



# ICT Training Center



Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda





# SPRING AI

## GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

---

Simone Scannapieco

Corso base per Venis S.p.A, Venezia, Italia

Settembre 2025



## SETUP PROGETTO SPRING AI

---

- ➔ Inizializzare un progetto Spring con le seguenti caratteristiche:
  - ➔ Maven come *build tool*
  - ➔ Spring Boot alla versione più recente **non SNAPSHOT**
  - ➔ Linguaggio Java 21
  - ➔ Group **it.venis.ai.spring**
  - ➔ Artifact **demo**
  - ➔ jar packaging
  - ➔ Spring Web e OpenAI come dipendenze

- 1** Inizializzare un nuovo progetto Spring AI
- 2** Verifica JRE per progetto
- 3** Esecuzione del progetto

- 1 Da IDE (VSCode)
- 2 Da Spring Initializr

# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VS CODE)

### 1 Installare le estensioni VSC necessarie

The screenshot shows the Visual Studio Code interface with the 'Marketplace' tab selected. A search bar at the top contains the query 'Extension: Extension Pack for Java'. Below the search bar, the 'Extension Pack for Java' by Microsoft is displayed, showing a rating of 4.5 stars and 37,832 reviews. An 'Install' button is visible. The main content area displays seven extensions included in the pack, each with its name, developer, description, and an 'Install' button:

- Language Support for Java™ by Red Hat
- Debugger for Java
- Test Runner for Java
- Maven for Java
- Gradle for Java
- Project Manager for Java

On the right side of the screen, there are sections for 'Marketplace' and 'Categories'. The 'Marketplace' section shows details for the extension: Identifier: vscjava.vscode-java-pack, Version: 0.29.2, Published: 2017-09-27, Last Released: 2025-08-18. The 'Categories' section lists Programming Languages, Snippets, Linters, Debuggers, and Formatters under the 'Extension Packs' category.

**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.

**Extensions Included**

By installing Extension Pack for Java, the following extensions are installed:

- Language Support for Java™ by Red Hat

**Resources**

Marketplace  
Issues  
Repository  
License  
Microsoft

**Finish Setup**

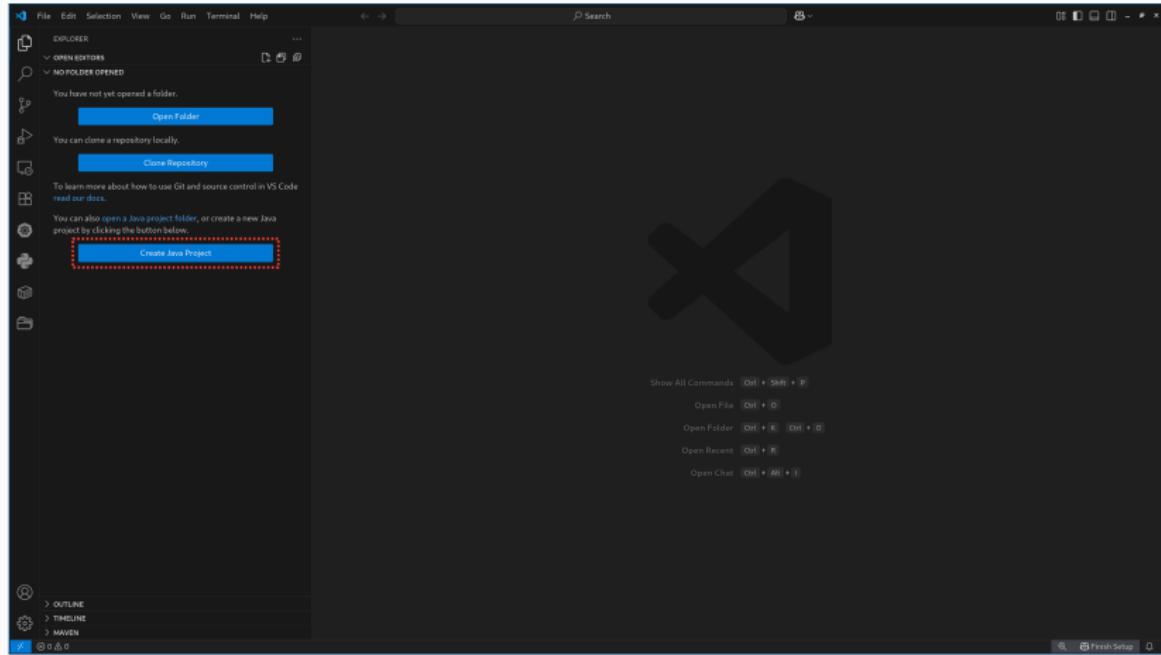
### 1 Installare le estensioni VSC necessarie

The screenshot shows the VS Code Marketplace interface. The main search bar at the top contains the text "Extension: Spring Boot Extension Pack". Below the search bar, the "Spring Boot Extension Pack" by VMware is displayed. It has a green gear icon, a rating of 4.5 stars from 21 reviews, and 3,563,390 installs. A blue "Install" button with a checkmark and the text "Auto Update" are present. A note below says, "This extension is recommended based on the files you recently opened." On the left sidebar, there are icons for file operations like Open, Save, Find, and others. The right sidebar contains sections for "Marketplace" (Identifier: v mware.vscode-boot-dev-pack, Version: 0.2.2, Published: 2017-11-28, Last Released: 2025-01-28), "Categories" (Programming Languages, Linters, Extension Packs), and "Resources" (Marketplace, Issues, Repository, License, VMware). The main content area displays the "VS Code Spring Boot Application Development Extension Pack" page, which includes sections for "Spring Boot" and "Spring Initializr Java", each with their own descriptions and links.

# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VS CODE)

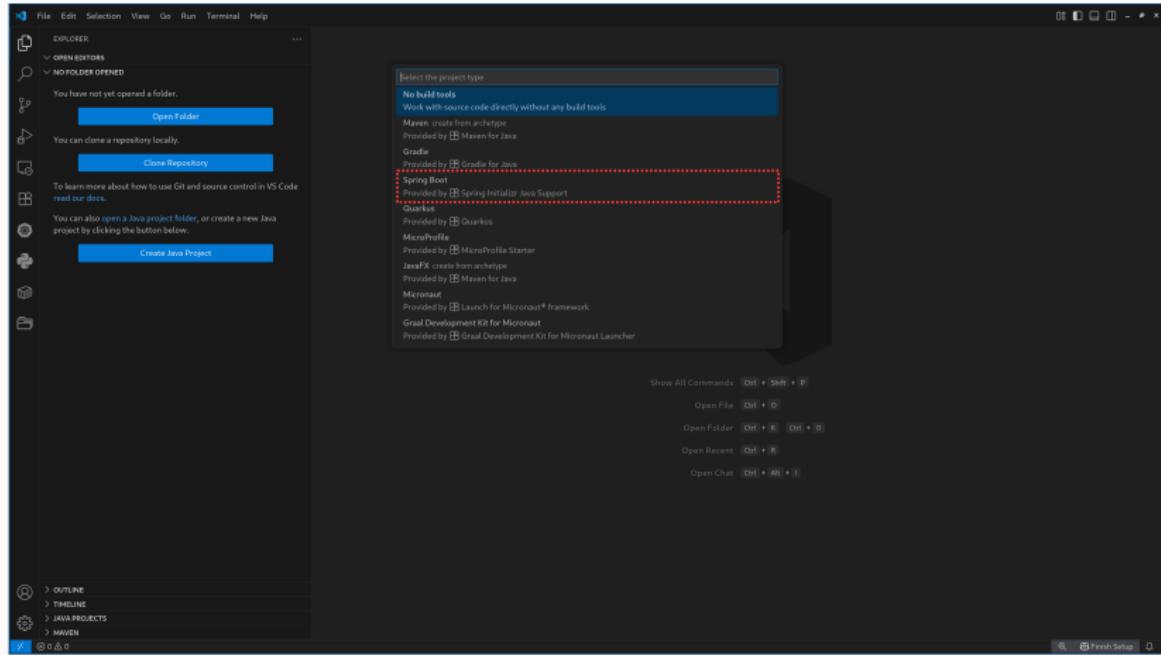
### 2 Aprire una nuova finestra VSC e selezionare Create Java Project



