



ICT Training Center

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



Note



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

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

Settembre 2025

 Simone Scannapieco

 Spring AI - Corso base

 Venis S.p.A, Venezia, IT

2 / 11

Note



SETUP PROGETTO SPRING AI

Note

- ➔ 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

Note

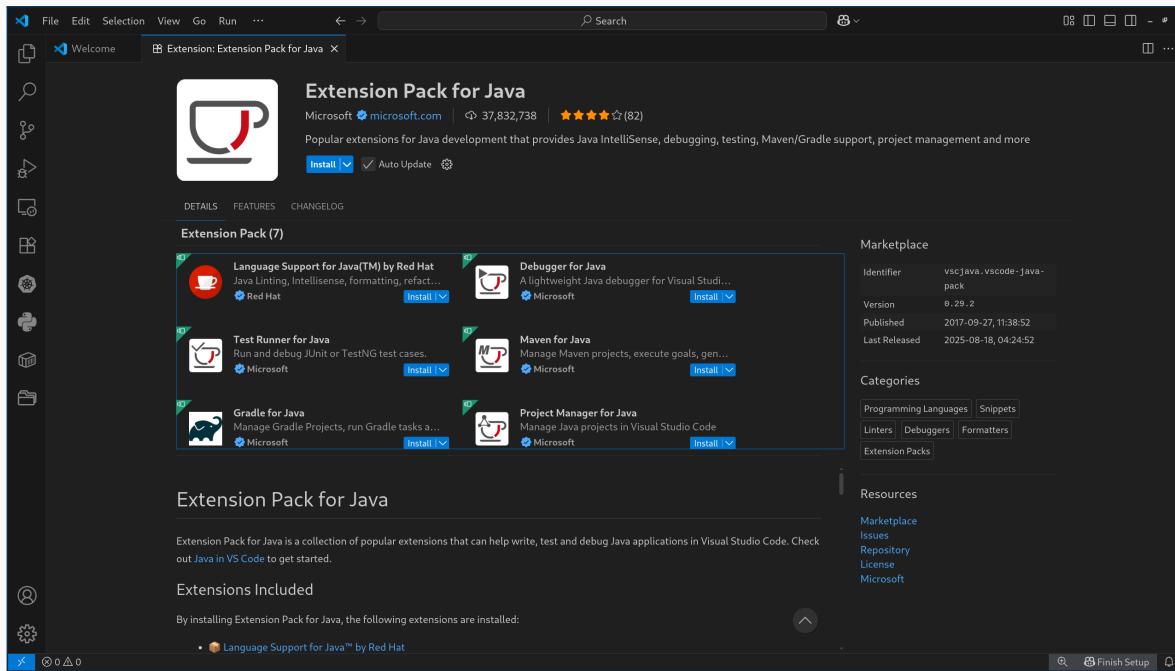
- 1 Inizializzare un nuovo progetto Spring AI
 - 2 Verifica JRE per progetto
 - 3 Esecuzione del progetto

