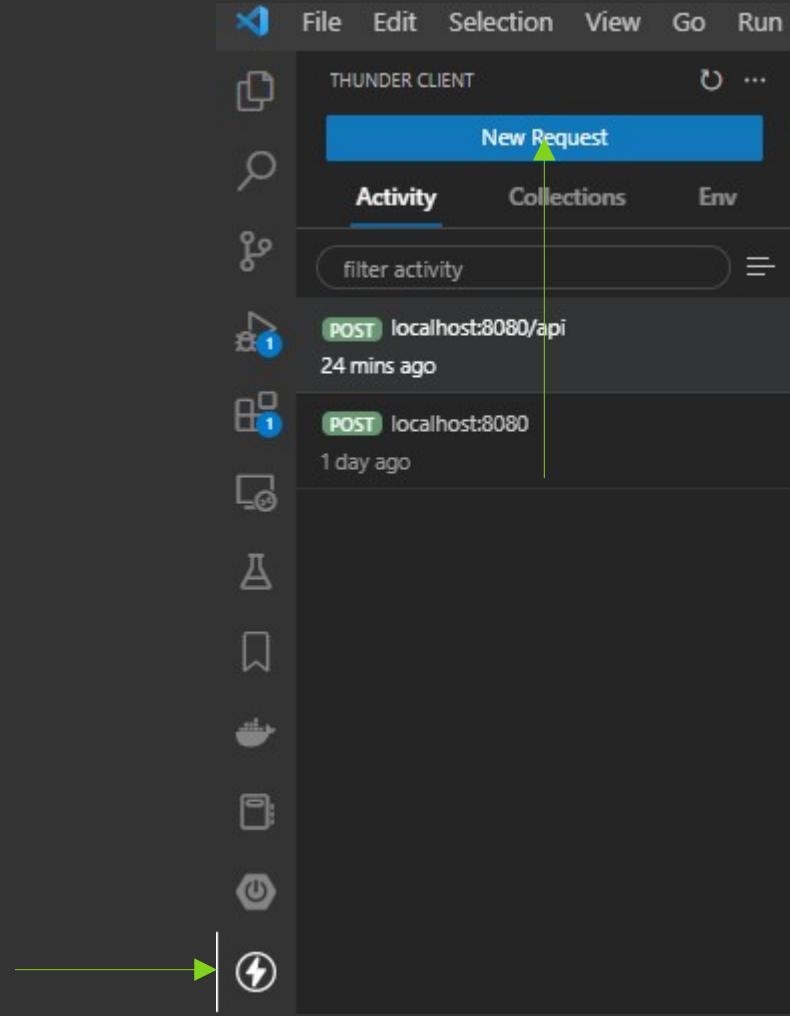
New Requ

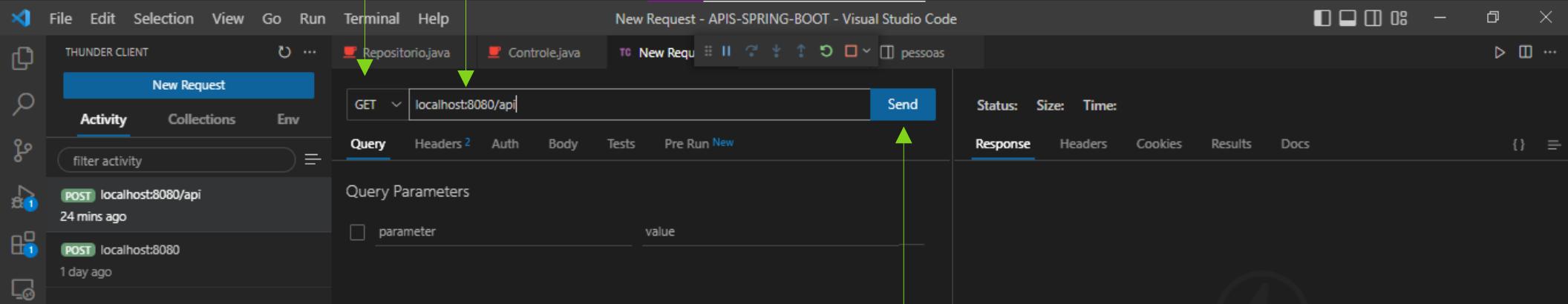


```
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support
1     package br.com.projeto.api.controle;
2
3     import java.util.List;
4
5     import org.springframework.beans.factory.annotation.Autowired;
6     import org.springframework.web.bind.annotation.GetMapping;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

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

Vamos no thunder client e realizamos uma nova requisição





THUNDER CLIENT

New Request

Activity Collections Env

filter activity

GET localhost:8080/api just now

POST localhost:8080/api 24 mins ago

POST localhost:8080 1 day ago

Repository.java Controle.java New Req pessoas

GET localhost:8080/api Send

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

parameter value

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

Response Headers 4 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 ]
```