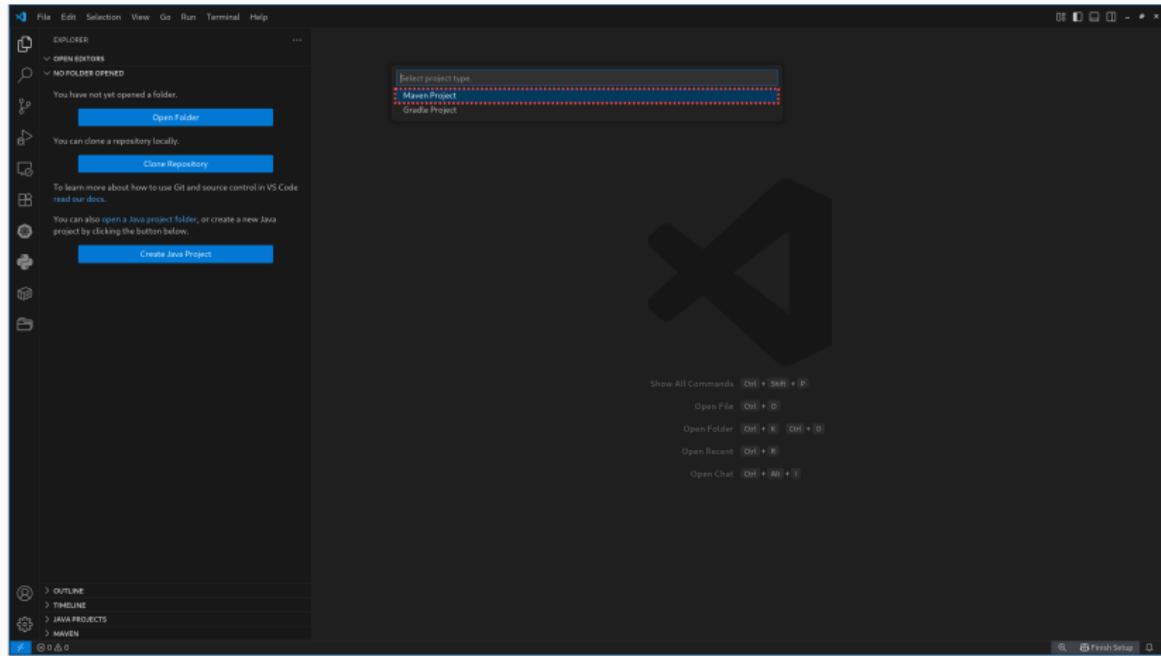
# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VS CODE)

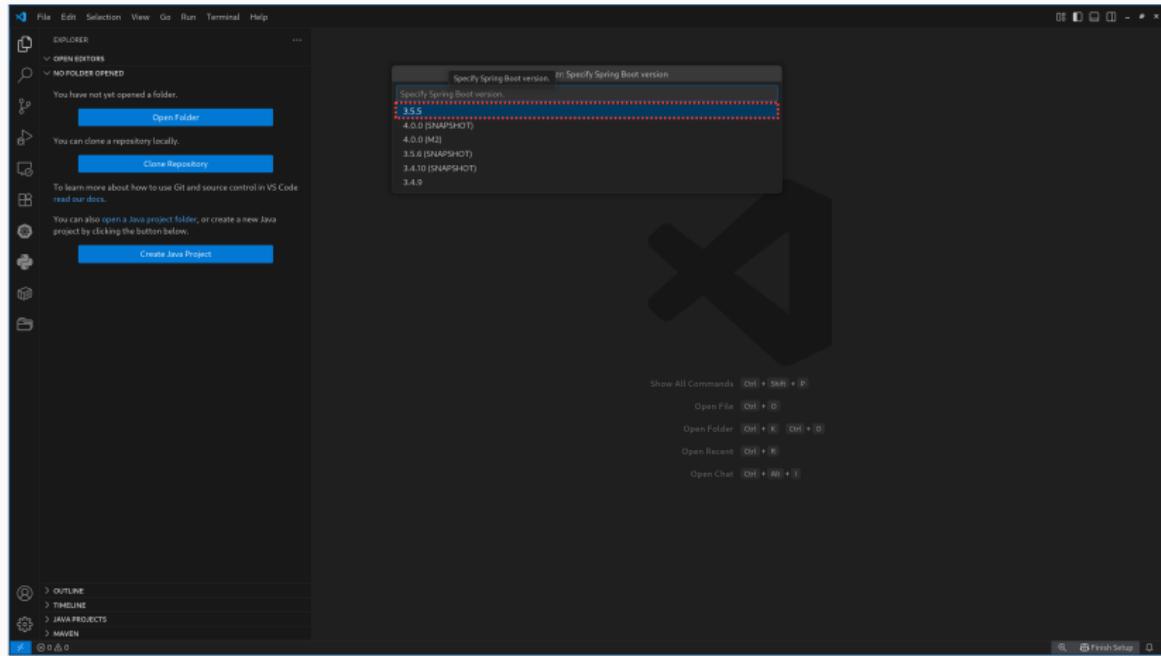
### 3 Selezionare Spring Boot come tipo di progetto (NON Maven)



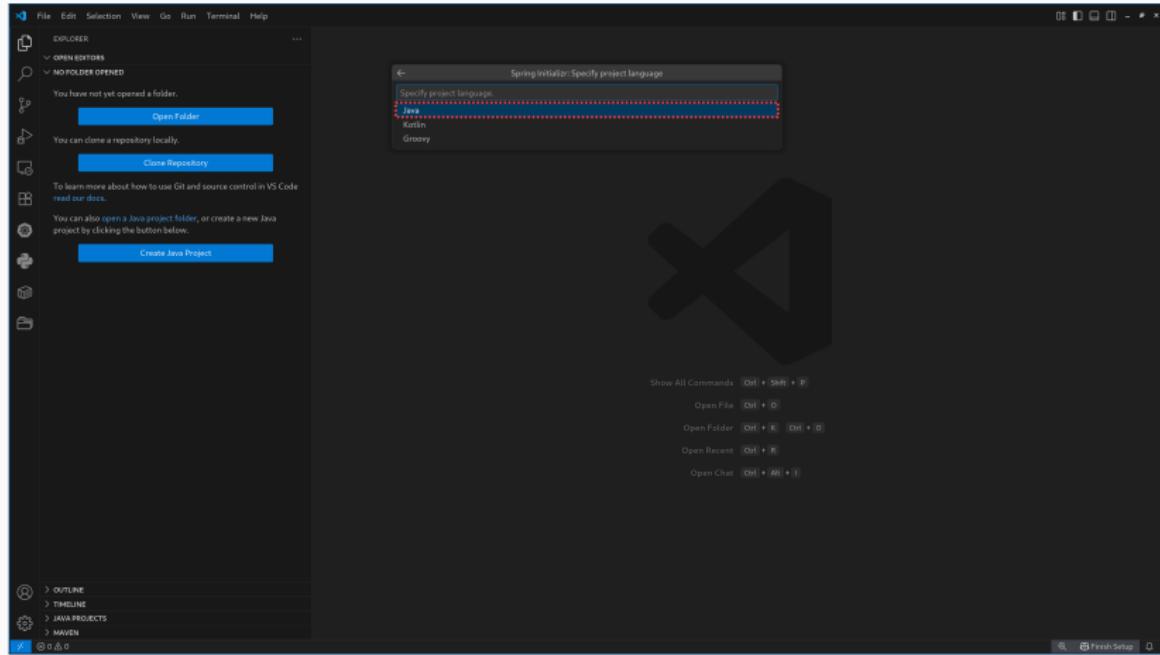
### 4 Selezionare Maven project come tipo di progetto



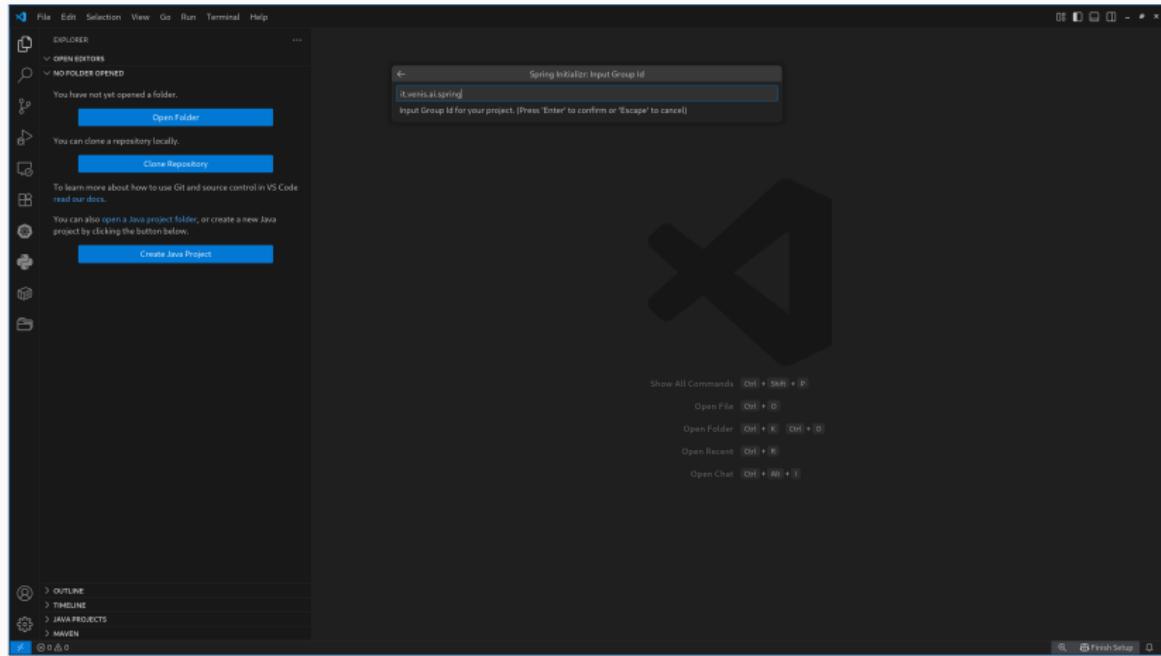
### 5 Selezionare la versione di Boot per il progetto