Note

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

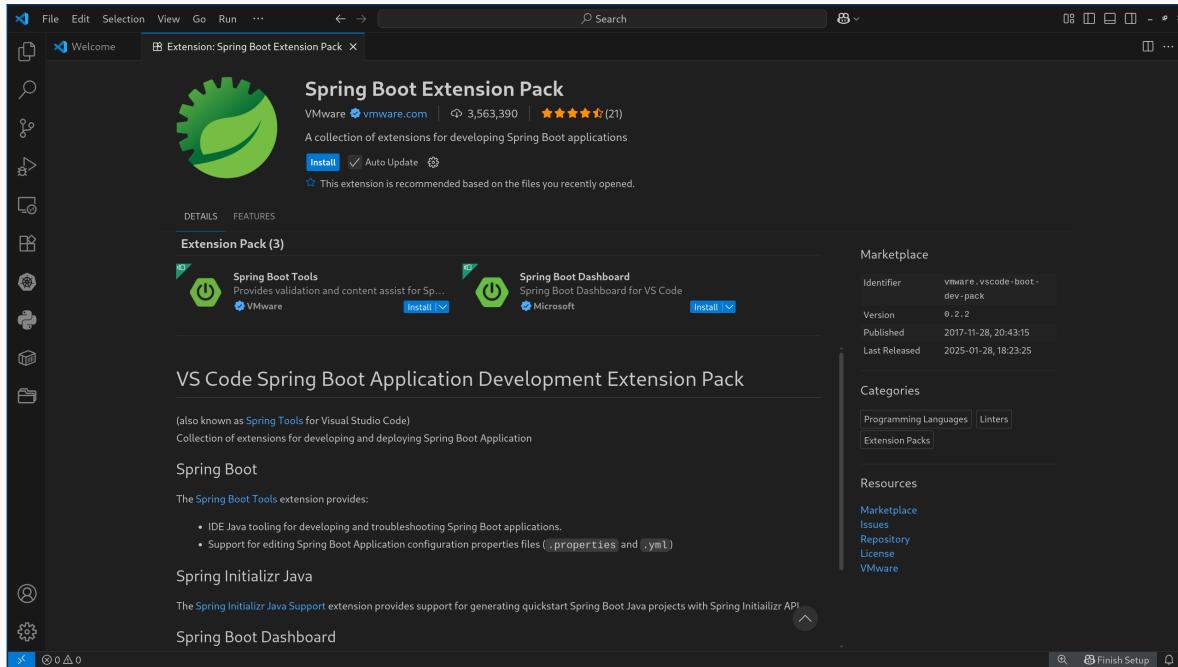
Note

1 Installare le estensioni VSC necessarie



Note

1 Installare le estensioni VSC necessarie



 Simone Scannapieco

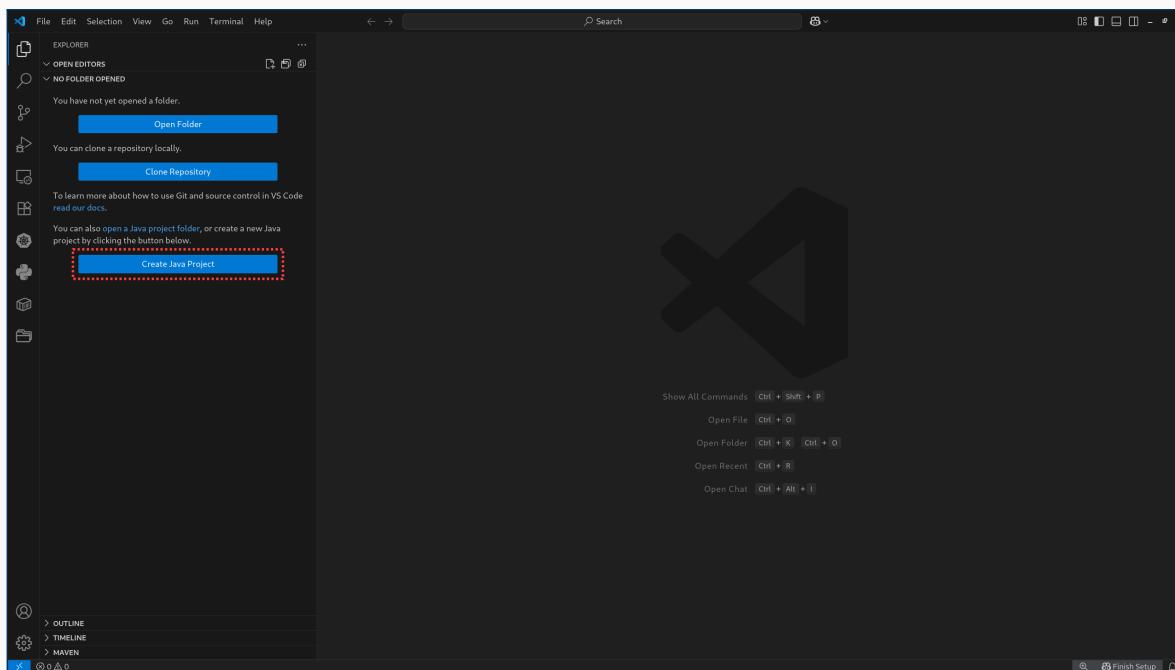
 Spring AI - Corso base

 Venis S.p.A, Venezia, IT

6 / 11

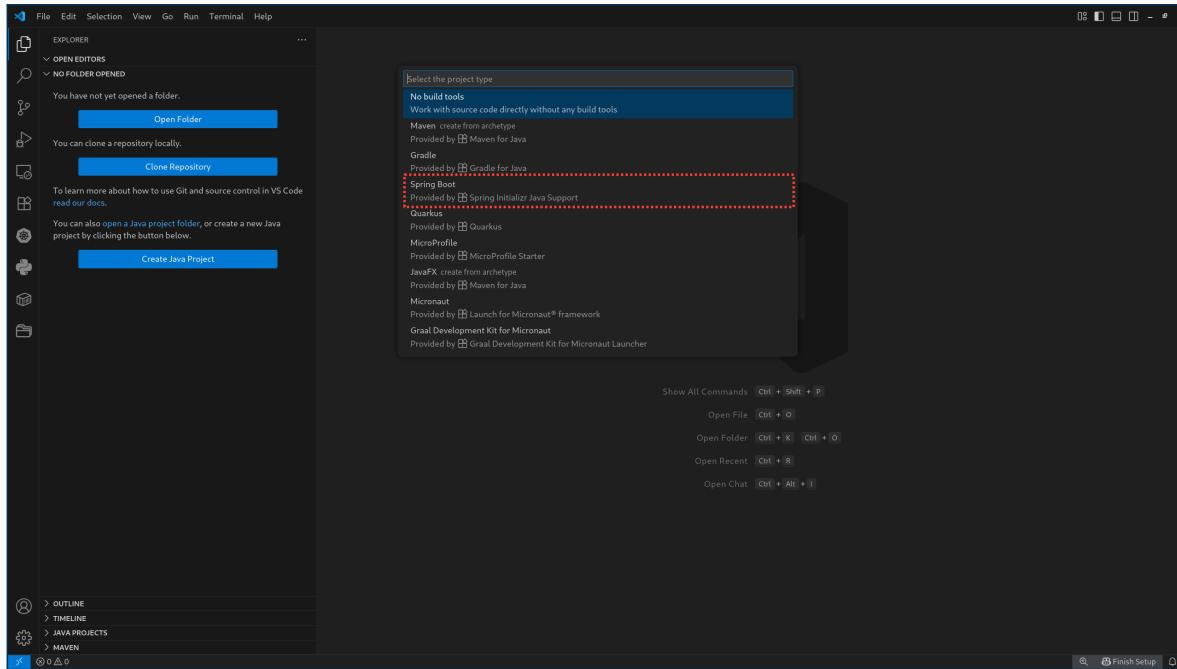
Note

