

Azure SDK for Java: How to create an Azure ResourceGroup

IMPORTANT NOTE: for more information about this example please visit these URLs:

<https://github.com/Azure/azure-sdk-for-java/tree/main>

<https://github.com/Azure/azure-sdk-for-java/blob/main/sdk/resourcemanager/docs/SAMPLE.md>

1. Prerequisites

1.1. Download and install Apache Maven

Download Apache Maven from the web page: <https://maven.apache.org/download.cgi>

The screenshot shows the Apache Maven Project website for version 3.9.5. The page includes a sidebar with navigation links, a main content area with system requirements, and a table of download links.

Downloading Apache Maven 3.9.5

Apache Maven 3.9.5 is the latest release: it is the recommended version for all users.

System Requirements

| Requirement | Details |
|----------------------------|---|
| Java Development Kit (JDK) | Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains . |
| Memory | No minimum requirement |
| Disk | Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB. |
| Operating System | No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) and Windows batch files. |

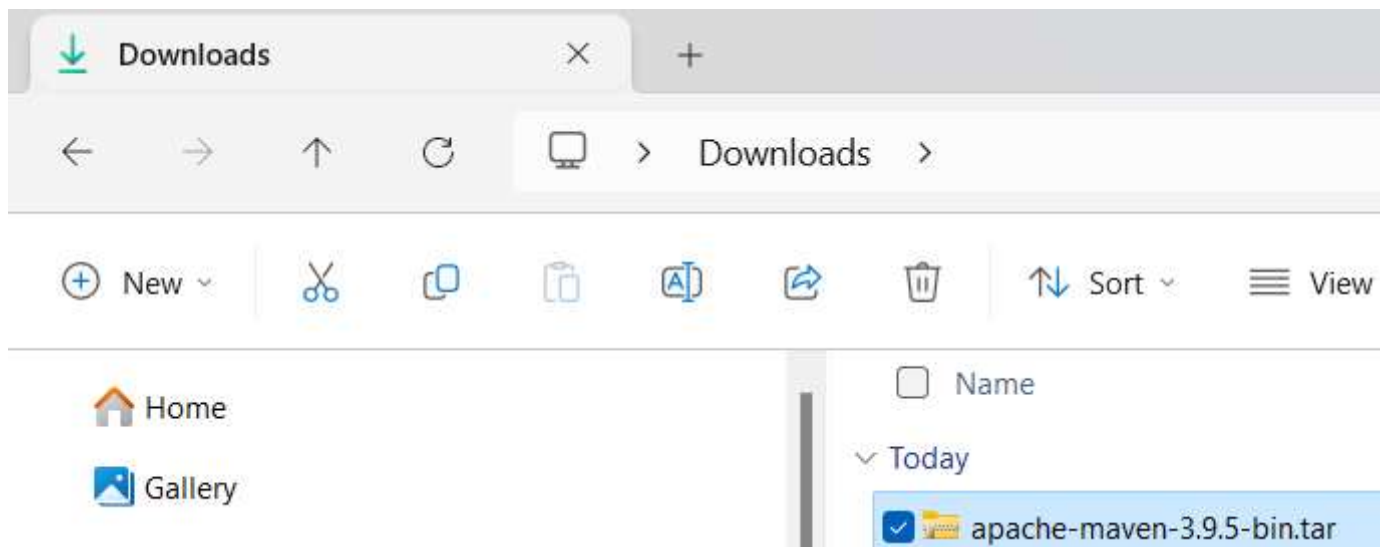
Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