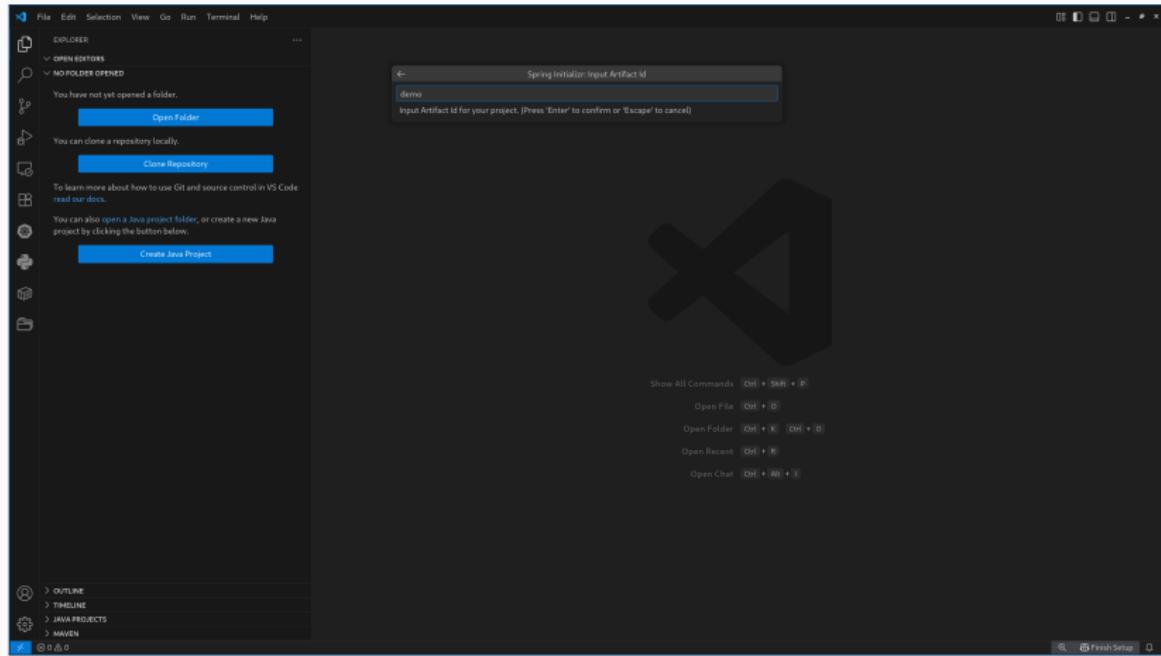
### 6 Selezionare Java come linguaggio di progetto



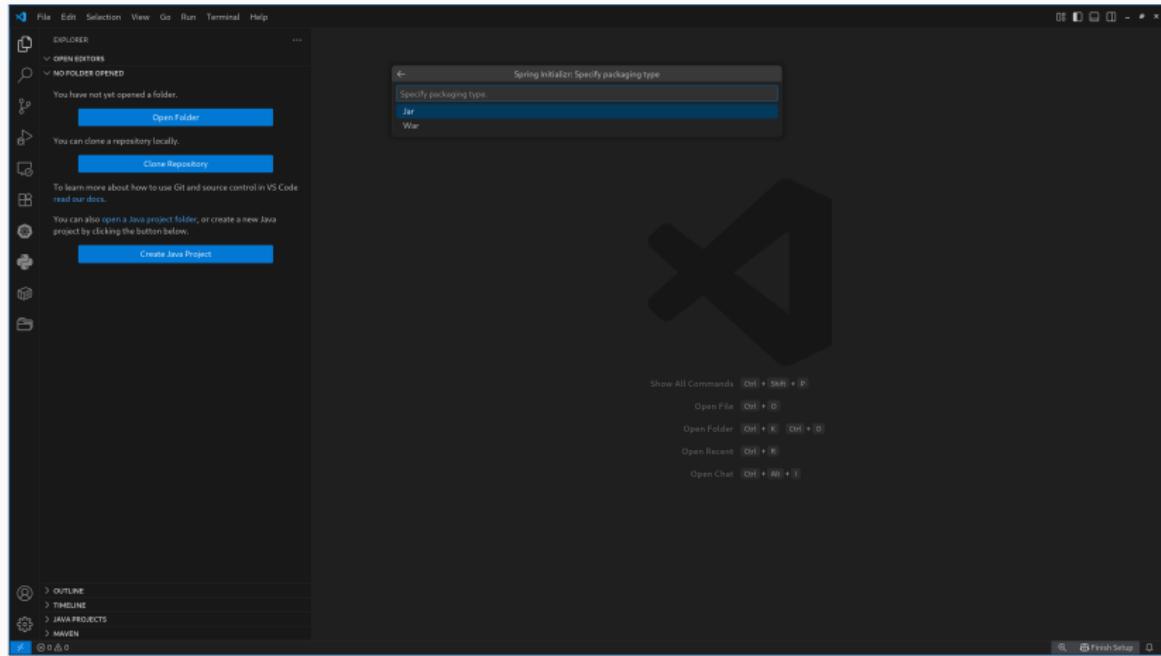
### 7 Definire il Group ID di riferimento del progetto



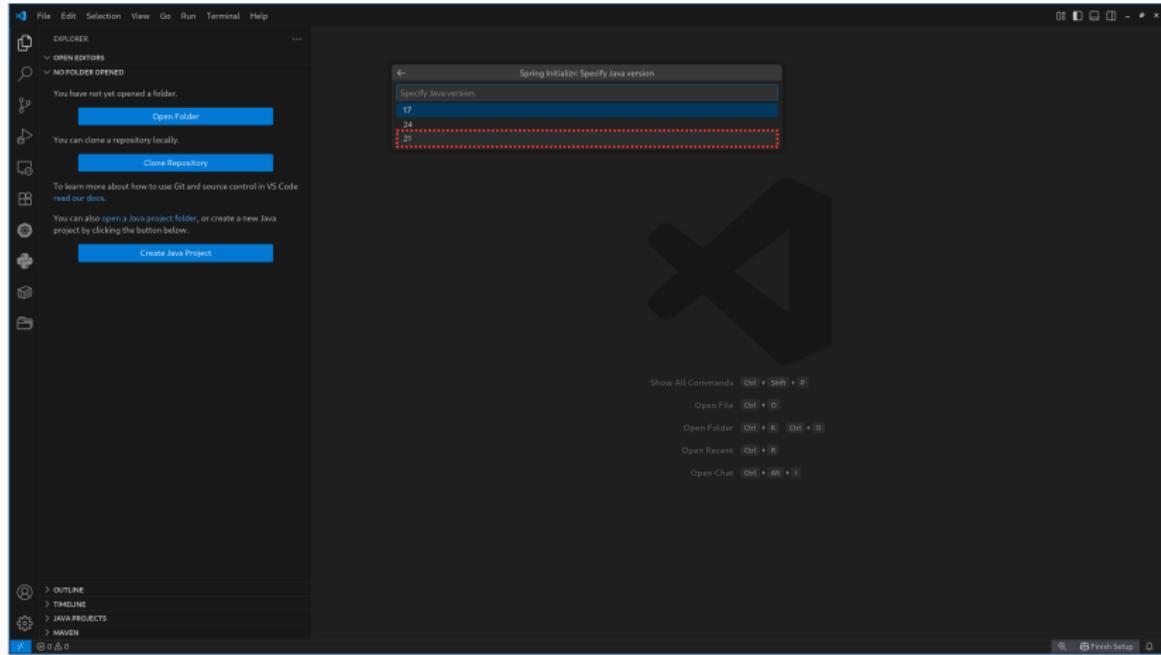
### 8 Definire l'Artifact ID di riferimento del progetto