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



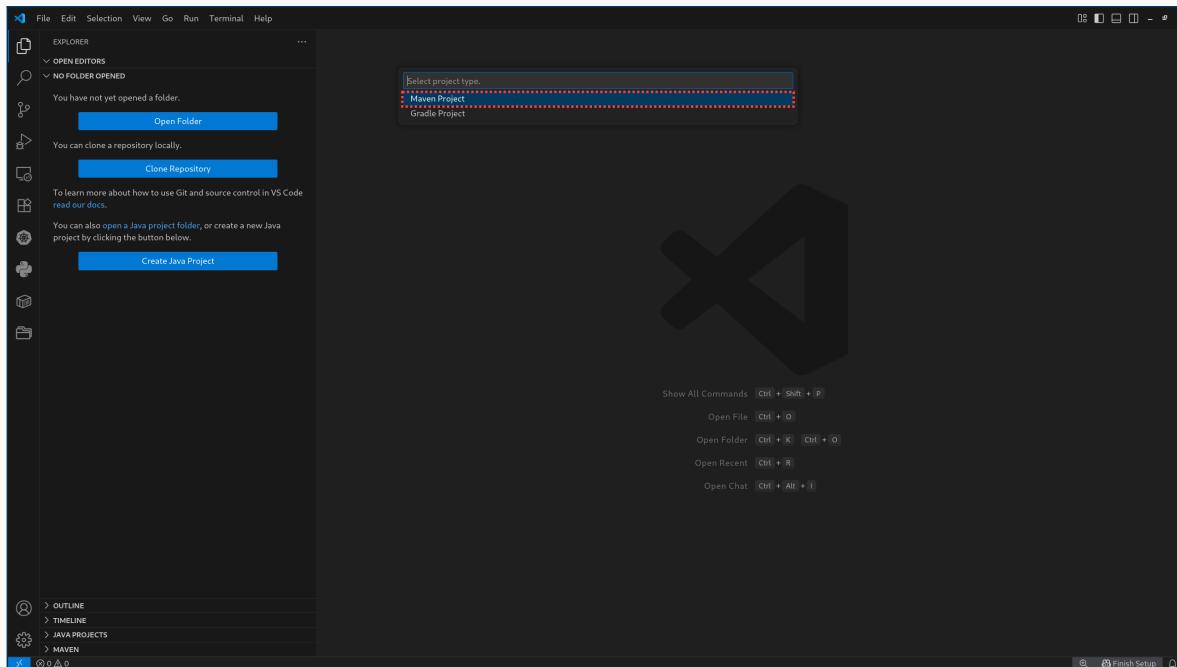
Note

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



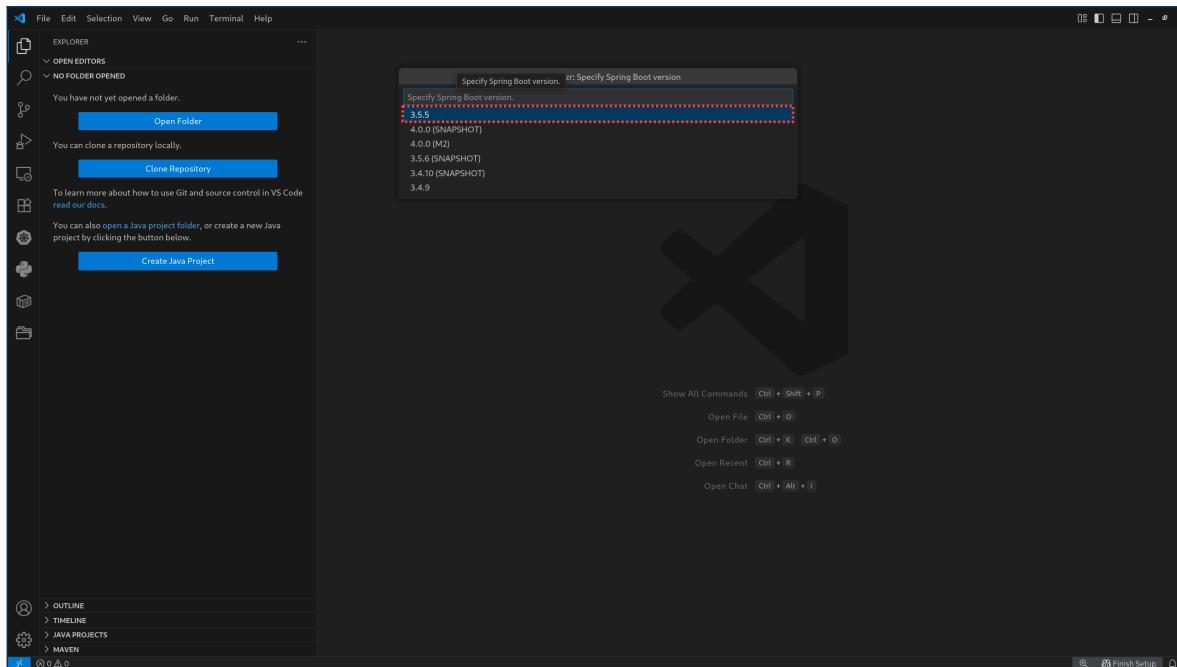
Note

4 Selezionare Maven project come tipo di progetto



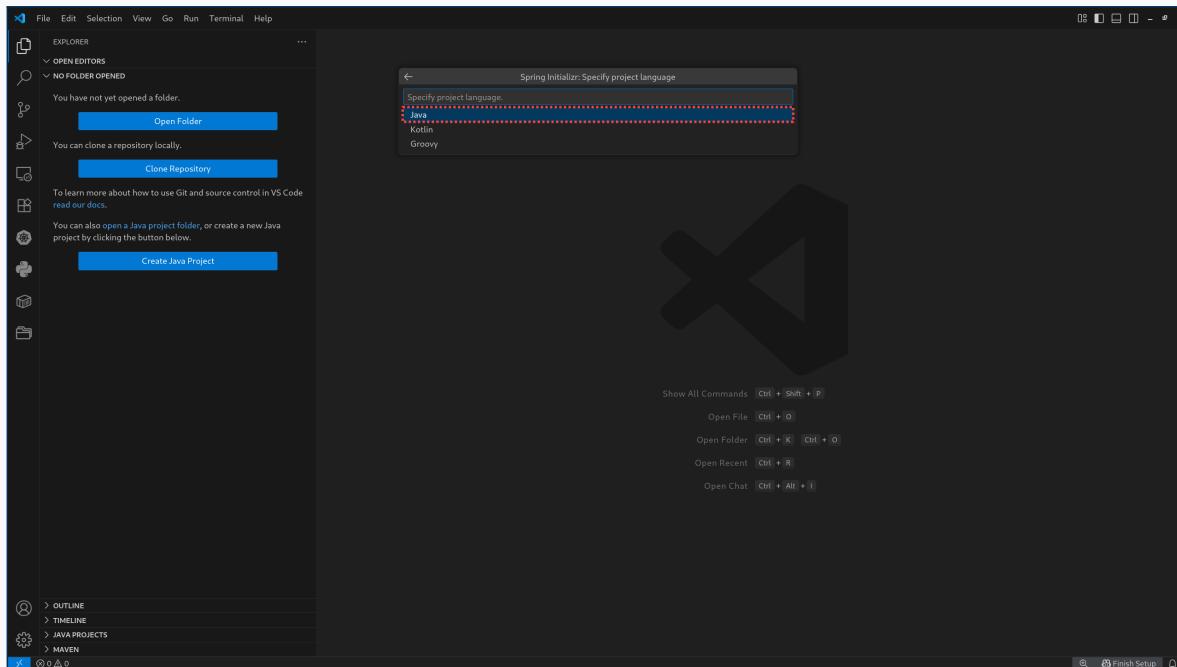
Note

5 Selezionare la versione di Boot per il progetto



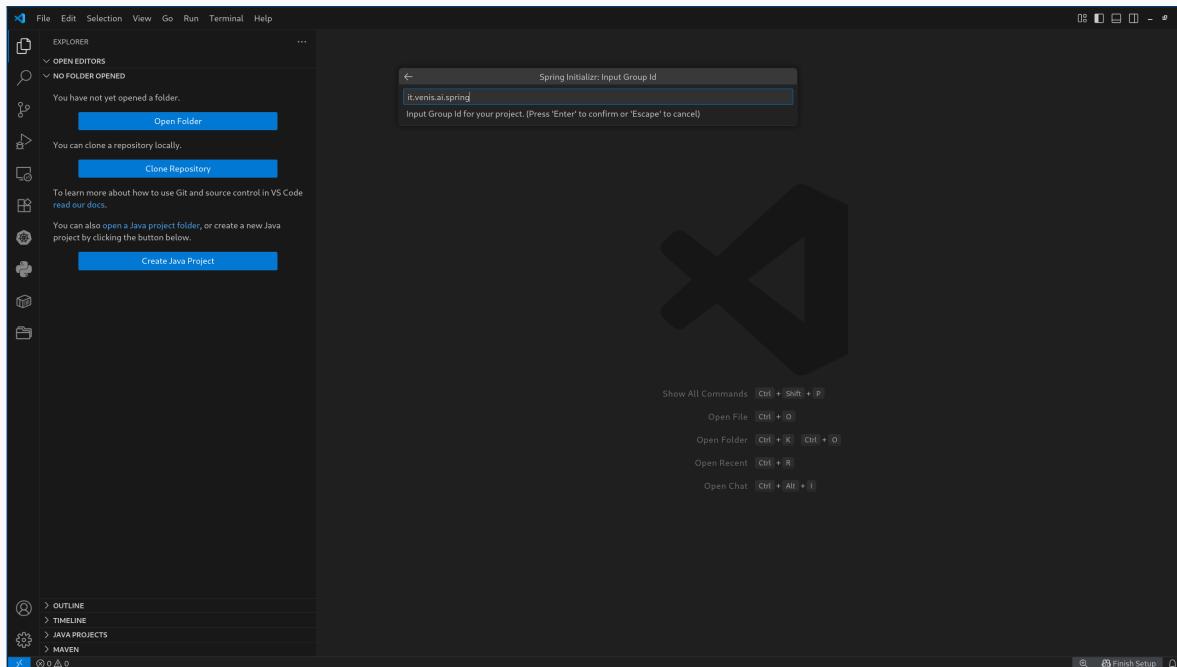