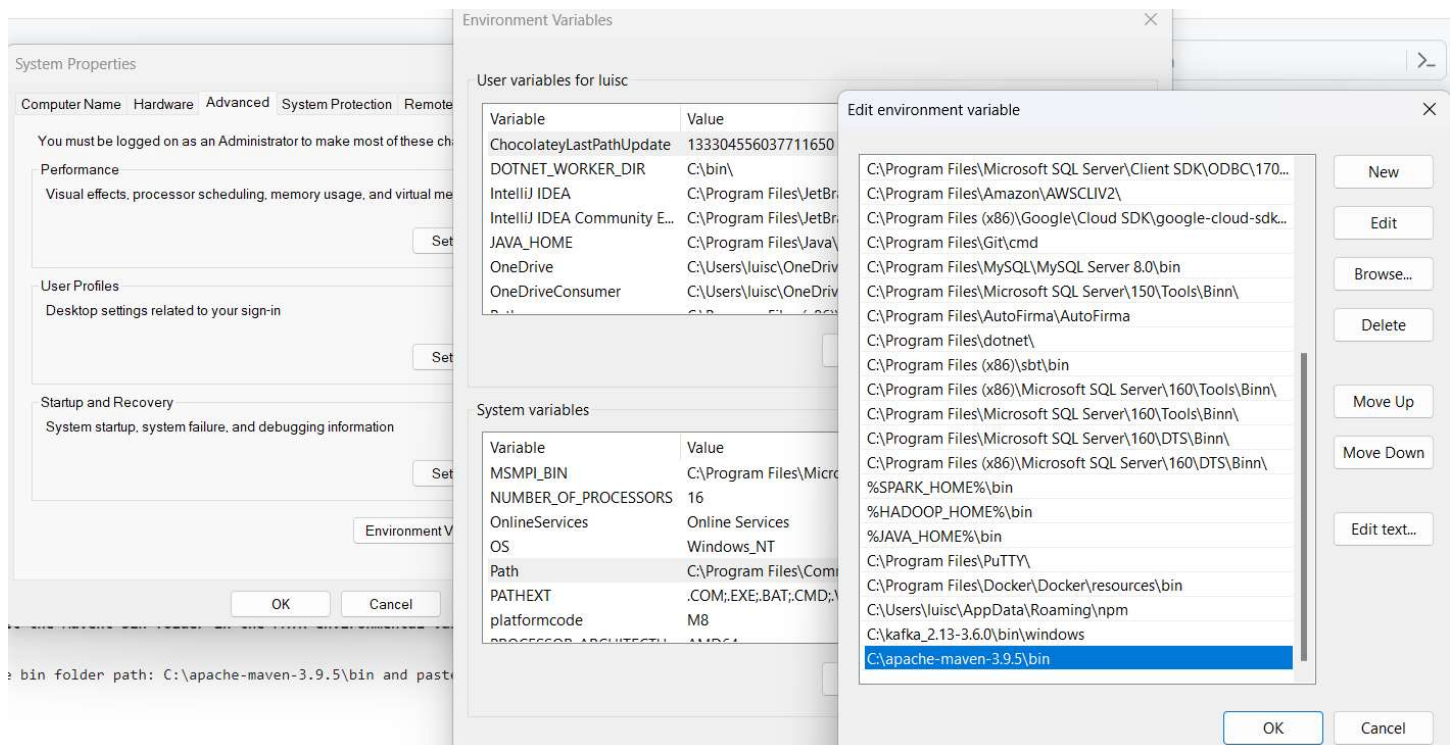
| | Link | Checksums | Signature |
|-----------------------|--|--|---|
| Binary tar.gz archive | apache-maven-3.9.5-bin.tar.gz | apache-maven-3.9.5-bin.tar.gz.sha512 | apache-maven-3.9.5-bin.tar.gz.asc |

Unzip the `apache-maven-3.9.5-bin.tar` file and place the folder in the C: root hard disk



1.2. Set the Mavent bin folder in the PATH environmental variable

Copy the bin folder path: **C:\apache-maven-3.9.5\bin** and paste in the **PATH** environmental variable