### 9 Definire la tipologia di *packaging* del progetto



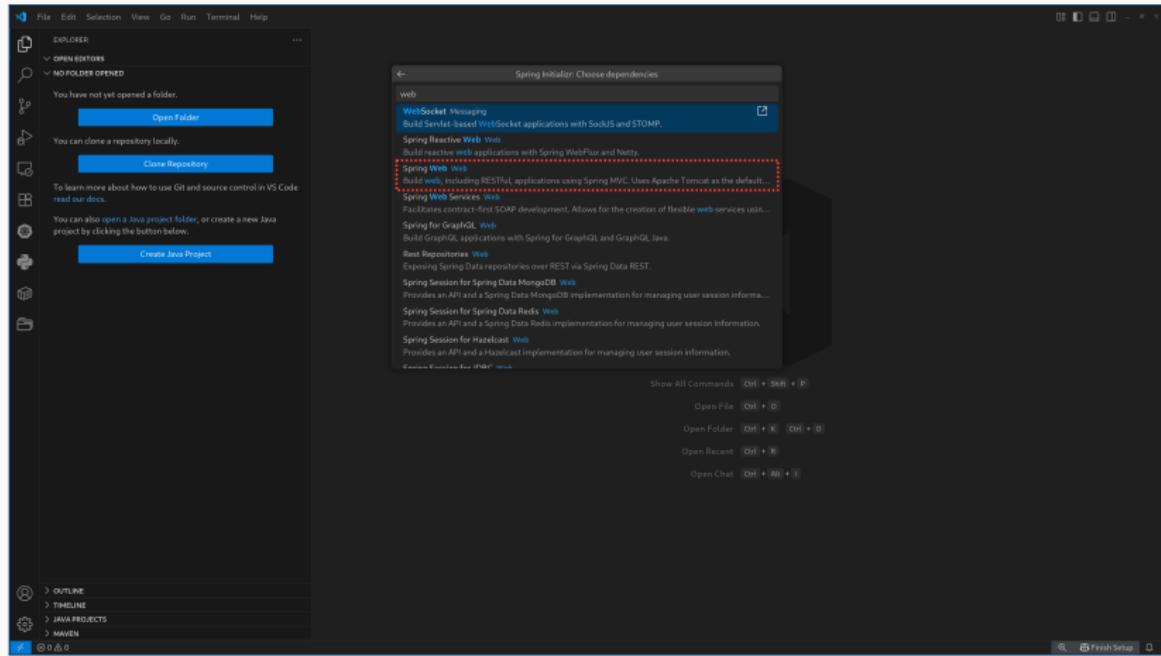
### 10 Selezionare la versione di Java di riferimento



# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VSCode)

### 11 Selezionare come prima dipendenza di progetto Spring Web

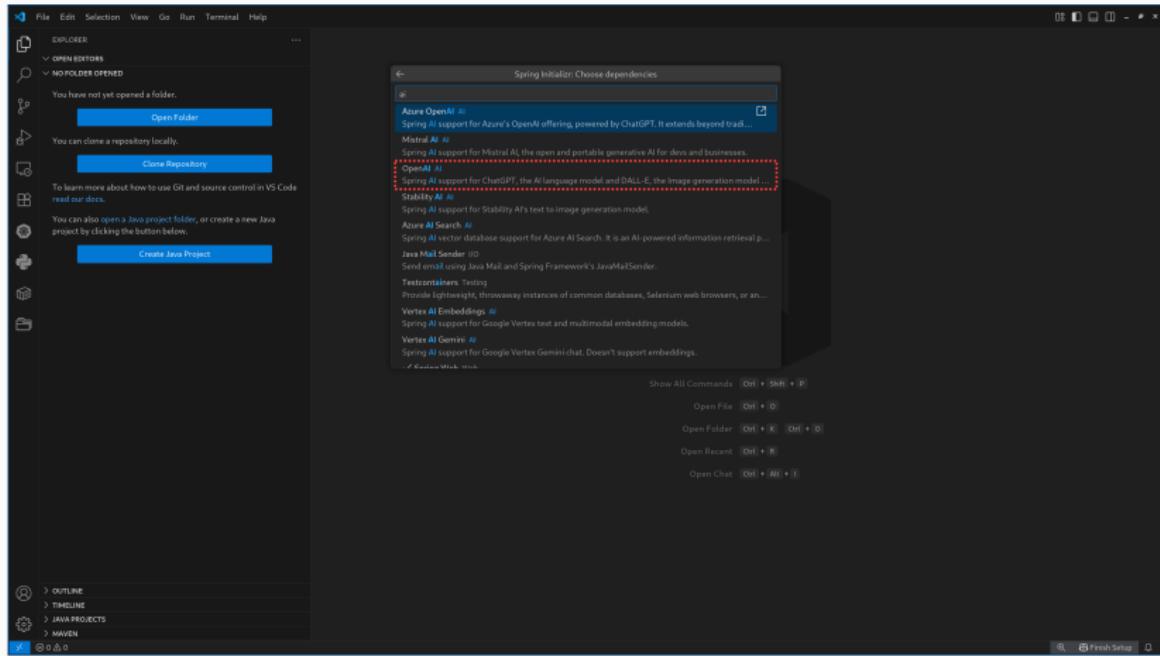


# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VSCode)

12

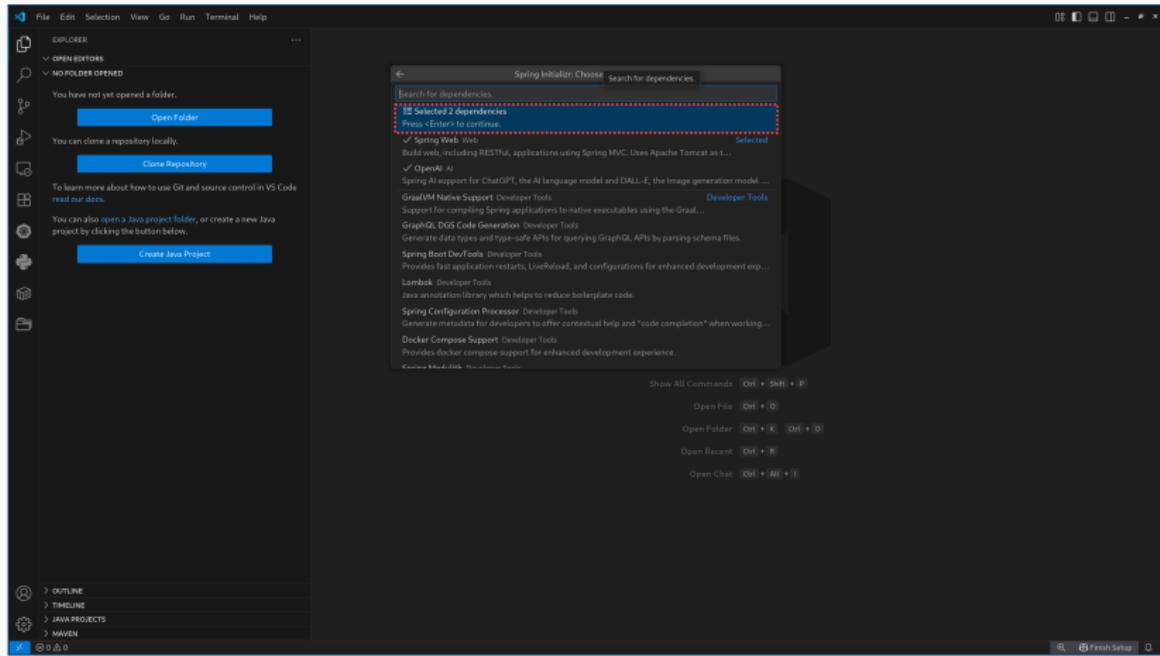
### Selezionare come seconda dipendenza di progetto OpenAI



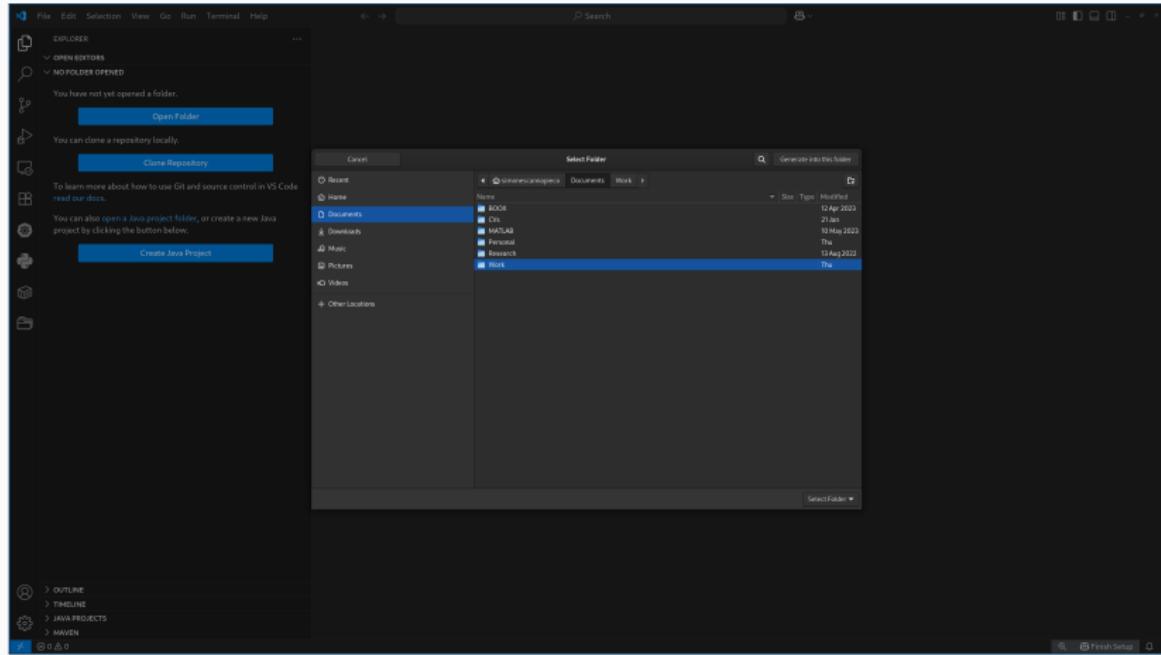
# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VSCode)