Note

6 Selezionare Java come linguaggio di progetto



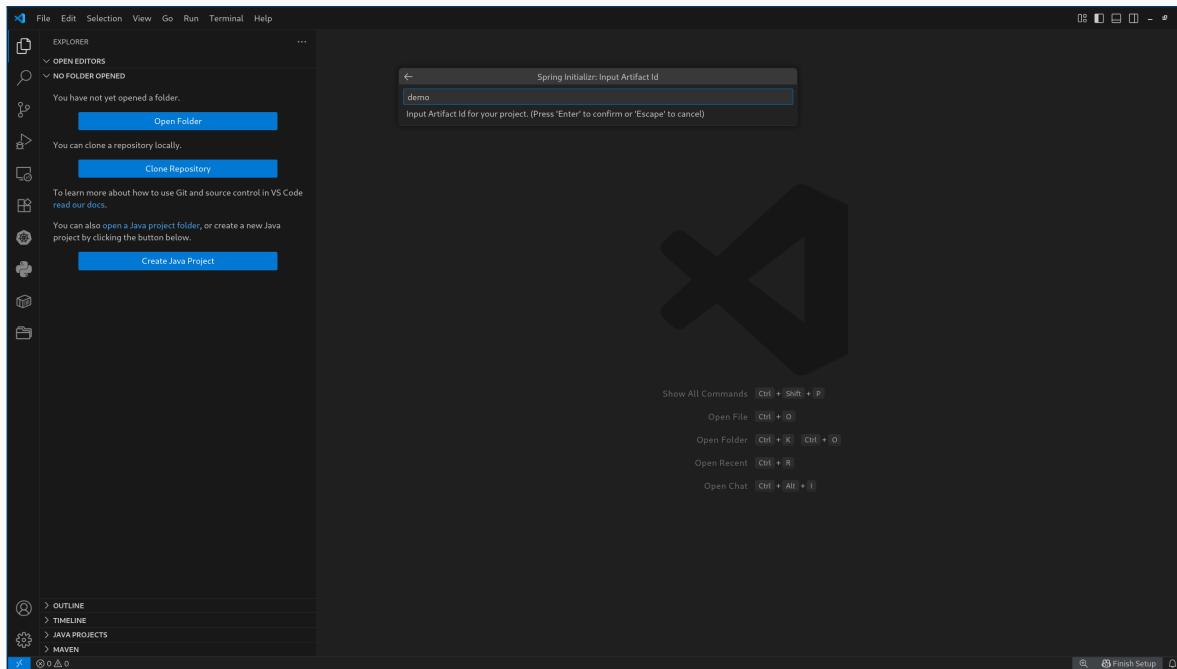
Note

7 Definire il *Group ID* di riferimento del progetto



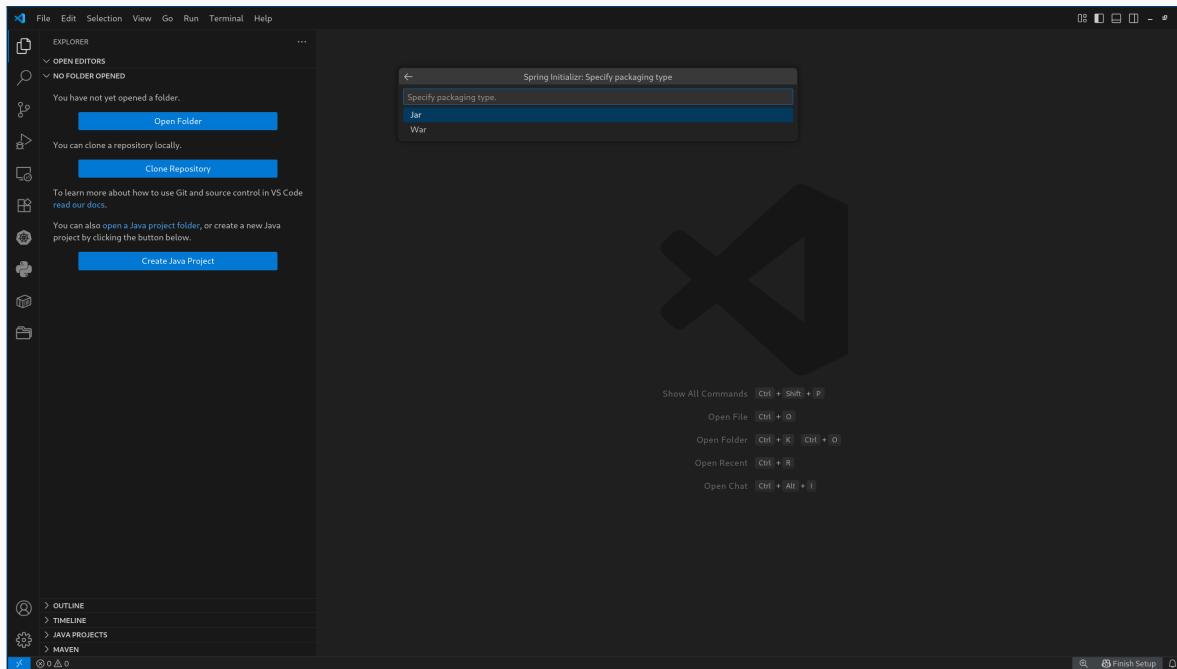
Note

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



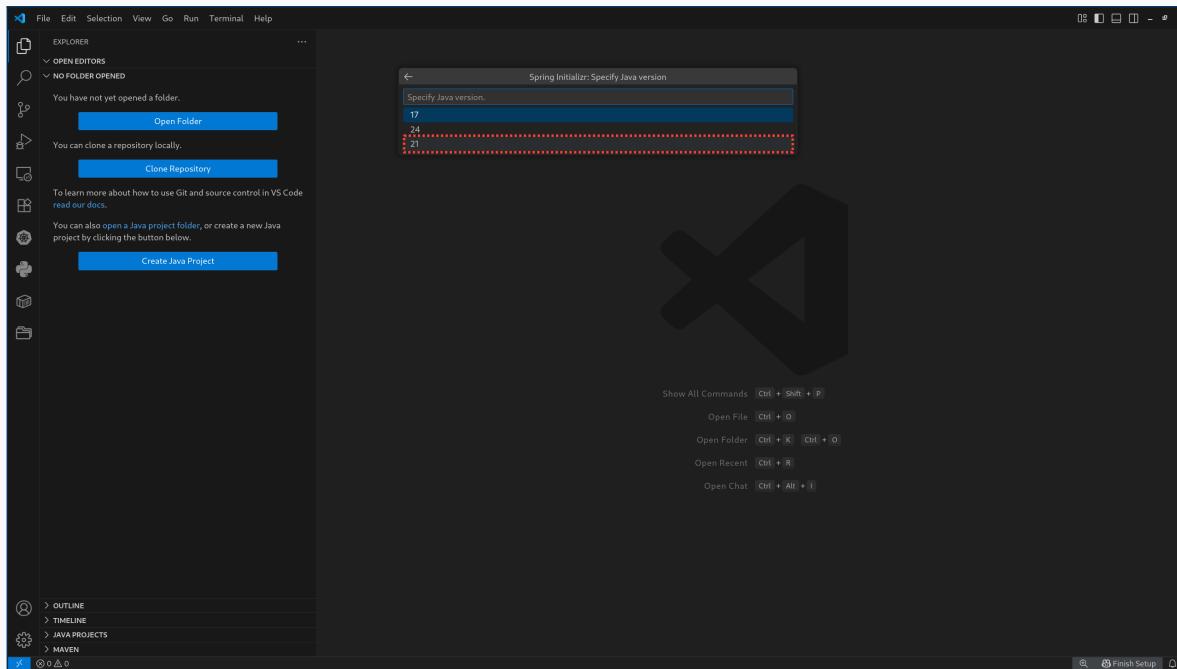
Note

9 Definire la tipologia di *packaging* del progetto