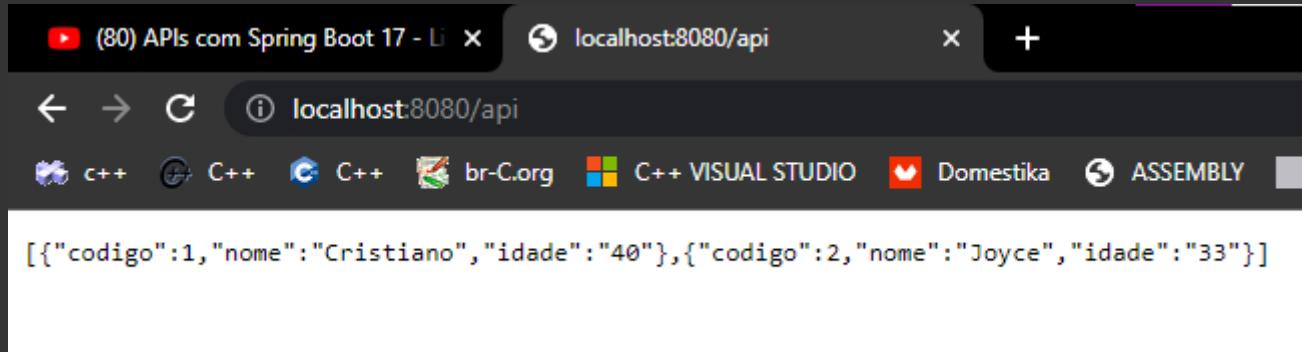
Dados passados

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 web browser window. 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

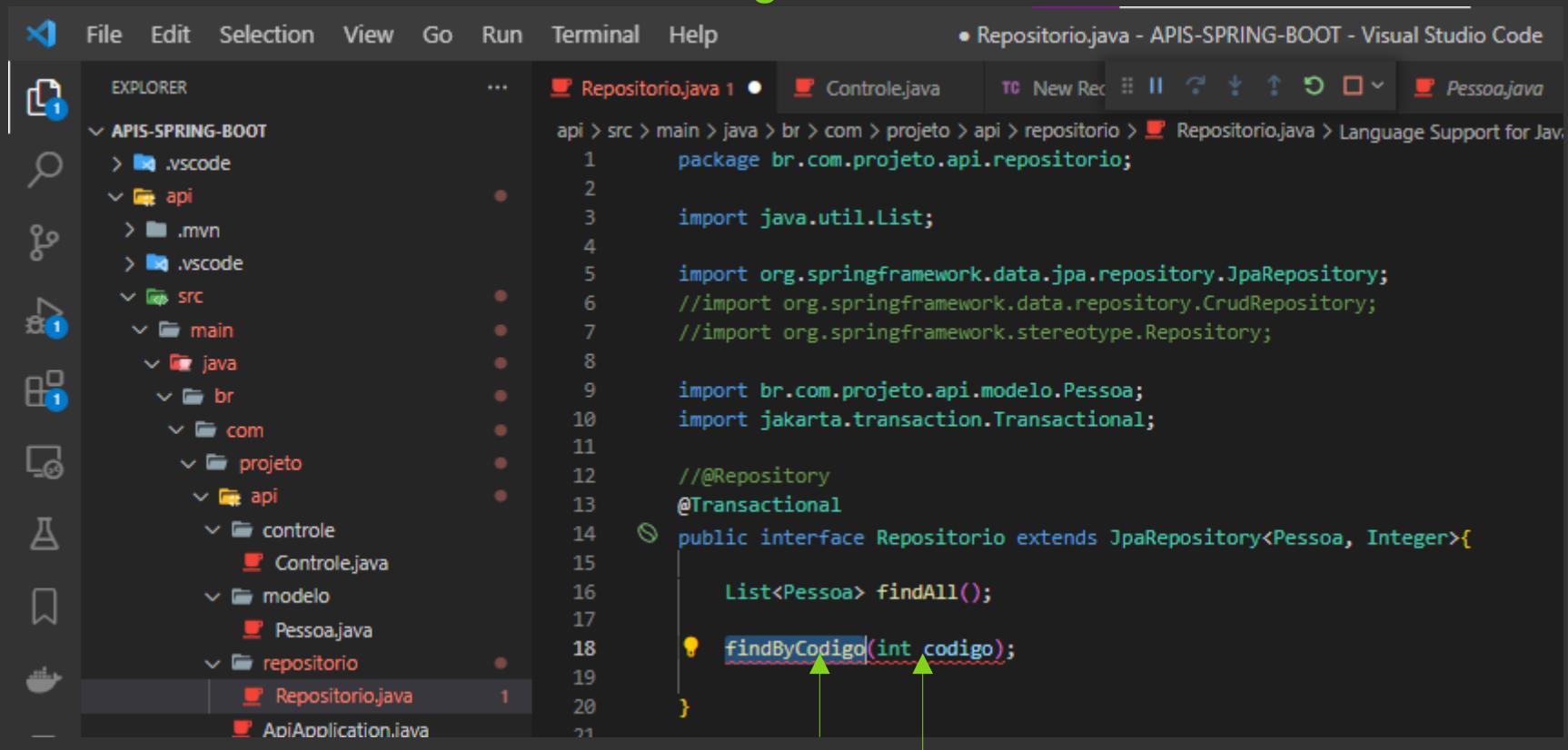
The screenshot shows the VS Code interface with the following details:

- 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:

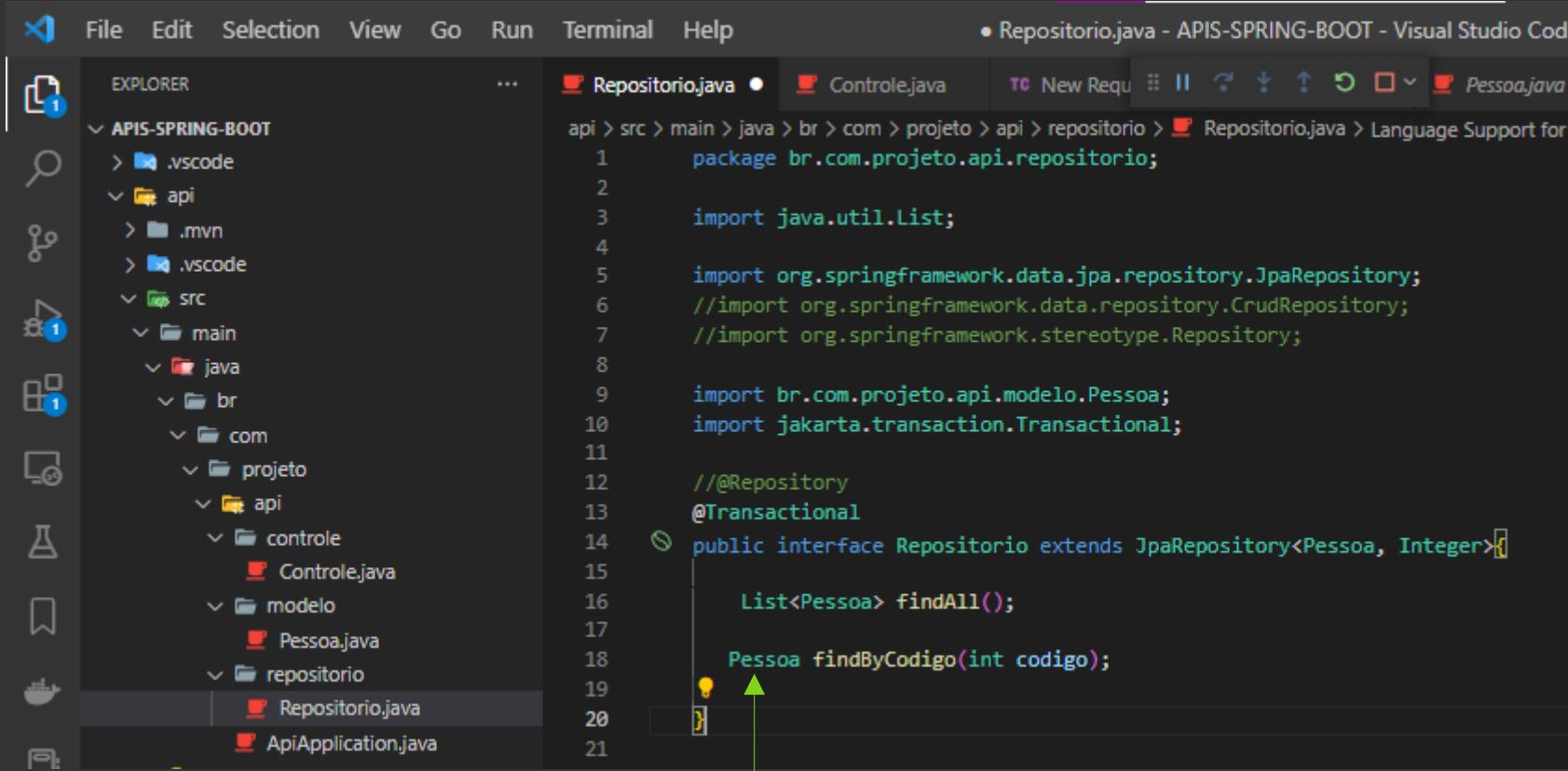
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Repository.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer View:** Shows the project structure:
 - APIS-SPRING-BOOT
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 -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

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

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

Visual Studio Code features: A red squiggly underline under the `findByCodigo` method name, with two green arrows pointing upwards from the underlined text towards the method declaration.

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:

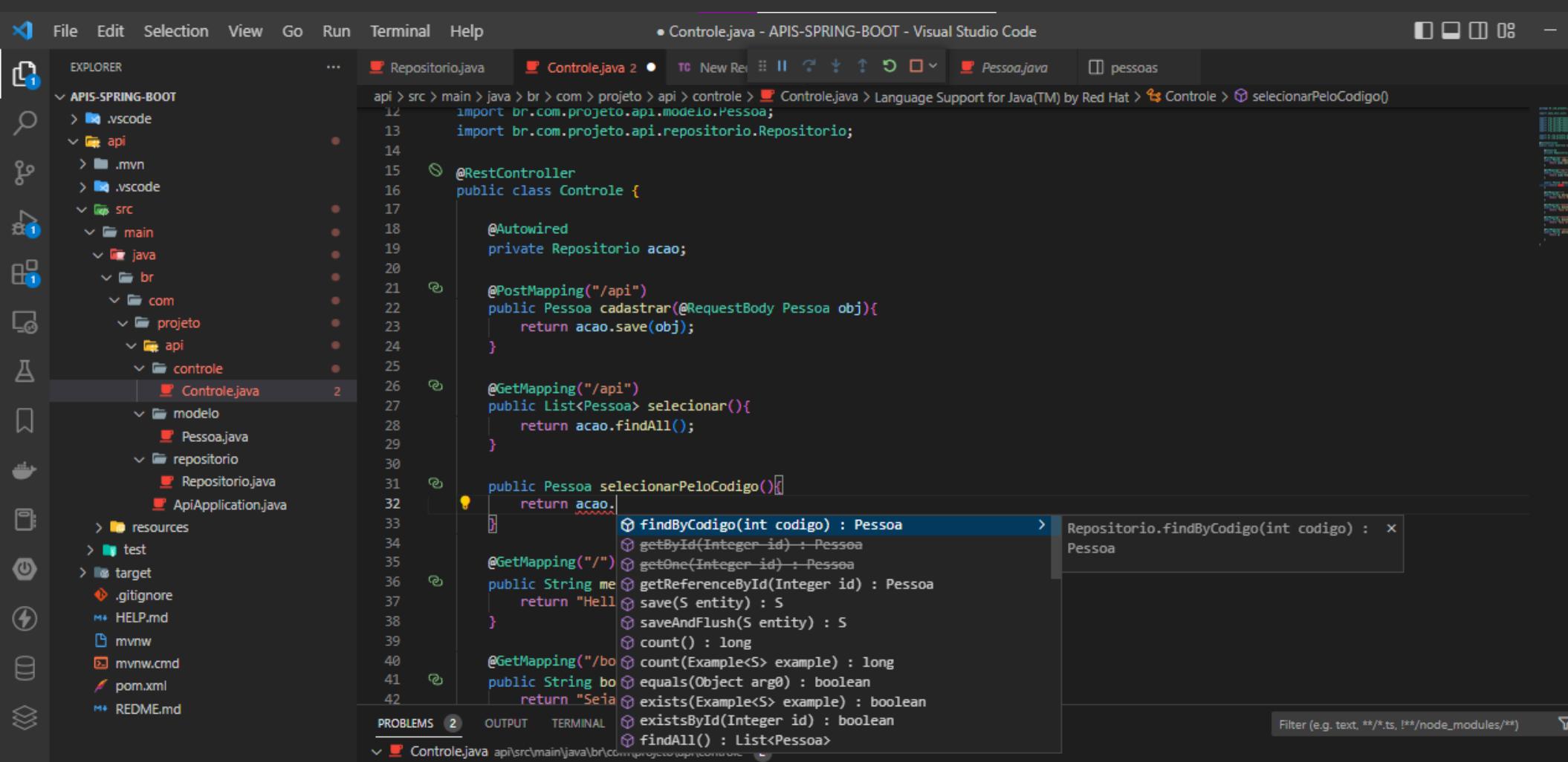
- File Explorer (Left):** Shows the project structure under "APIS-SPRING-BOOT".
 - Root: .vscode, .mvn, .vscode
 - api: .vscode
 - src: .vscode, main
 - java: br
 - com: projeto, api
 - controle: Controle.java
 - modelo: Pessoa.java
 - repositorio: Repositorio.java
 - ApiApplication.java
- Code Editor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



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

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

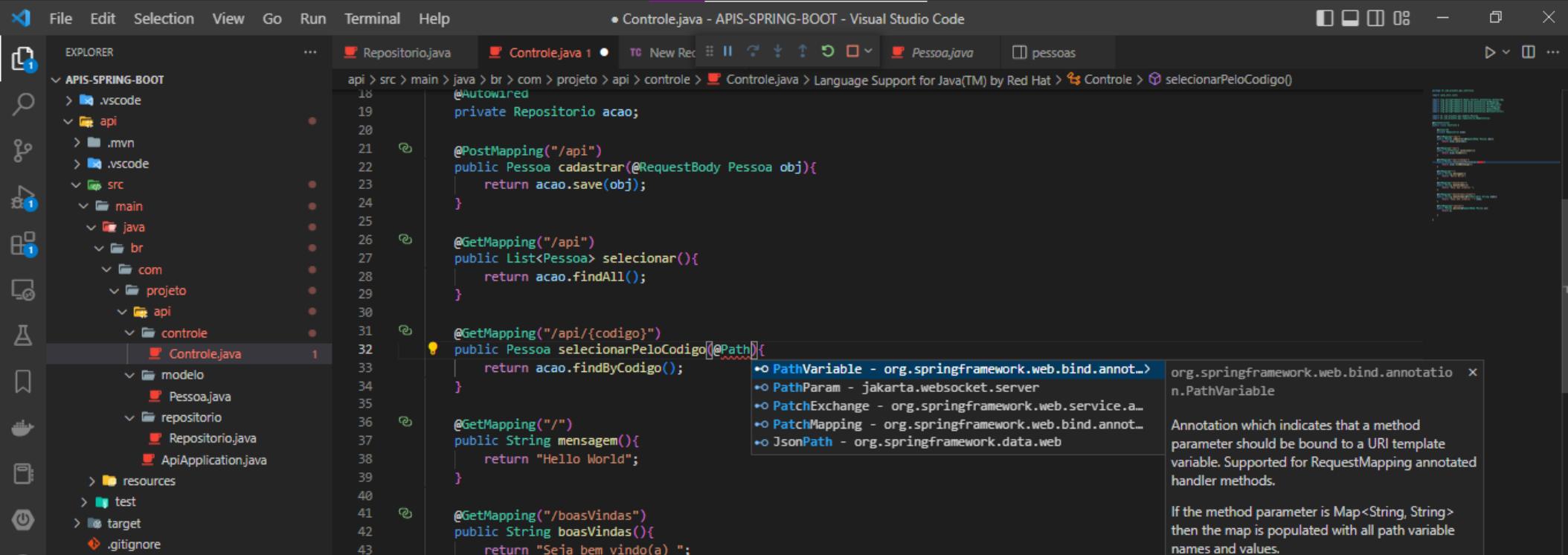
```
12     import br.com.projeto.api.modelo.Pessoa;
13     import br.com.projeto.api.repositorio.Repositorio;
14
15     @RestController
16     public class Controle {
17
18         @Autowired
19         private Repositorio acao;
20
21         @PostMapping("/api")
22         public Pessoa cadastrar(@RequestBody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

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

Como eu pego esse código

Isso é um parametro

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



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

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

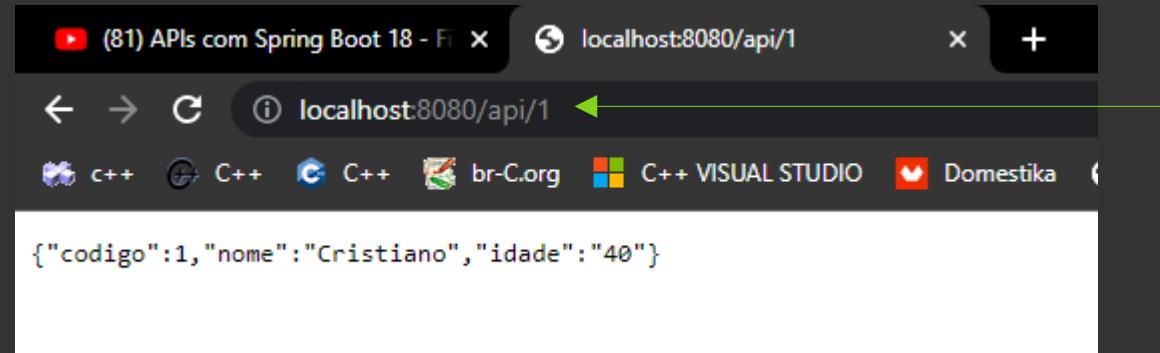
```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30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