### 13 Continuare la creazione del progetto premendo Enter



### 14 Navigare nel FS e selezionare la cartella padre del progetto

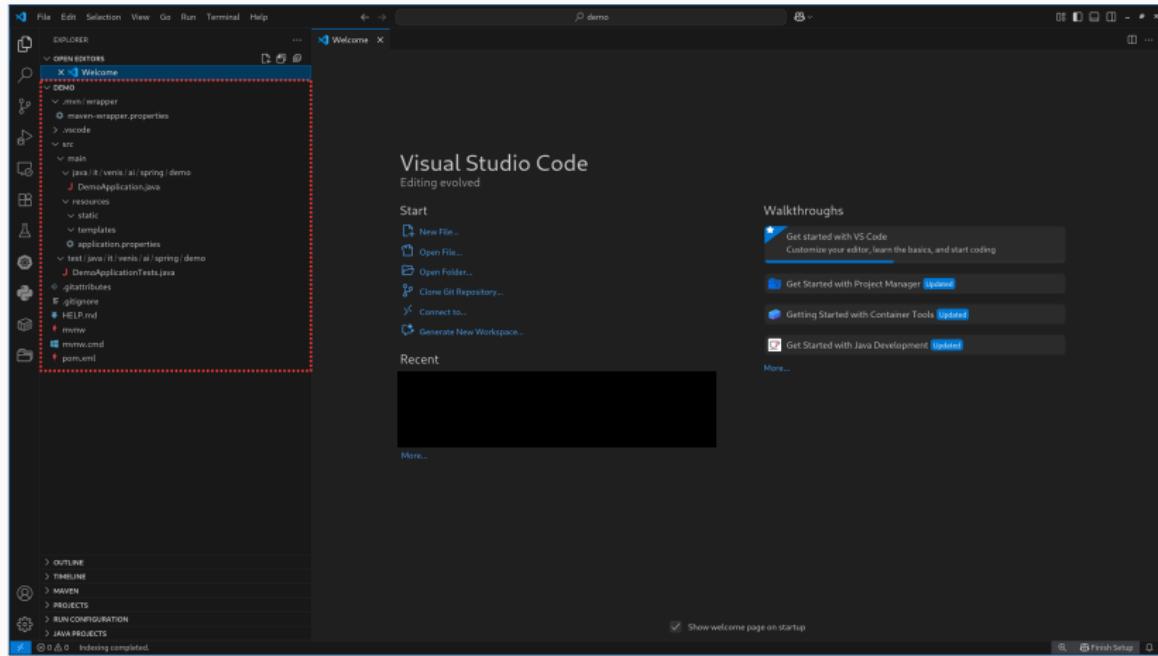


# PROGETTO SPRING AI

## INIZIALIZZAZIONE DA IDE (VS CODE)

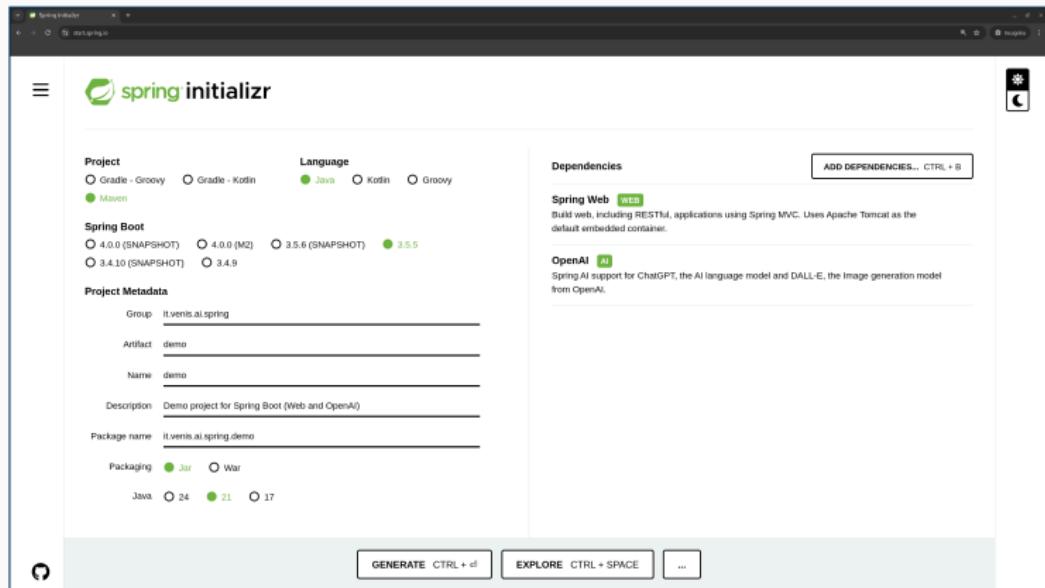
15

Verificare i file di progetto nel pannello Explorer (CTRL + SHIFT + E)



# PROGETTO SPRING

## INIZIALIZZAZIONE DA SPRING INITIALizr

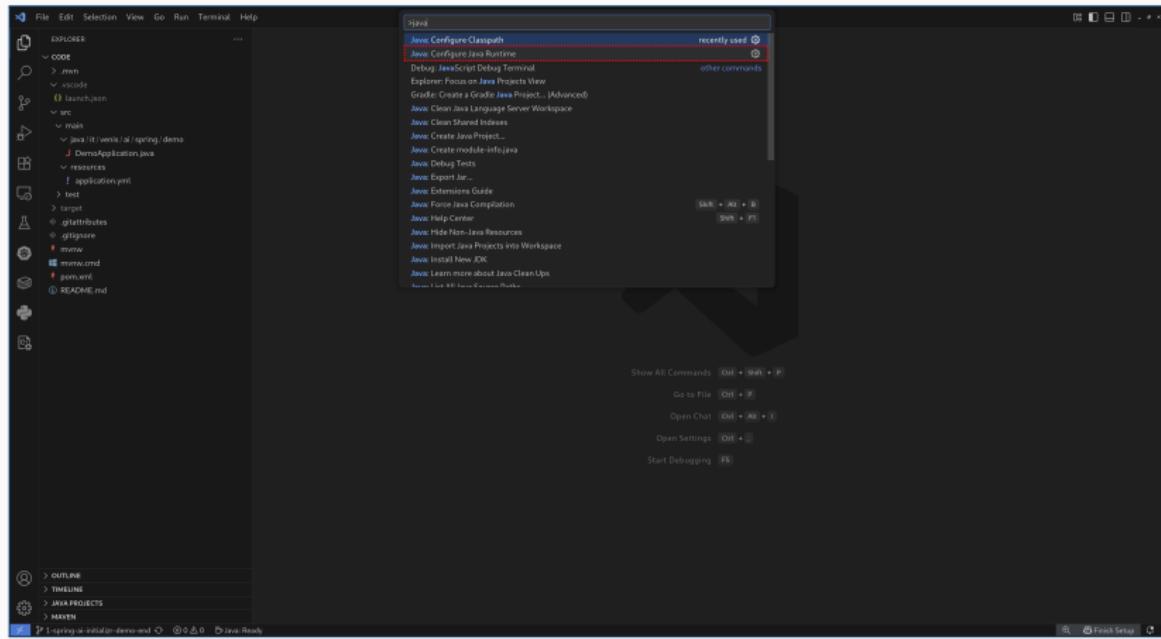


The screenshot shows the Spring Initializr interface. In the Project section, Maven is selected. Language Java is chosen. Under Spring Boot, version 3.5.6 (Snapshot) is selected. In the Project Metadata section, Group is set to it.venis.ai.spring, Artifact to demo, Name to demo, Description to Demo project for Spring Boot (Web and OpenAI), and Package name to it.venis.ai.spring.demo. Packaging is set to Jar. Java version 21 is selected. On the right, the Dependencies section shows Spring Web (selected) and OpenAI (selected). Buttons at the bottom include GENERATE (CTRL + d) and EXPLORE (CTRL + SPACE).