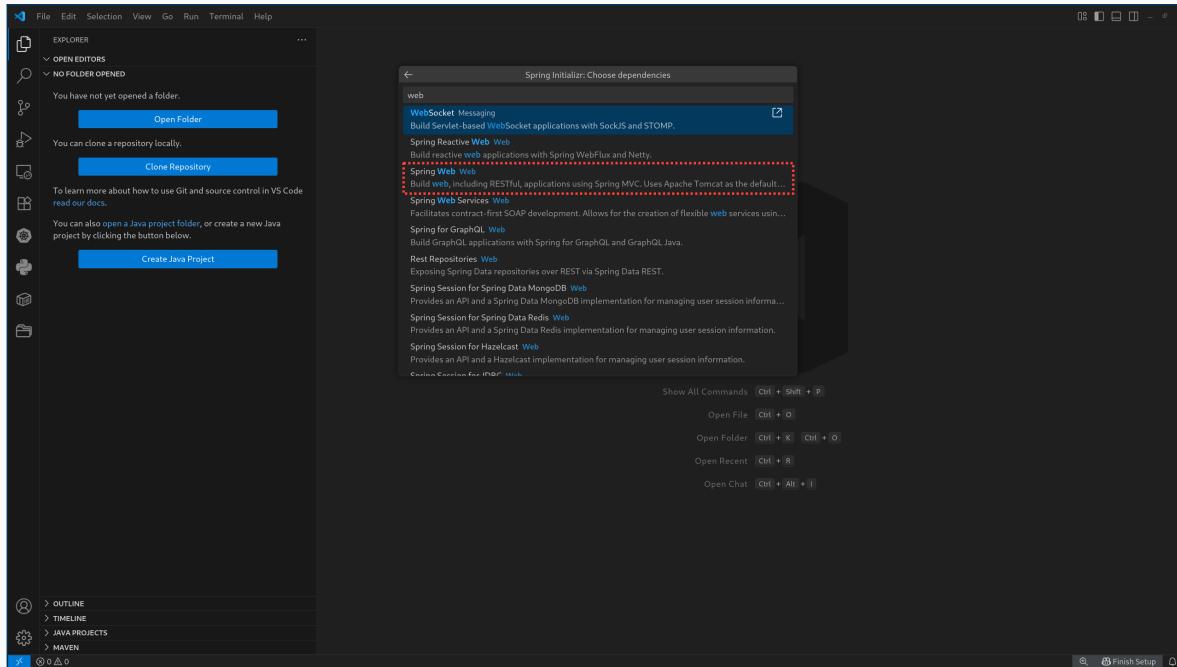
Note

10 Selezionare la versione di Java di riferimento



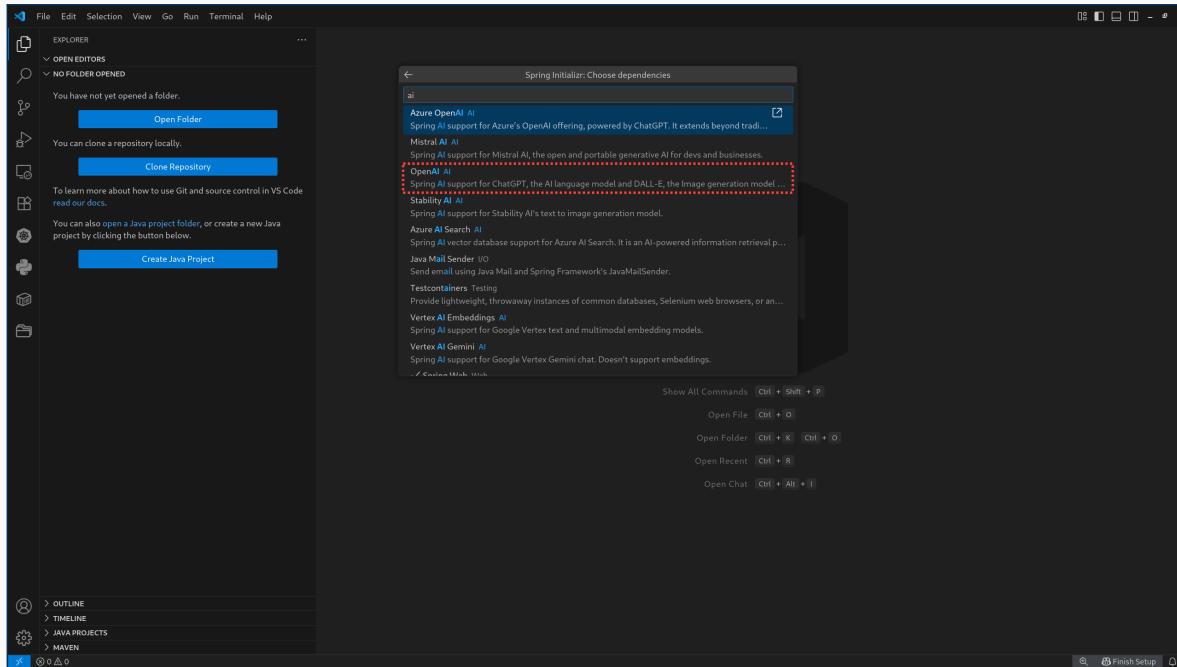
Note

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



Note

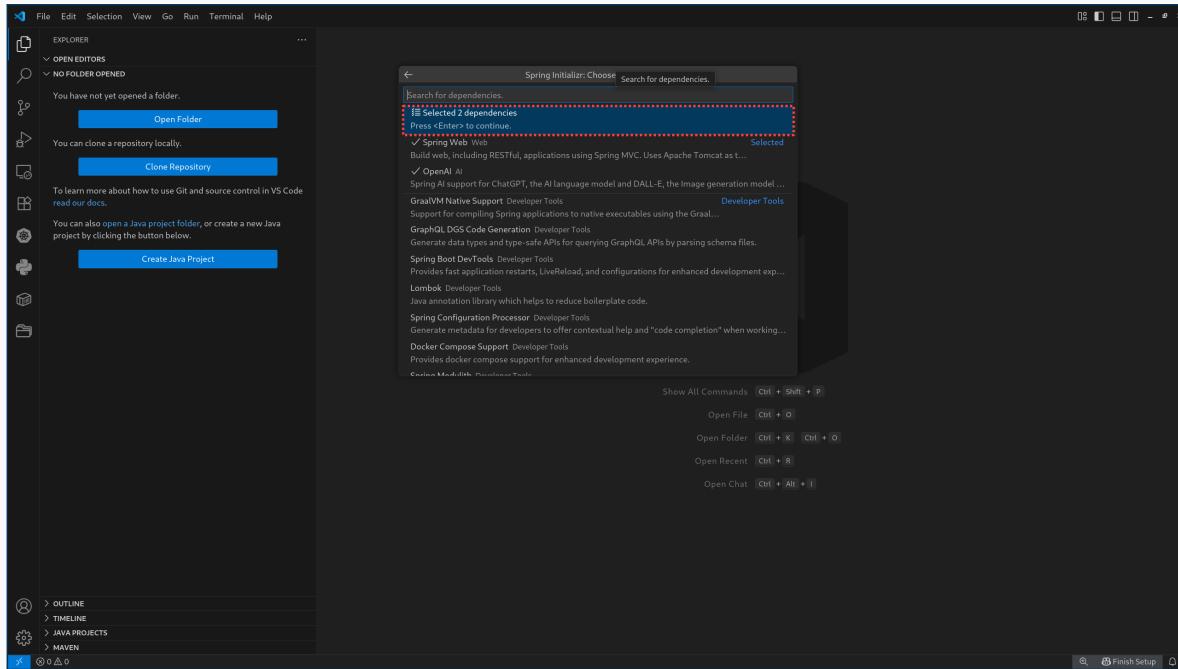
12 Selezionare come seconda dipendenza di progetto OpenAI



Note

PROGETTO SPRING AI INIZIALIZZAZIONE DA IDE (VS CODE)