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

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

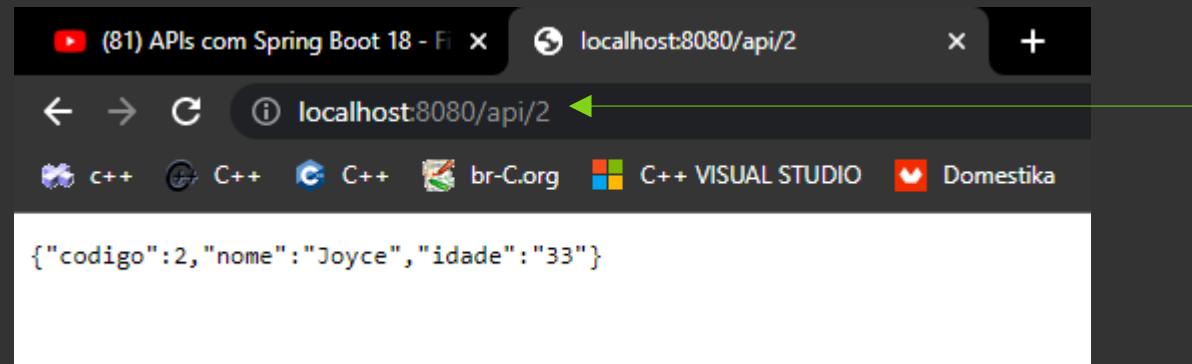
```
18     @Autowired
19     private Repositorio acao;
20
21     @PostMapping("/api")
22     public Pessoa cadastrar(@RequestBody Pessoa obj){
23         return acao.save(obj);
24     }
25
26     @GetMapping("/api")
27     public List<Pessoa> selecionar(){
28         return acao.findAll();
29     }
30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

Salve e va na url testar



A screenshot of a web browser window. The title bar says "(81) APIs com Spring Boot 18 - Fi". 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 definition.

Criamos o metodo

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure for "APIS-SPRING-BOOT".
- Code Editor:** The file "Controle.java" is open. The code defines a class "Controle" with methods for creating, reading, updating, and deleting a "Pessoa" object.
- Tooltips:** A tooltip for the annotation "@PutMapping" is displayed, providing information about its purpose, usage, and related annotations.