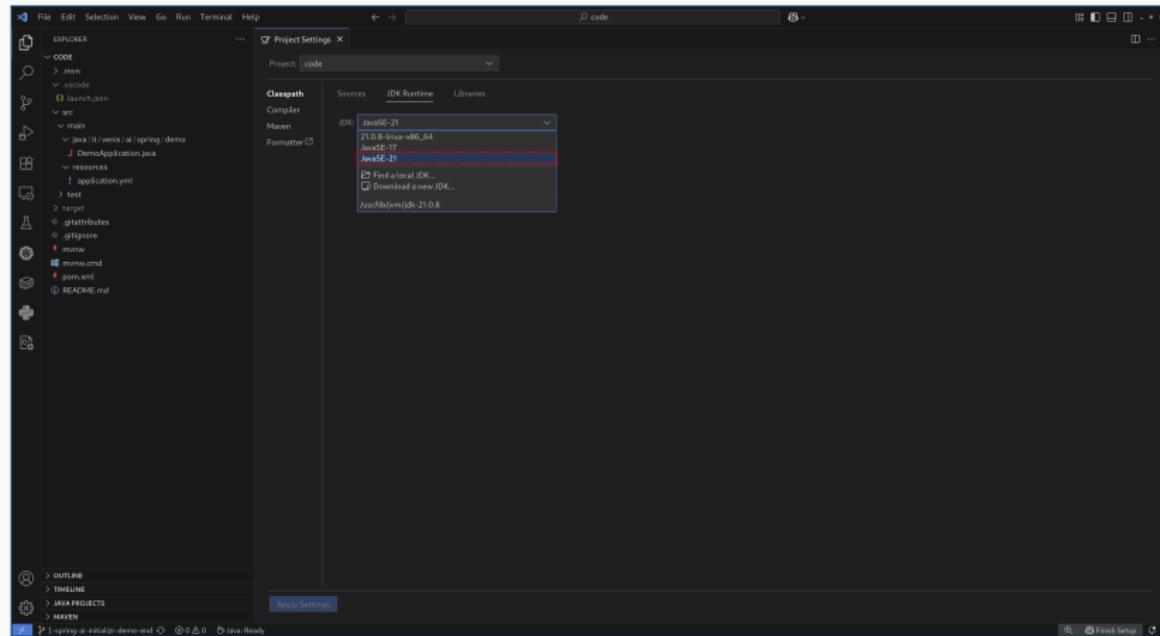
## Portale ufficiale

<https://start.spring.io/>

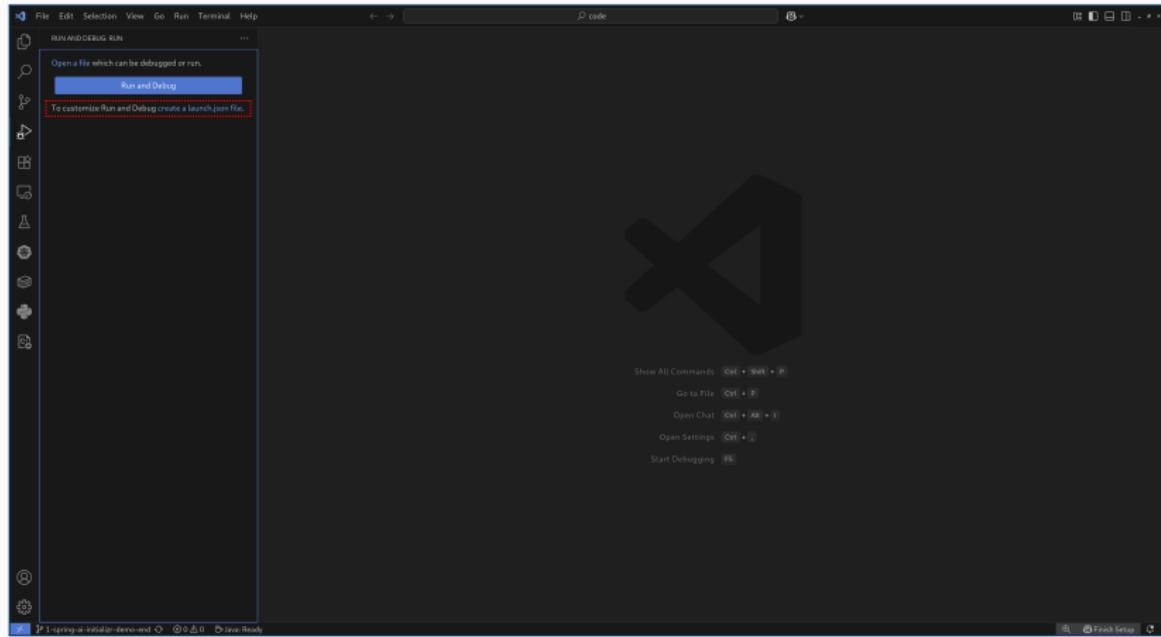
### 1 Dal pannello dei comandi (CTRL + SHIFT + P), Java: Configure Java Runtime



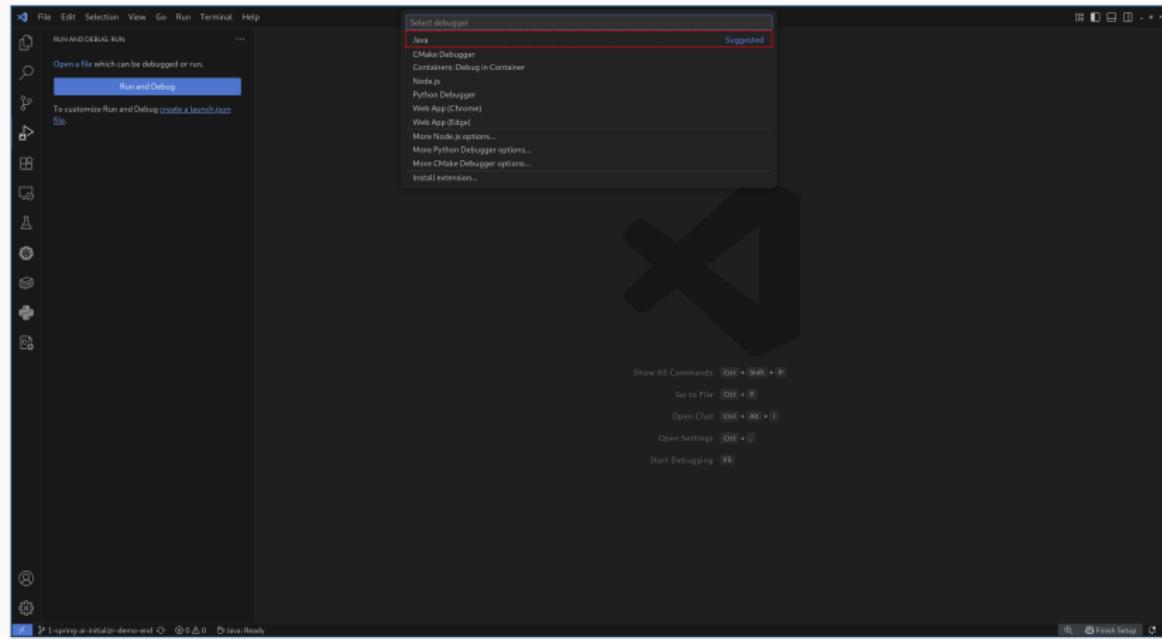
### 2 Selezionare la JDK 21 precedentemente installata