13 Continuare la creazione del progetto premendo **Enter**



 Simone Scannapieco

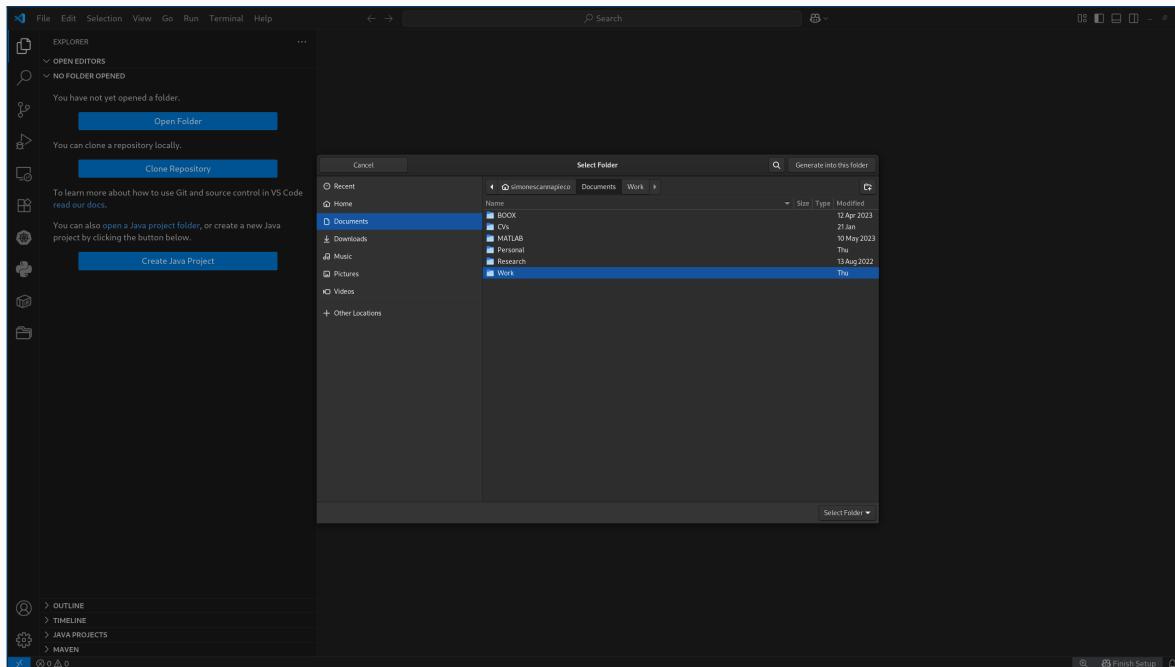
 Spring AI - Corso base

 Venis S.p.A, Venezia, IT

6 / 11

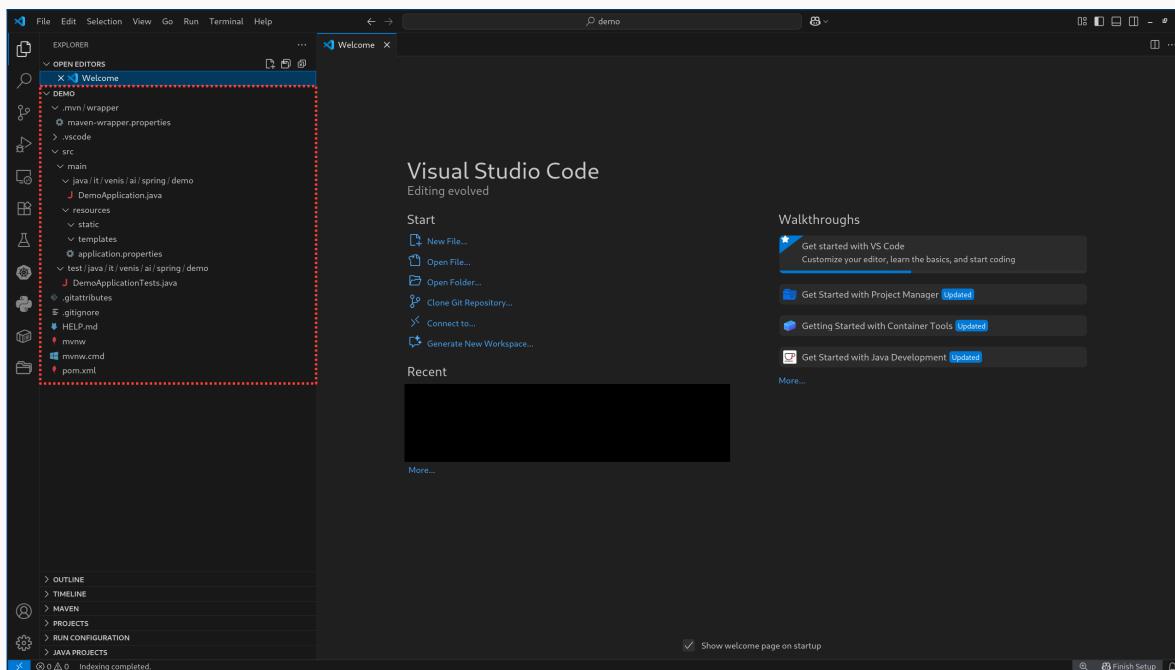
Note

14 Navigare nel FS e selezionare la cartella padre del progetto



Note

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



Note

PROGETTO SPRING INIZIALIZZAZIONE DA SPRING INITIALIZER

The screenshot shows the Spring Initializr web interface. At the top, there's a navigation bar with tabs for 'start-spring.js' and 'Incohesive'. The main content area has a header 'spring initializr' with a green logo.

Project settings:

- Language: Java (selected)
- Build Tools: Maven (selected)
- Frameworks: Gradle - Groovy, Gradle - Kotlin, Groovy (radio buttons)

Spring Boot settings:

- Version: 3.5.6 (SNAPSHOT) (selected)
- Other options: 4.0.0 (M2), 3.4.9 (radio buttons)

Project Metadata settings:

- Group: it.venis.ai.spring
- Artifact: demo
- Name: demo
- Description: Demo project for Spring Boot (Web and OpenAI)
- Package name: it.venis.ai.spring.demo
- Packaging: Jar (selected)
- Java version: 24 (selected)
- Java version: 21 (radio button)
- Java version: 17 (radio button)

Dependencies section:

- Add Dependencies... (button)
- Spring Web (selected): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- OpenAI (AI): Spring AI support for ChatGPT, the AI language model and DALL-E, the Image generation model from OpenAI.

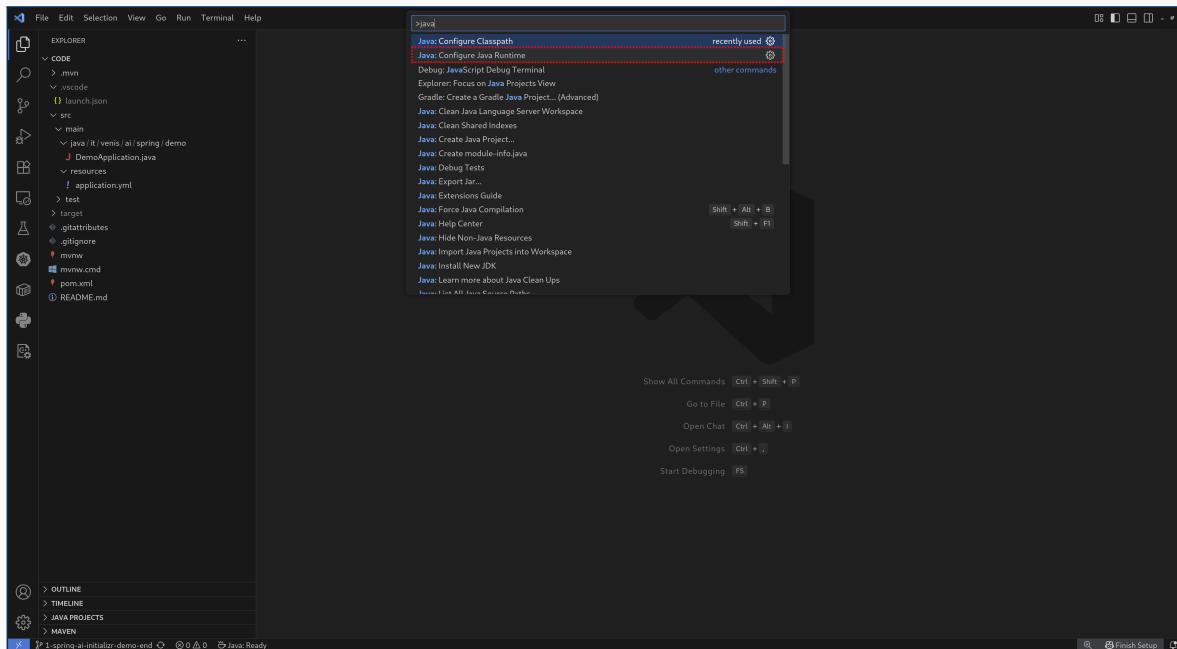
At the bottom, there are 'GENERATE' and 'EXPLORE' buttons, and a three-dot menu icon.