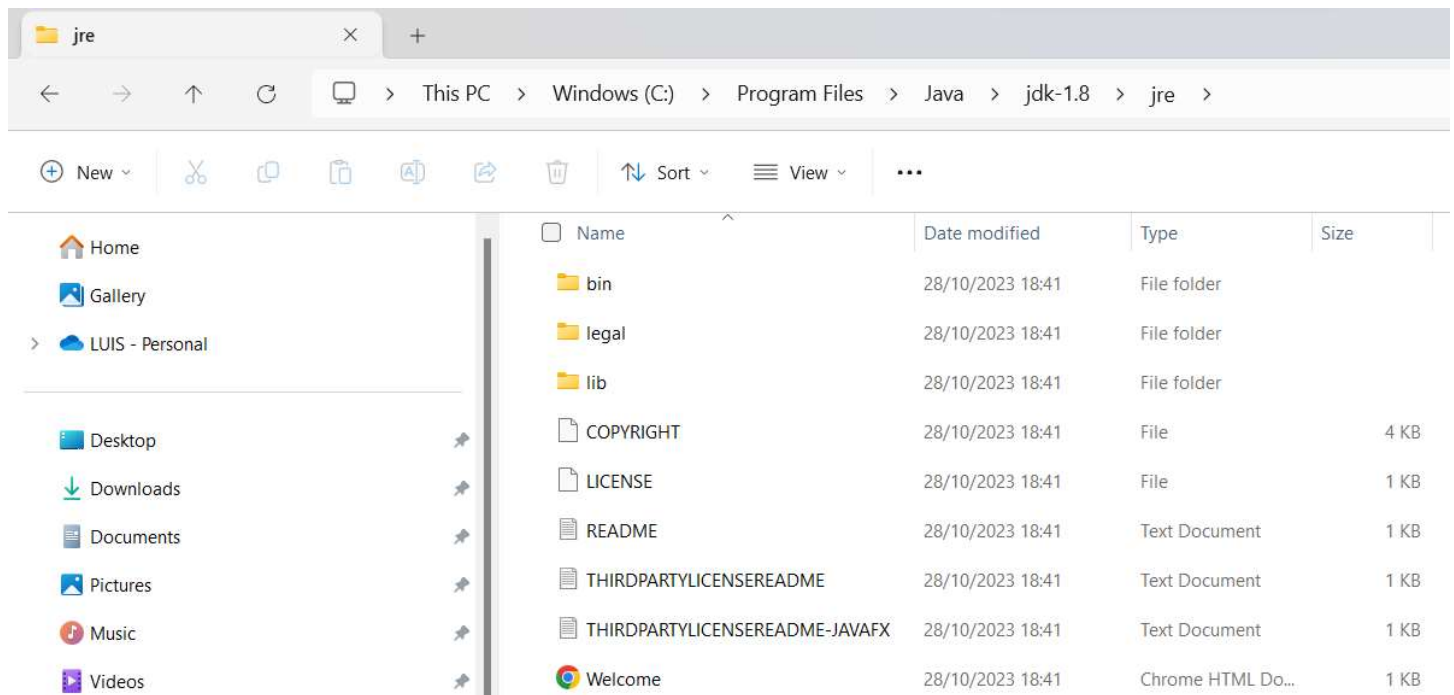
1.3. Verify Maven installation

Run this command to verify the Maven installation

```
mvn -v
```

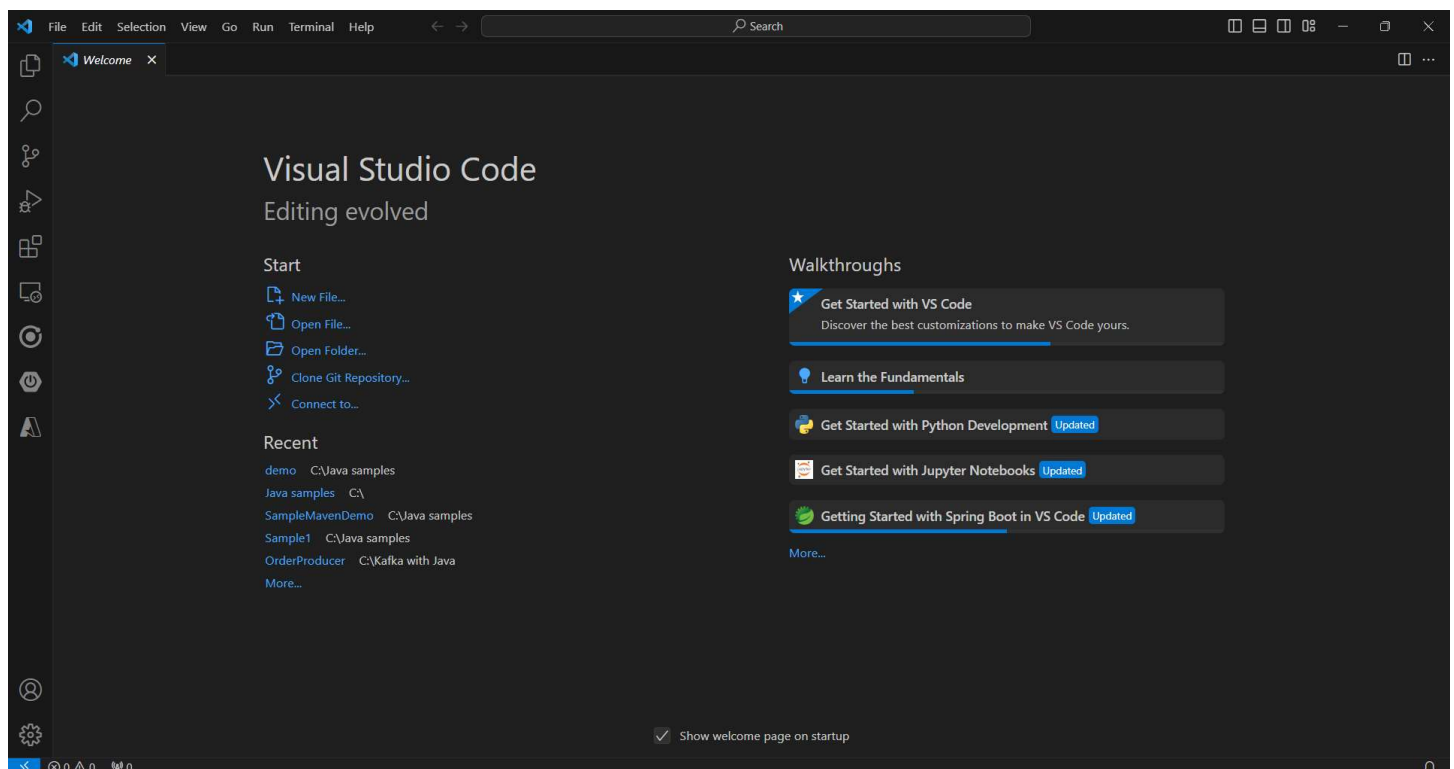
```
C:\Users\luisc>mvn -v
Apache Maven 3.9.5 (57804ffe001d7215b5e7bcb531cf83df38f93546)