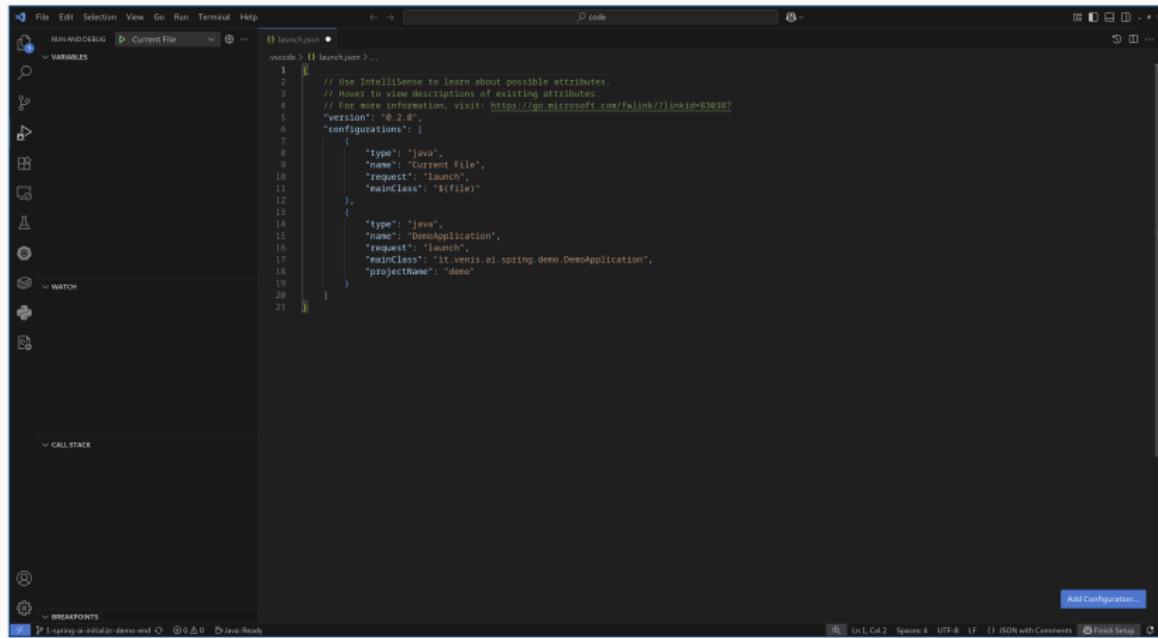
- 1 Dal pannello di debug (**CTRL + SHIFT + D**), creare un nuovo file `launch.json`



### 2 Selezionare Java come tipologia di esecuzione/debug



### 3 Verificare la presenza delle due configurazioni di base

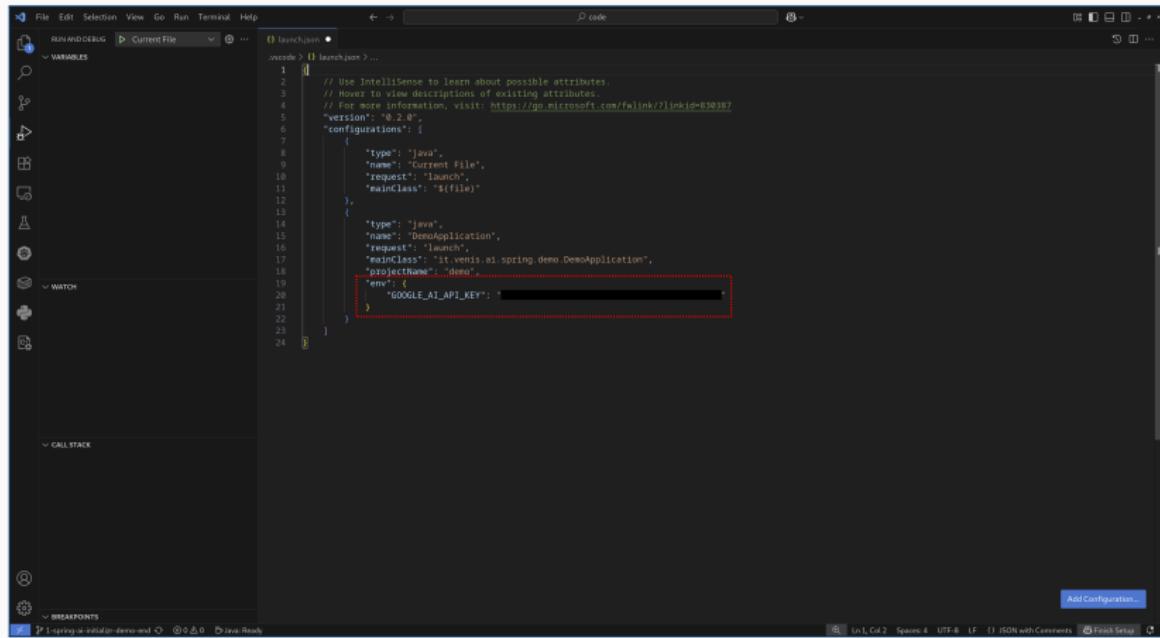


The screenshot shows the VS Code interface with the "RUN AND DEBUG" tab selected. On the left, the "VARIABLES" and "WATCH" toolbars are visible. The main editor area displays the contents of the `launch.json` file:

```
version: "0.2.0"
"configurations": [
    {
        "type": "java",
        "name": "Current File",
        "request": "launch",
        "mainClass": "${file}"
    },
    {
        "type": "java",
        "name": "DemoApplication",
        "request": "launch",
        "mainClass": "it.venis.ai.spring.demo.DemoApplication",
        "projectName": "demo"
    }
]
```

The status bar at the bottom indicates the file is `Spring AI - Corso base`, has 0 errors, and is in "Run Ready" mode. A "Breakpoints" section is also visible in the bottom-left corner.

### 4 Aggiungere metadati se necessario (es. variabili di ambiente)

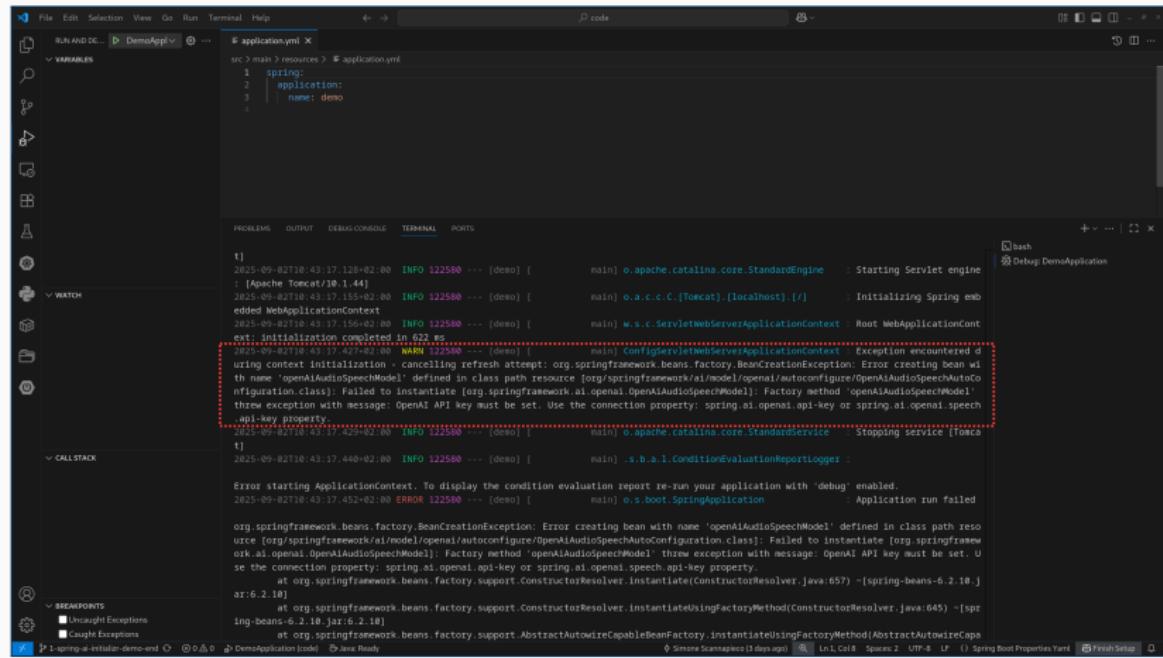


The screenshot shows the VS Code interface with the 'RUN AND DEBUG' tab selected. The left sidebar displays the 'VARIABLES' and 'WATCH' sections. The main editor area shows the 'launch.json' configuration file:

```
1 // Use IntelliSense to learn about possible attributes.
2 // Hover to view descriptions of existing attributes.
3 // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
4 {
5     "version": "0.2.0",
6     "configurations": [
7         {
8             "type": "java",
9             "name": "Current File",
10            "request": "launch",
11            "mainClass": "${file}"
12        },
13        {
14            "type": "java",
15            "name": "DemoApplication",
16            "request": "launch",
17            "mainClass": "it.venis.ai.spring.demo.DemoApplication",
18            "projectName": "demo",
19            "env": {
20                "GOOGLE_AI_API_KEY": "████████████████████████████████████████"
21            }
22        }
23    ]
24 }
```

A red dotted rectangle highlights the 'env' object in the second configuration block, specifically the 'GOOGLE\_AI\_API\_KEY' key. The bottom right corner of the editor has a 'Finish Setup' button.