Portale ufficiale

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

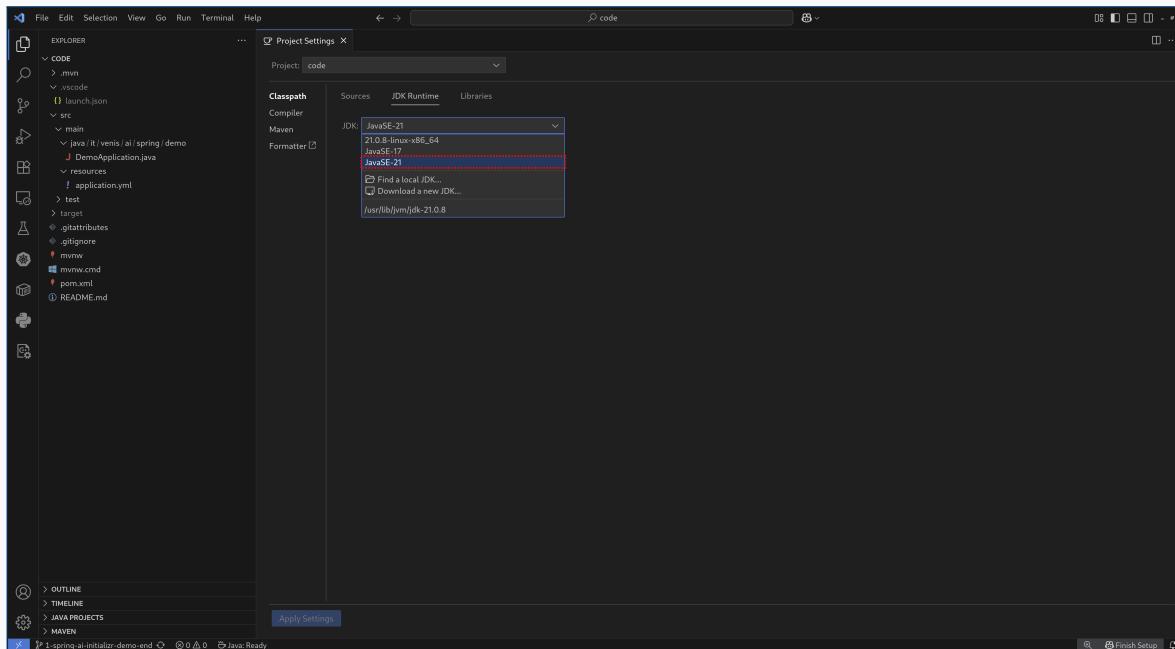
Note

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



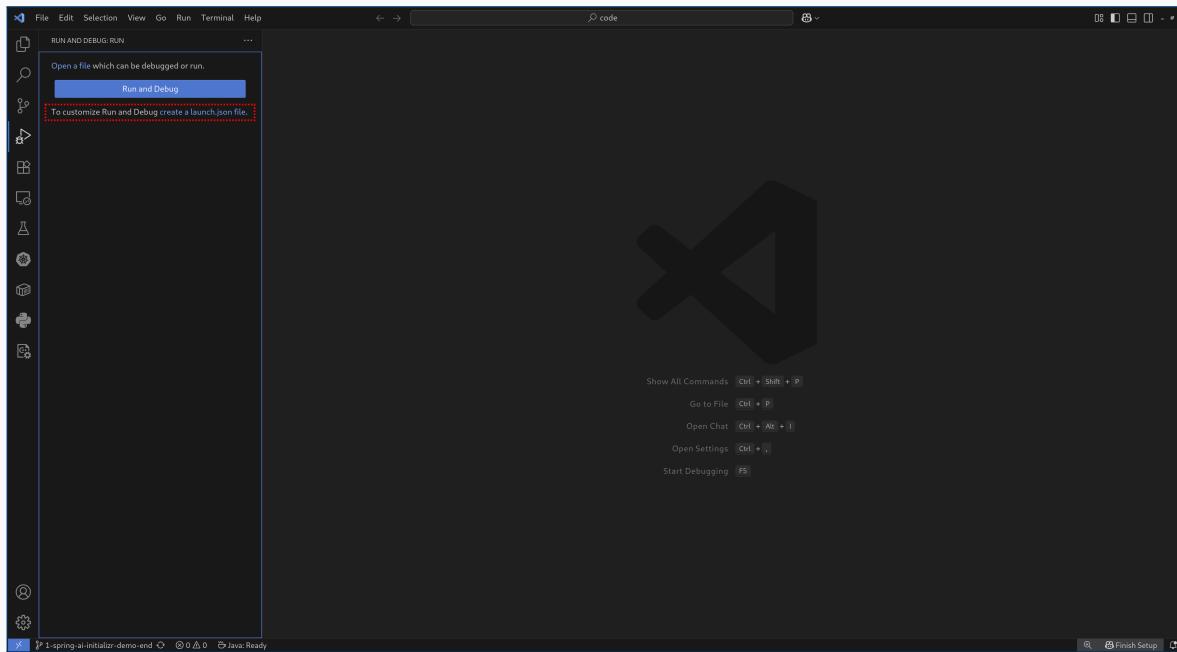
Note

2 Selezionare la JDK 21 precedentemente installata



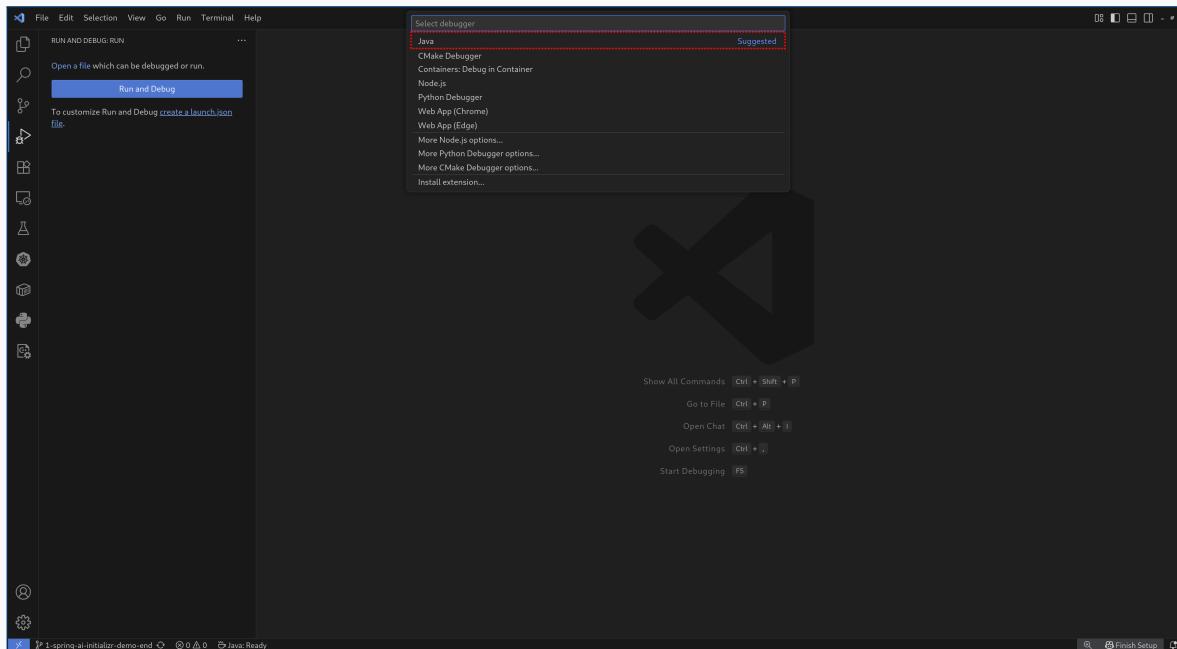
Note

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



Note

2 Selezionare Java come tipologia di esecuzione/debug



Note

3 Verificare la presenza delle due configurazioni di base



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

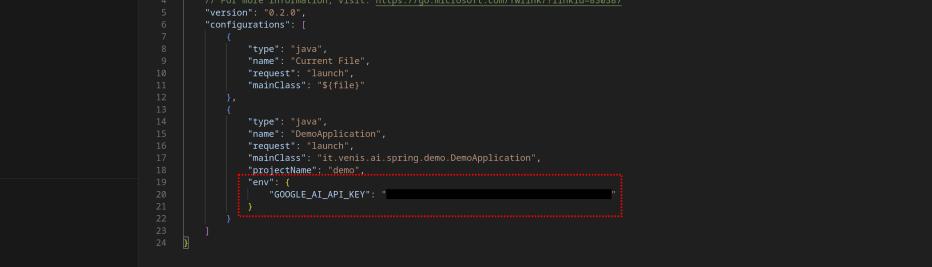
- File Explorer:** On the left, showing a folder structure with 'src' and 'target'.
- Launch View:** The main editor area displays the `launch.json` configuration file. The code is as follows:

```
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"
```

- Output:** Shows 'Java Ready'.
- Bottom Status Bar:** Includes icons for file status (green checkmark), breakpoints (red dot), and Java support (green checkmark).
- Bottom Right:** Buttons for 'Add Configuration...', 'Ln 1, Col 2', 'Spaces: 4', 'UTF-8', 'LF', 'JSON with Comments', and 'Finish Setup'.