Maven home: C:\apache-maven-3.9.5
Java version: 1.8.0_391, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-1.8\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

IMPORTANT NOTE: We already installed before "C:\Program Files\Java\jdk-1.8\jre"

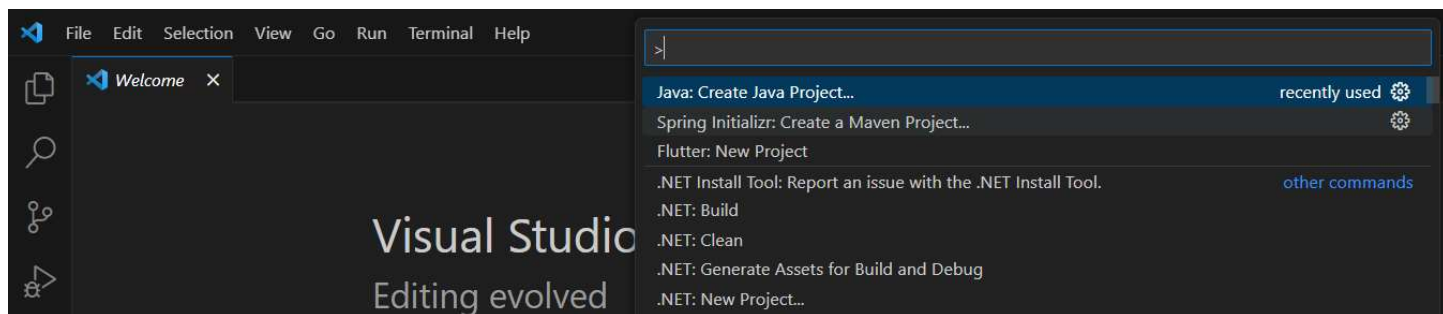


2. Open VSCode and create a blank Java with Maven project

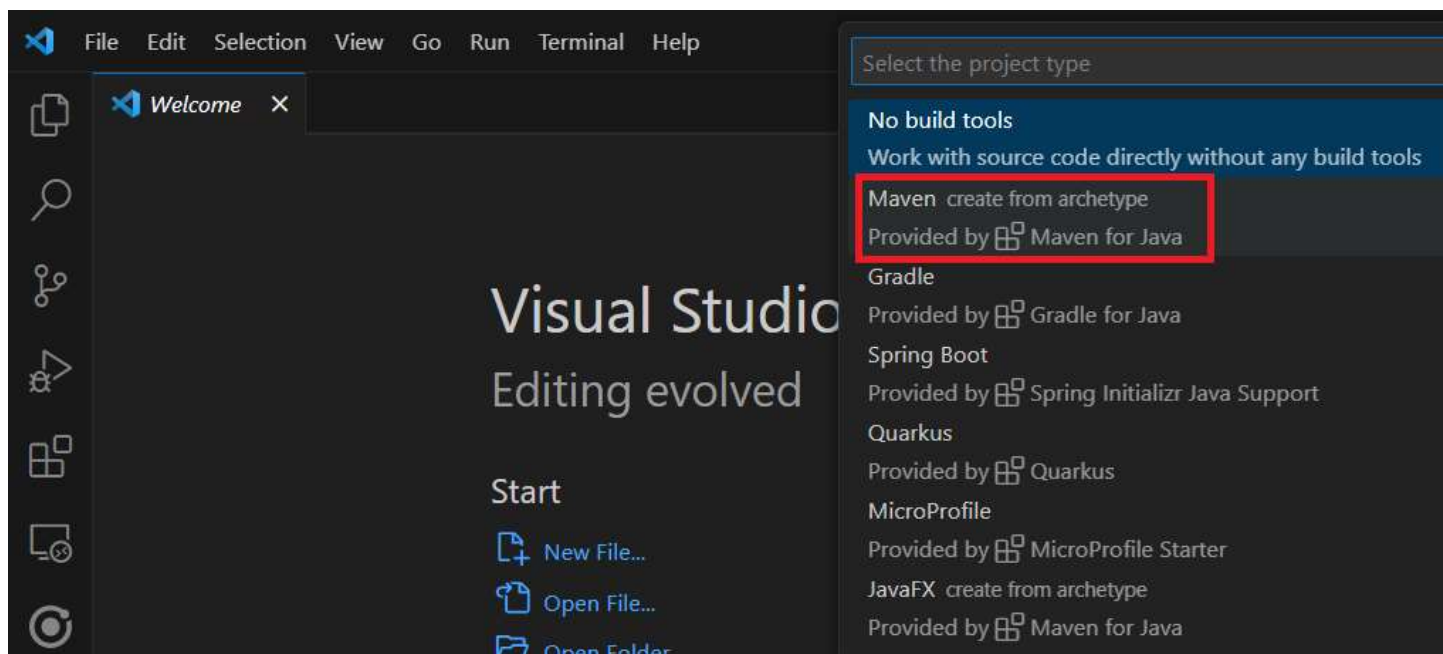
Open VSCode



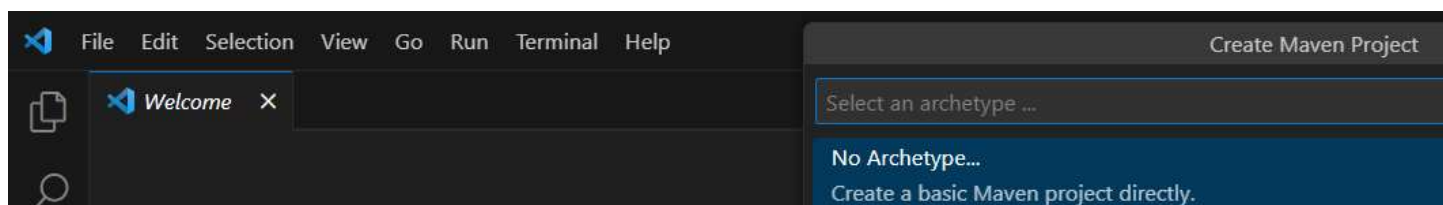
Press the keys **Ctrl+Shift+P** for creating a new Java application