```
• Controle.java - APIS-SPRING-BOOT - Visual Studio Code
api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support
18
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/{codigo}")
38     public void alterar(@RequestBody Pessoa obj, @PathVariable int codigo) {
39         acao.update(obj, codigo);
40     }
41
42     @GetMapping("/")
43     public String mensagem(){
```

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

Adicionamos a anotation `@PutMapping`

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

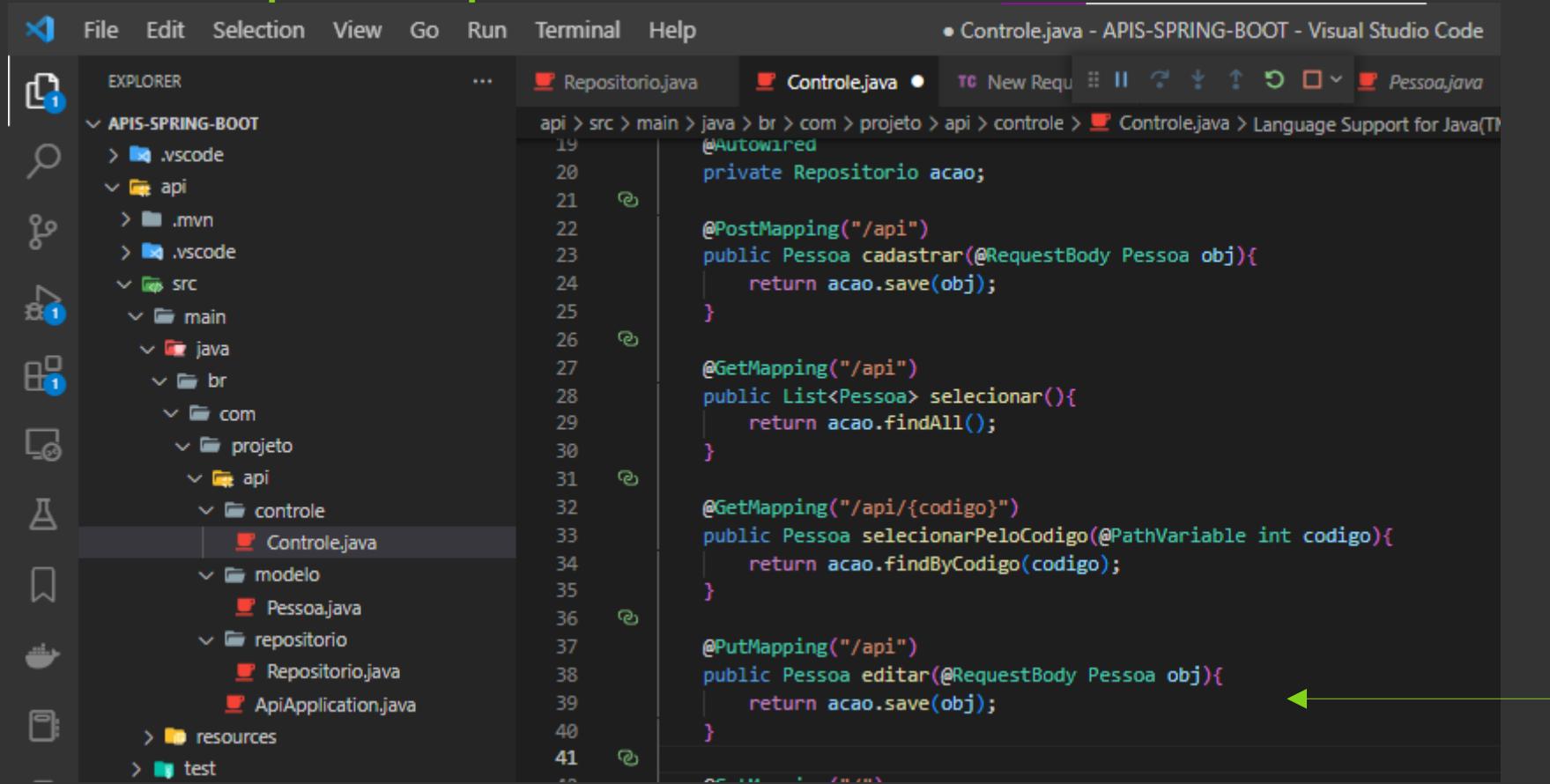
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
- Terminal:** Shows the command "Controle.java - APIS-SPRING-BOOT - Visual Studio Code".
- Code Editor:** Displays Java code for "Controle.java".

```
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



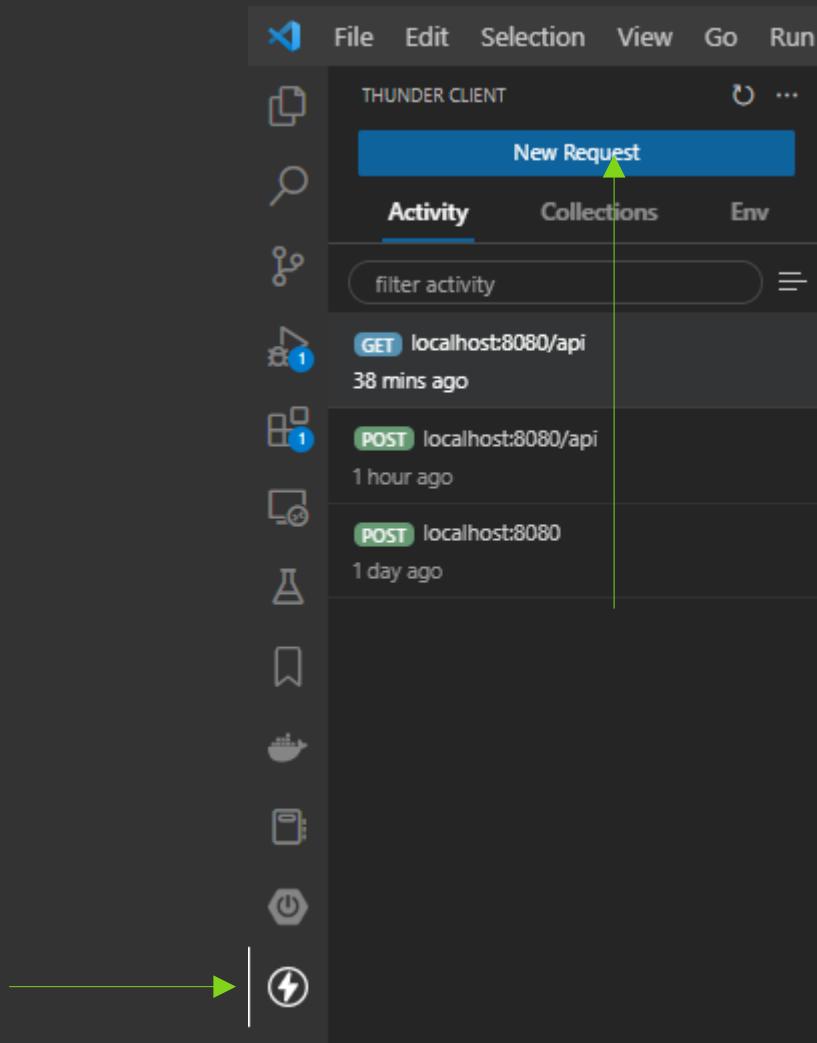
The screenshot shows the Visual Studio Code interface with the following details:

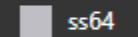
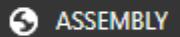
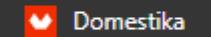
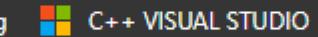
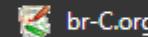
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code.
- Explorer:** Shows the project structure under "APIS-SPRING-BOOT".
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle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

Status: Size: Time:

Response Headers Cookies Results Docs

PUT localhost:8080/api

Send

Query Headers 2 Auth Body Tests Pre Run New

Json Xml Text Form Form-encode Graphql Binary

Format

Json Content

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

The screenshot shows 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
 - Controle.java
 - resources
 - test
 - target
 - .gitignore
 - HELP.md

REPOSITORY

Repository.java

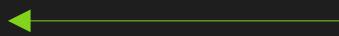
Controle.java

New Request

New Request

Pessoa.java

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

Criamos um metodo remover(){}

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

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

@GetMa
public void DefaultValue() {
}

```

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

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

- Since:
 - 6.0
- Author:
 - Rossen Stoyanchev

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

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal:** Controle.java - APIS-SPRING-BOOT - Visual Studio Code
- Explorer:** Shows the project structure under APIS-SPRING-BOOT:
 - .vscode
 - api
 - .mvn
 - .vscode
 - src
 - main
 - java
 - br
 - com
 - projeto
 - api
 - controle
 - Controle.java
 - modelo
 - Pessoa.java
 - 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 View:** Shows the current file path: api > src > main > java > br > com > projeto > api > controle > Controle.java > Language Support for Java.
- Code Editor View:** Displays the Controle.java code with syntax highlighting and line numbers.

```
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
46     }
47 }
```

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 "/api" endpoint using Spring's REST annotations. It includes methods for creating, reading, updating, and deleting "Pessoa" objects from a repository named "acao".