Note

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



```
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",
        "env": {
            "GOOGLE_AI_API_KEY": ""
        }
    }
]
```

Note

1 Eseguendo il progetto, dovremmo ottenere una eccezione!

The screenshot shows the Eclipse IDE interface with the following details:

- File, Edit, Selection, View, Go, Run, Terminal, Help** menu bar.
- Toolbar icons** for RUN and DE, Variables, Problems, Output, Debug Console, Terminal, and Ports.
- application.xml** file open in the editor, defining a Spring application named "demo".
- TERMINAL** tab selected, displaying the application's startup logs:

 - INFO [demo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine [Apache Tomcat/10.1.44]
 - INFO [demo] [main] o.a.c.c.C.Tomcat : [localhost] [/] : Initializing Spring embedded WebApplicationContext
 - INFO [demo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext ext: initialization completed in 622 ms...
 - WARN [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/ai/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.speech.api-key property.
 - INFO [demo] [main] o.apache.catalina.startup.TomcatService : Stopping service [Tomcat]

- WATCH** and **CALL STACK** toolbars are visible.
- BREAKPOINTS** toolbar with checkboxes for Uncaught Exceptions and Caught Exceptions.
- PROBLEMS**, **OUTPUT**, **DEBUG CONSOLE**, **TERMINAL**, **PORTS** tabs at the bottom.
- Terminal Output**:

```
2025-09-02T10:43:17.128+02:00 INFO 122580 --- [demo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine [Apache Tomcat/10.1.44]
2025-09-02T10:43:17.155+02:00 INFO 122580 --- [demo] [main] o.a.c.c.C.Tomcat : [localhost] [/] : Initializing Spring embedded WebApplicationContext
2025-09-02T10:43:17.156+02:00 INFO 122580 --- [demo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext ext: initialization completed in 622 ms...
2025-09-02T10:43:17.157+02:00 WARN 122580 --- [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/ai/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.speech.api-key property.
2025-09-02T10:43:17.158+02:00 INFO 122580 --- [demo] [main] o.apache.catalina.startup.TomcatService : Stopping service [Tomcat]
```
- bash** terminal tab and **Debug:DemoApplication** configuration tab.

Note

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

The screenshot shows a Java application running in a Spring Boot IDE. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a back/forward button. A search bar labeled 'code' is positioned at the top right. The main workspace displays a code editor with 'application.yml' containing configuration for a 'spring.application.name: demo' and an 'OpenAI' section with 'api-key: \$GOOGLE_AI_API_KEY'. Below the code editor is a toolbar with PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS buttons. The TERMINAL tab is active, showing logs from Apache Tomcat and the Spring boot application. The logs indicate the engine is starting, initializing Spring, and attempting to load an 'openAiAudioSpeechModel'. It also shows a warning about the API key and an error where the application failed to run due to the missing API key. The right side of the interface features a 'WATCH' panel with variables like 'bash' and 'Debug: DemoApplication' and a 'CALL STACK' panel.

```
src > main > resources > application.yml
1 spring:
2   application:
3     name: demo
4   api:
5     openai:
6       api-key: $GOOGLE_AI_API_KEY
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
t]
2025-09-02T10:43:17.128+02:00 INFO 122580 --- [demo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine
: [Apache Tomcat/10.1.44]
2025-09-02T10:43:17.155+02:00 INFO 122580 --- [demo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring mb
embedded WebApplicationContext
2025-09-02T10:43:17.156+02:00 INFO 122580 --- [demo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationCont
ext: initialization completed in 622 ms
2025-09-02T10:43:17.427+02:00 WARN 122580 --- [demo] [main] ConfigServletWebServerApplicationContext : Exception encountered d
uring context initialization - cancelling refresh attempt: org.springframework.beans.factory.BeanCreationException: Error creating bean wi
th name 'openAiAudioSpeechModel' defined in class path resource [org/springframework/ai/model/openai/autoconfigure/OpenAIAudioSpeechAutoCo
nfiguration.class]: Failed to instantiate [org.springframework.ai.model.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.spe
ech.api-key property.
2025-09-02T10:43:17.429+02:00 INFO 122580 --- [demo] [main] o.apache.catalina.core.StandardService : Stopping service [Tomca
t]
2025-09-02T10:43:17.440+02:00 INFO 122580 --- [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 122580 --- [demo] [main] o.s.boot.SpringApplication : Application run failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'openAiAudioSpeechModel' defined in class path reso
urce [org/springframework/ai/model/openai/autoconfigure/OpenAIAudioSpeechAutoConfiguration.class]: Failed to instantiate [org.springframew
ork.ai.model.OpenAIAudioSpeechModel]: Factory method 'openAiAudioSpeechModel()' threw exception with message: OpenAI API key must be set. U
se the connection property: spring.ai.openai.api-key or spring.ai.openai.speech.api-key property.
        at org.springframework.beans.factory.support.ConstructorResolver.instantiate(ConstructorResolver.java:657) ~[spring-beans-6.2.10.j
ar: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
bleBeanFactory.java:621) ~[spring-beans-6.2.10.jar:6.2.10]
        at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:537)
~[spring-beans-6.2.10.jar:6.2.10]
        at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:335) ~[spring-beans-6.2.10.j
ar:6.2.10]
        at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:234) ~[spring-bean
s-6.2.10.jar:6.2.10]
        at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:333) ~[spring-beans-6.2.10.jar:6.2.10]
        at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:115) ~[spring-context-6.2.10.j
ar:6.2.10]
        at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:875) ~[spring-c
ontext-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.web.servlet.context.AnnotationConfigServletWebServerApplicationContext.refresh(AnnotationConfigServletWebServerApplicati
onContext.java:163) ~[spring-boot-3.1.0.jar:3.1.0]
        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:342) ~[spring-boot-3.1.0.jar:3.1.0]
        at org.springframework.boot.SpringApplication.run(SpringApplication.java:331) ~[spring-boot-3.1.0.jar:3.1.0]
        at com.demoproject.DemoApplication.main(DemoApplication.java:10) ~[main-class:DemoApplication]
```

Note

3 Ora non dovremmo ottenere eccezioni!

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

- File Explorer:** Shows a project structure with files like `application.yml`, `build.gradle`, and `src/main/resources/application.yml`.
- Terminal:** Displays the output of the Spring Boot application startup, including logs from `main`, `venus.ai.spring.demo.DemoApplication`, and `o.s.b.w.embedded.tomcat.TomcatWebServer`.
- Output:** Shows the output of the build process.
- Debug:** A debug session named "Debug: DemoApplication" is active.
- Breakpoints:** A list of breakpoints, some marked as "Caught Exceptions".

The bottom status bar indicates the file is "Not Committed Yet" and shows other system information.

Note

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

Branch: 1-spring-ai-initializr

Note