## 1 Eseguendo il progetto, dovremmo ottenere una eccezione!



```
File Edit Selection View Go Run Terminal Help
RUN AND DE... DemoAplic... Variables PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash Debug DemoApplication

application.yml X
src/main/resources/application.yml
1 spring:
2   | application:
3   |   name: demo
4

t]
2025-09-02T10:43:17.128+02:00 INFO 122500 --- [demo] (main) o.apache.catalina.core.StandardEngine : Starting Servlet engine
: Apache Tomcat/10.1.44
2025-09-02T10:43:17.155+02:00 INFO 122500 --- [demo] (main) o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-09-02T10:43:17.156+02:00 INFO 122500 --- [demo] (main) w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext
ext initialization completed in 622 ms
2025-09-02T10:43:42+02:00 WARN 122500 --- [demo] (main) ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'openAiAudioSpeechModel' defined in class path resource [org/springframework/framework/el/module/openai/autoload/AudioSpeechAutoConfiguration.class]: Failed to instantiate [org.springframework.ai.openai.OpenAIAudioSpeechModel]: Factory method 'openAiAudioSpeechModel' threw exception with message: OpenAI API key must be set. Use the connection property: spring.ai.openai.api-key or spring.ai.openai.speech.api-key property
2025-09-02T10:43:42+02:00 INFO 122500 --- [demo] (main) o.apache.catalina.core.StandardService : Stopping service [Tomcat]
t]
2025-09-02T10:43:17.429+02:00 INFO 122500 --- [demo] (main) .s.b.a.l.ConditionEvaluationReportLogger :
ERROR starting ApplicationContext. To resolve the condition evaluation report re-run your application with 'debug' enabled.
2025-09-02T10:43:17.452+02:00 ERROR 122500 --- [demo] (main) o.s.boot.SpringApplication : Application run failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'openAiAudioSpeechModel' defined in class path resource [org/springframework/framework/el/module/openai/autoload/AudioSpeechAutoConfiguration.class]: Failed to instantiate [org.springframework.ai.openai.OpenAIAudioSpeechModel]: Factory method 'openAiAudioSpeechModel' threw exception with message: OpenAI API key must be set. Use the connection property: spring.ai.openai.api-key or spring.ai.openai.speech.api-key property
at org.springframework.beans.factory.support.ConstructorResolver.resolveConstr...
xr6.2.18)
at org.springframework.beans.factory.support.ConstructorResolver.instantiateUsingFactoryMethod(ConstructorResolver.java:657) ~[spring-beans-6.2.18.j...
xr6.2.18)
at org.springframework.beans.factory.support.ConstructorResolver.instantiate(ConstructorResolver.java:637) ~[sprin...
ing-beans-6.2.18.jar:6.2.18]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.instantiateUsingFactoryMethod(AbstractAutowireCapa...
Simone Scannapieco 1-spring-ai-initial-demo-and 0 0 0 0 DemoApplication [code] Java Ready 0 Since Scanapieco (3 days ago) 0 Unl. Cola 0 Spazio 2 UTP-S 1P (...) Spring Boot Properties Yardi Final Setup
```

## 2 Definiamo la chiave richiesta da Spring AI (spring.ai.openai.api-key)

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

- IDE Layout:** The left sidebar includes "RUN AND DE...", "DemoApp", "VARIABLES", "WATCH", and "CALL STACK". The right sidebar includes "bash" and "Debug DemoApplication".
- application.yml Content:**

```
spring:
  application:
    name: demo
  openai:
    api-key: ${GOOGLE_AI_API_KEY}
```

A red dashed box highlights the line `api-key: ${GOOGLE_AI_API_KEY}`.
- Terminal Log Output:**

```
t]
2025-09-02T10:43:17.128+02:00 INFO 122500 --- [demo] (           main) o.apache.catalina.core.StandardEngine : Starting Servlet engine
[Apache Tomcat/10.1.44]
2025-09-02T10:43:17.155+02:00 INFO 122500 --- [demo] (           main) o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-09-02T10:43:17.156+02:00 INFO 122500 --- [demo] (           main) w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext initialized at 622 ms
2025-09-02T10:43:17.427+02:00 WARN 122500 --- [demo] (           main) ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'openaiAudioSpeechModel' defined in class path resource [org/springframework/framework/a/i/model/openai/autoconfigure/OpenAIAudioSpeechAutoConfiguration.class]: Failed to instantiate [org.springframework.ai.openai.OpenAIAudioSpeechModel]: Factory method 'openAIAudioSpeechModel' threw exception with message: OpenAI API key must be set. Use the connection property: spring.ai.openai.api-key or spring.ai.openai.api-key property.
2025-09-02T10:43:17.429+02:00 INFO 122500 --- [demo] (           main) o.apache.catalina.core.StandardService : Stopping service [Tomcat]
2025-09-02T10:43:17.440+02:00 INFO 122500 --- [demo] (           main) .s.b.a.l.ConditionEvaluationReportLogger :
ERROR starting ApplicationContext. To display the condition evaluation report re-run your application with --debug enabled.
2025-09-02T10:43:17.452+02:00 ERROR 122500 --- [demo] (           main) o.s.boot.SpringApplication : Application run failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'openaiAudioSpeechModel' defined in class path resource [org/springframework/framework/a/i/model/openai/autoconfigure/OpenAIAudioSpeechAutoConfiguration.class]: Failed to instantiate [org.springframework.ai.openai.OpenAIAudioSpeechModel]: Factory method 'openAIAudioSpeechModel' threw exception with message: OpenAI API key must be set. Use the connection property: spring.ai.openai.api-key or spring.ai.openai.api-key property.
at org.springframework.beans.factory.support.ConstructorResolver.instantiate(ConstructorResolver.java:657) ~[spring-beans-6.2.10.j
sr.6.2.10]
at org.springframework.beans.factory.support.ConstructorResolver.instantiateUsingFactoryMethod(ConstructorResolver.java:645) ~[spr
ing-beans-6.2.10.jar:6.2.10]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.instantiateUsingFactoryMethod(AbstractAutowireCapa
bility.java:123) ~[spring-beans-6.2.10.jar:6.2.10]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:354)
~[spring-beans-6.2.10.jar:6.2.10]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:323)
~[spring-beans-6.2.10.jar:6.2.10]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:300)
~[spring-beans-6.2.10.jar:6.2.10]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:291)
~[spring-beans-6.2.10.jar:6.2.10]
at org.springframework.context.support.PostProcessorRegistrationDelegate.registerBeanPostProcessors(PostProcessorRegistrationDelegat
e.java:134) ~[spring-context-6.2.10.jar:6.2.10]
at org.springframework.context.support.AbstractApplicationContext.registerBeanPostProcessors(AbstractApplicationContext.java:750) ~[spr
ing-context-6.2.10.jar:6.2.10]
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:555) ~[spring-context-6.2.10.j
ar:6.2.10]
at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:734) ~[spring-boot-3.1.0.jar:3.1.0]
at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:435) ~[spring-boot-3.1.0.jar:3.1.0]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:321) ~[spring-boot-3.1.0.jar:3.1.0]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1267) ~[spring-boot-3.1.0.jar:3.1.0]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1255) ~[spring-boot-3.1.0.jar:3.1.0]
```

## 3 Ora non dovremmo ottenere eccezioni!

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

- IDE Layout:** The left sidebar contains "RUN AND DE..." and "DemoApp" buttons, a "WATCH" section with variables, and a "STACKTRACE" section. The bottom left has "BREAKPOINTS", "Uncaught Exceptions", and "Caught Exceptions".
- Code Editor:** The main editor shows an `application.yml` file with the following content:

```
spring:
  application:
    name: demo
  api:
    openai:
      api-key: ${GOOGLE_AI_API_KEY}
```
- Terminal:** The "TERMINAL" tab shows the following Spring Boot startup logs:

```
2025-09-02T10:48:24.985+02:00 [INFO] 128913 --- [main] i.venis.ai.spring.demo.DemoApplication : Starting DemoApplication
2025-09-02T10:48:25.594+02:00 [INFO] 128913 --- [main] i.venis.ai.spring.demo.DemoApplication : code/target/classes started
2025-09-02T10:48:25.622+02:00 [INFO] 128913 --- [main] i.venis.ai.spring.demo.DemoApplication : No active profile set, falling back to default profile: "default"
2025-09-02T10:48:25.685+02:00 [INFO] 128913 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8088 (http)
2025-09-02T10:48:25.593+02:00 [INFO] 128913 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomca]
2025-09-02T10:48:25.594+02:00 [INFO] 128913 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine
2025-09-02T10:48:25.622+02:00 [INFO] 128913 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-09-02T10:48:25.623+02:00 [INFO] 128913 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext initialized in 615 ms
2025-09-02T10:48:25.685+02:00 [INFO] 128913 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port: 8088 (http) with context path: ''
2025-09-02T10:48:26.087+02:00 [INFO] 128913 --- [main] i.venis.ai.spring.demo.DemoApplication : Started DemoApplication
```
- Bottom Status Bar:** Shows "Not Connected Yet", "Ln 6, Col 30", "Spaces 2", "UTF-8", "(3) Spring Boot Properties Yaml", and "English (US)".

<https://github.com/simonescannapieco/spring-ai-base-dgroove-venis-code.git>

*Branch:* 1-spring-ai-initializr