```
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
```

Esse metodo
retorna um
objeto
pessoa com
os dados

Criamos o objeto

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

PROBLEMS 3 OUTPUT TERMINAL

✓ Controle.java api\src\main\java\bri Syntax error, insert "VariableDe

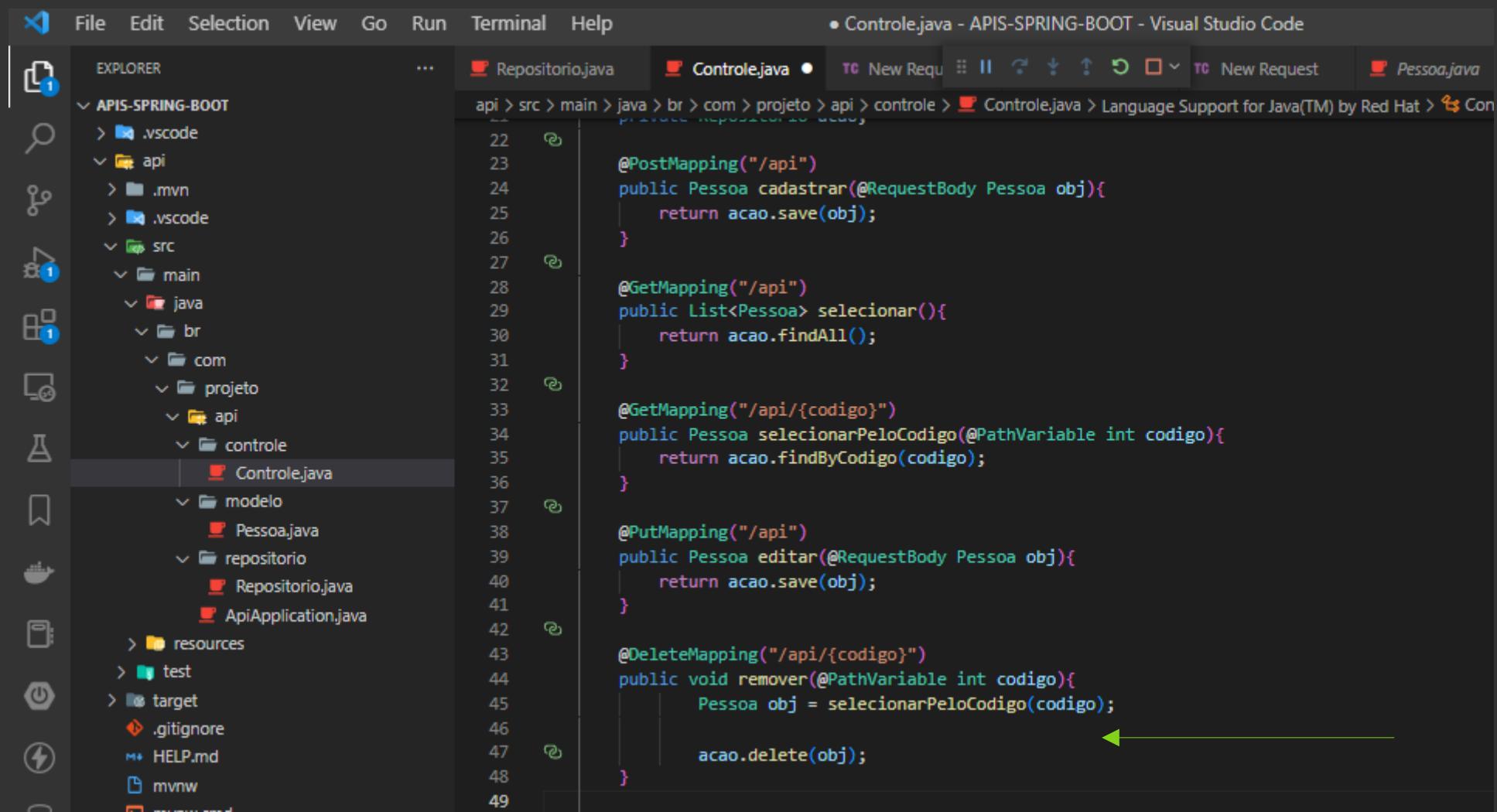
Deletes a given entity.

- **Parameters:**

- **entity** must not be null.

- **Throws:**

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



Salve e va no thunder client e faça uma new request

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

PUT localhost:8080/api
20 mins ago

GET localhost:8080/api
1 hour ago

POST localhost:8080/api
1 hour ago

POST localhost:8080
1 day ago

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

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

Status: Size: Time:

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

parameter value

Send Request Ctrl + Enter

Import Curl Ctrl + U

Change Environment Ctrl + E

Git Sync Details

PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

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

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

Activity:

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

New Request:

Request Type: DELETE

URL: localhost:8080/api/2

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

Query Parameters:

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

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

DEL localhost:8080/api/2 just now

PUT localhost:8080/api 20 mins ago

GET localhost:8080/api 1 hour ago

POST localhost:8080/api 1 hour ago

POST localhost:8080 1 day ago

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

DELETE localhost:8080/api/2 Send

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

Response Headers Cookies Results Docs

1

Query Headers 2 Auth Body Tests Pre Run New

Query Parameters

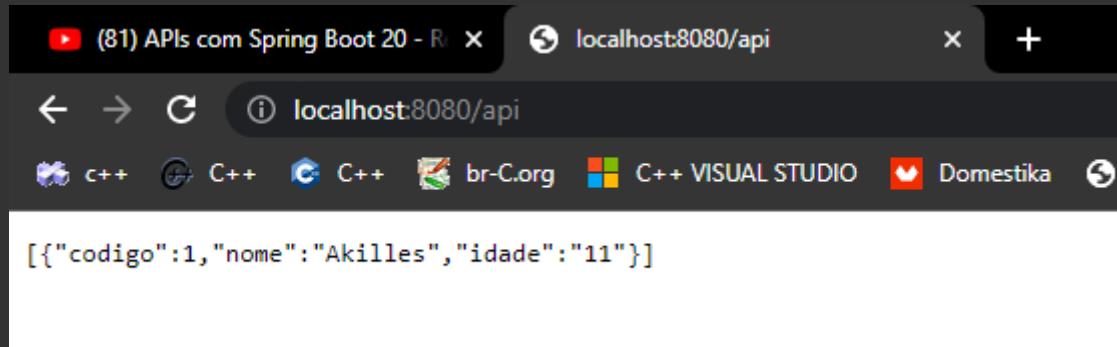
parameter value

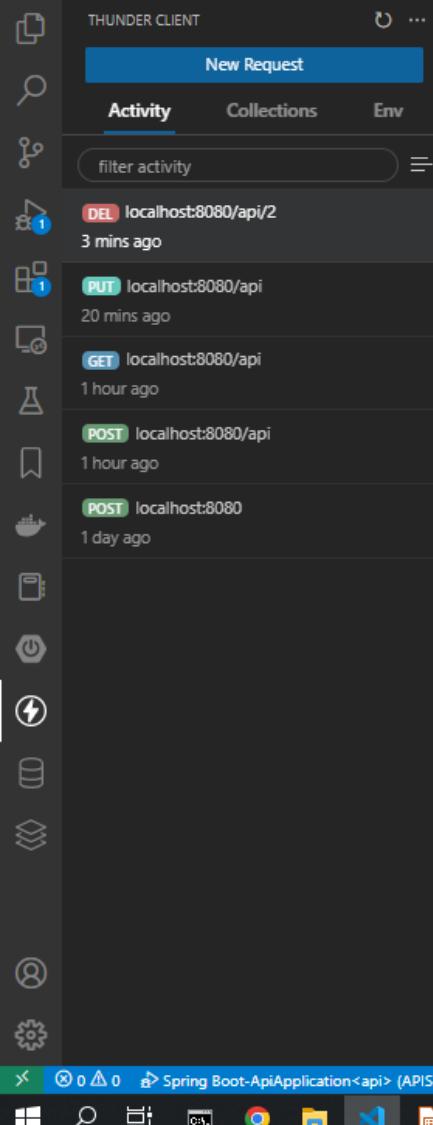
PROBLEMS OUTPUT TERMINAL

No problems have been detected in the workspace.

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

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





The screenshot shows the Postman interface with the following details:

- Request URL:** localhost:8080/api/2
- Method:** DELETE
- Send** button is highlighted.
- Response Headers:** Status: 500 Internal Server Error, Size: 9.81 KB, Time: 181 ms
- Response Body:** A JSON object representing the error. The content is as follows:

```
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  
.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
```
- Query Parameters:** parameter value
- Body:** None
- Tests:** None
- Pre Run:** None
- New:** None

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

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.