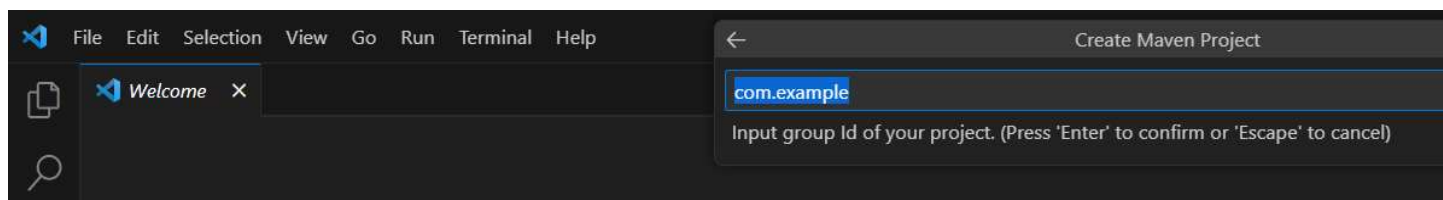
We select **Maven** project type



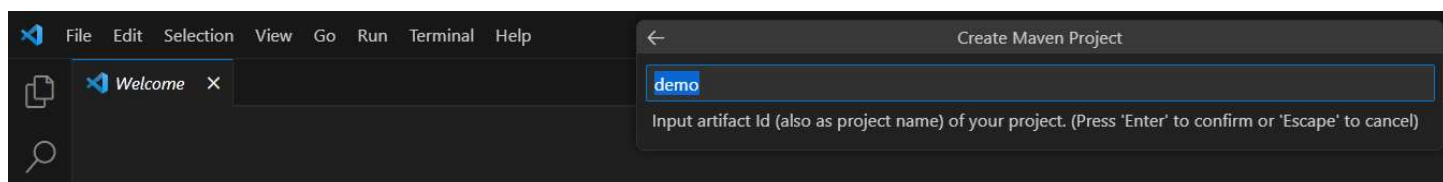
We select **No Archetype**



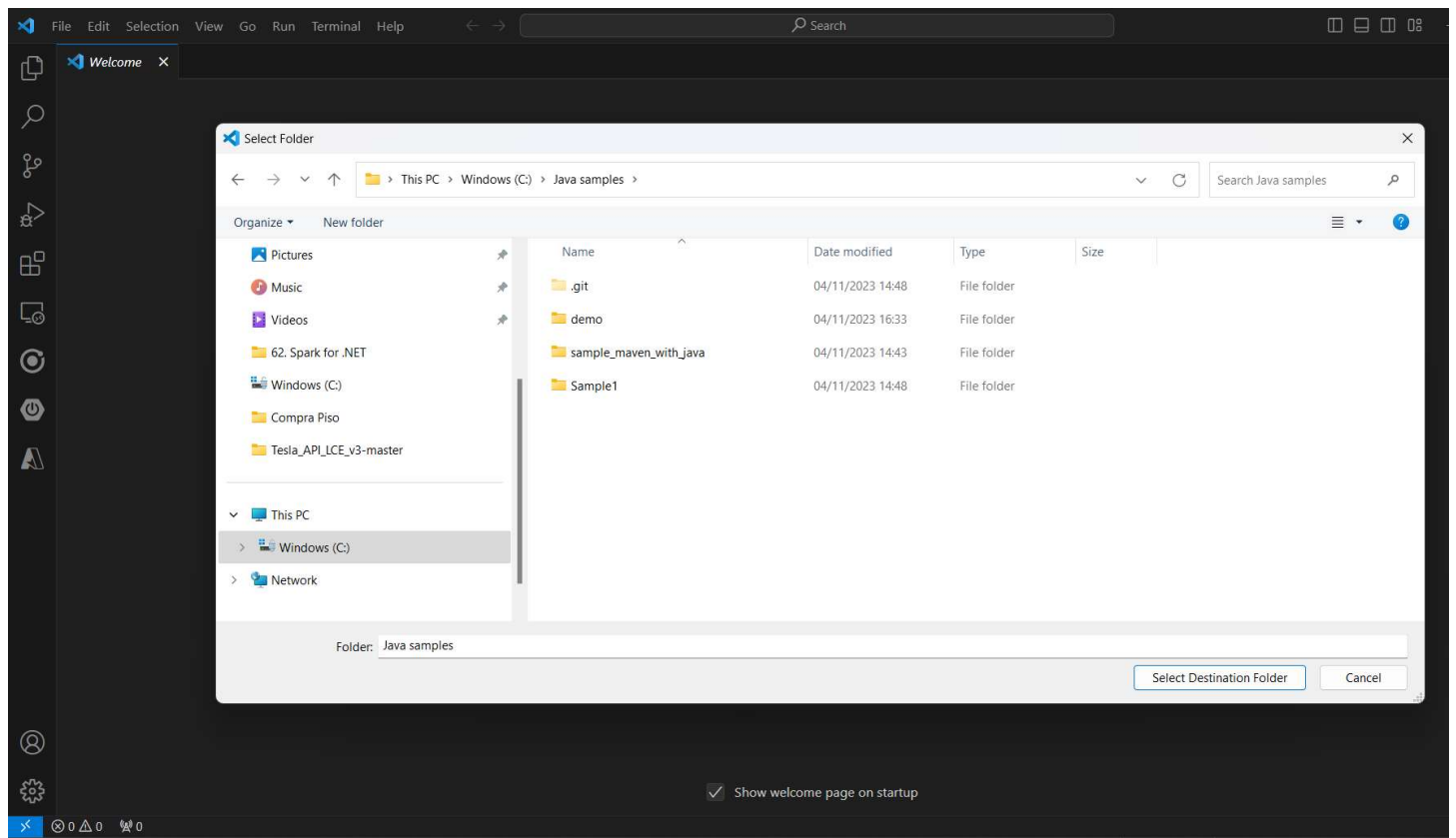
We set the package name **com.example**



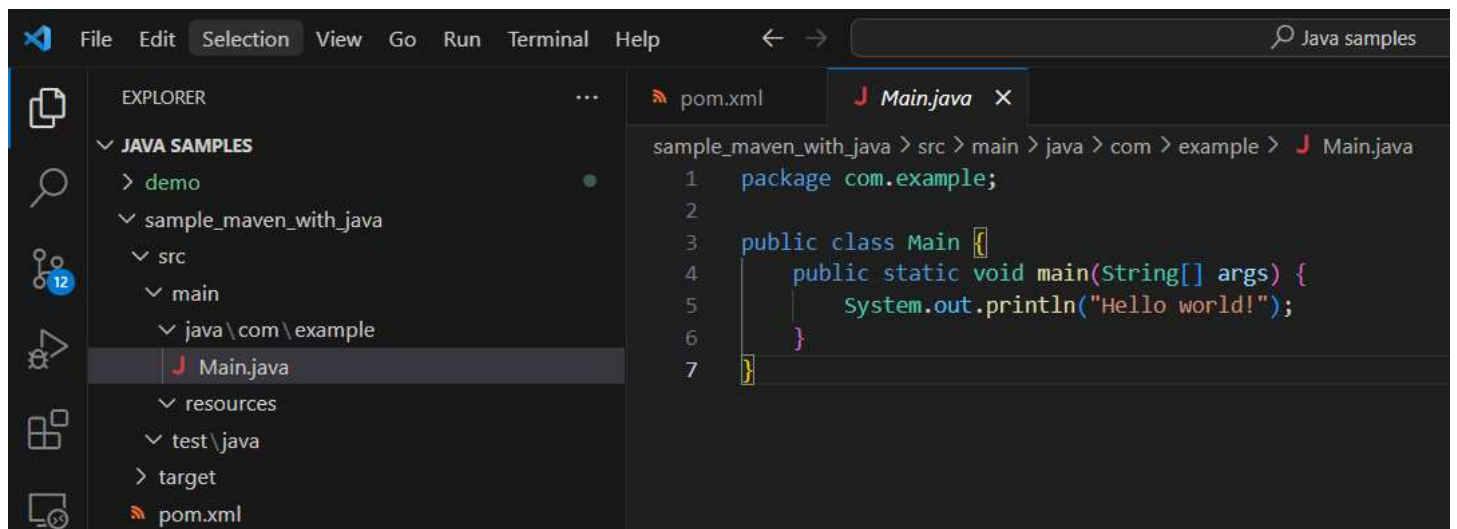
We set the application name



We select the folder where to place the new Java application



We open the new Java application



The original **pom.xml** file not include the tag. We will add this tag in the following sections below

```
<?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 http://maven.apache.org/xsd/mav
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>sample_maven_with_java</artifactId>
  <version>1.0-SNAPSHOT</version>
```

```

<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
</project>

```

Edit the **pom.xml** file and add the following code:

```

<?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 http://maven.apache.org/xsd/mav
<modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>sample_maven_with_java</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <build>
    <plugins>
      <plugin>
        <!-- Maven JAR Plugin Configuration -->
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.1.0</version>
        <configuration>
          <archive>
            <manifest>
              <mainClass>com.example.Main</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>

      <plugin>
        <!-- Exec Maven Plugin Configuration -->
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.1.0</version>
        <configuration>
          <mainClass>com.example.Main</mainClass>
        </configuration>
        <executions>
          <execution>
            <goals>

```

```

<goal>java</goal>
</goals>
</execution>
</executions>
</plugin>
</plugins>
</build>
</project>

```

IMPORTANT Do not forget to Navigate to the new project directory, and then compile it with the command:

```
mvn clean install
```

The screenshot shows an IDE with a project named 'DEMO'. The Explorer view on the left shows the project structure: .vscode, src, resources, test, target, and pom.xml. The src/main/java/com/example directory is expanded, showing Main.java. The Main.java file is open in the editor, showing the following code:

```

1 package com.example;
2
3 public class Main {
4     public static void main(String[] args) {
5         System.out.println(x:"Hello world!");
6     }
7 }

```

The Terminal view at the bottom shows the output of the command 'mvn clean install' executed in the directory 'C:\Java samples\demo'. The output includes the following information:

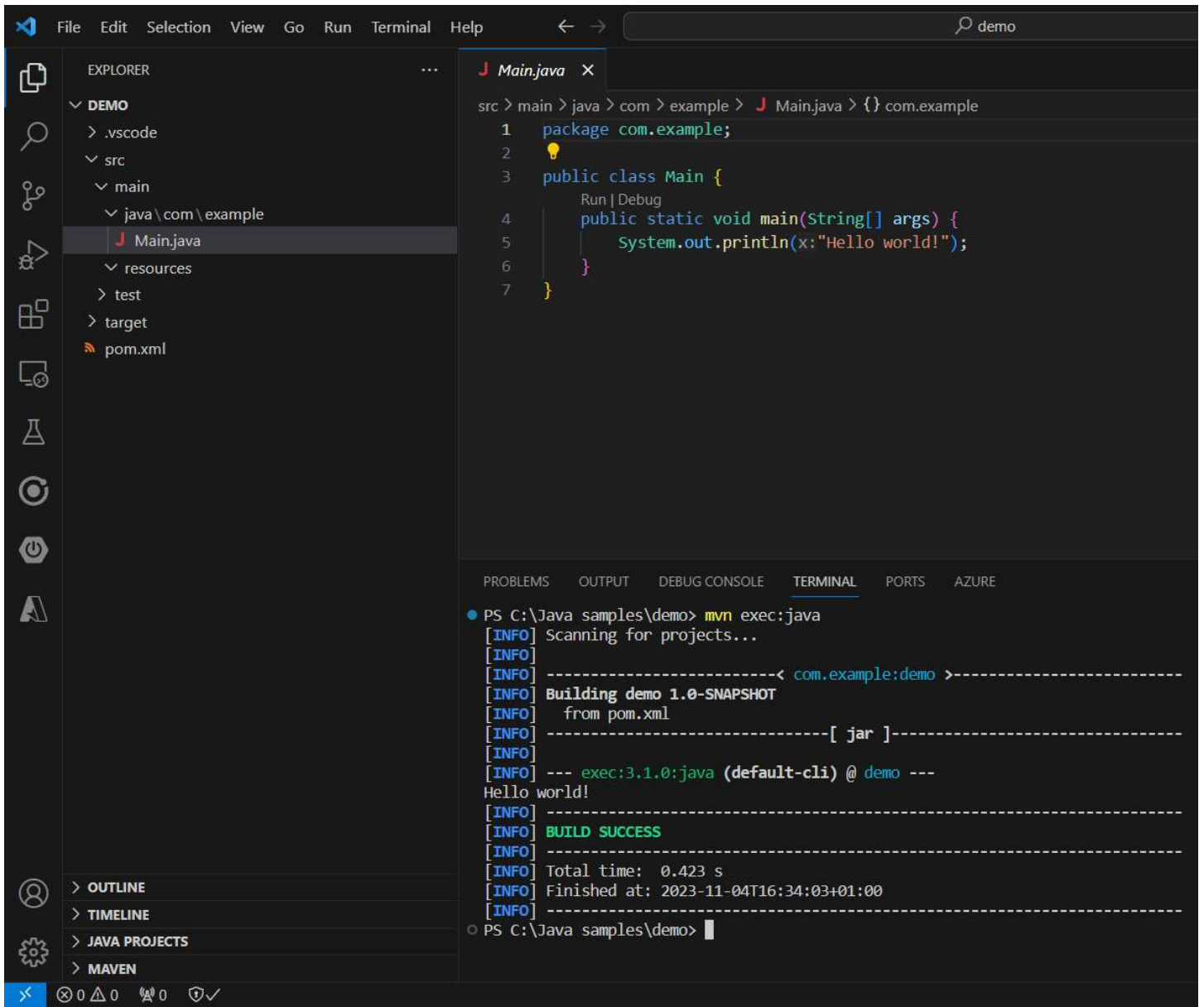
```

PS C:\Java samples\demo> mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:demo >-----
[INFO] Building demo 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ demo ---
[INFO] Deleting C:\Java samples\demo\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ demo ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ demo ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ demo ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Java samples\demo\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ demo ---
[INFO] Changes detected - recompiling the module! :dependency
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ demo ---
[INFO]
[INFO] --- jar:3.1.0:jar (default-jar) @ demo ---

```

To run the application execute the command:

```
mvn exec:java
```



3. Input the application source code for Azure Authentication and for creating the new Azure ResourceGroup

For more advance examples navigate to the URL: <https://github.com/Azure-Samples/resources-java-manage-resource/tree/master>

In the `main.java` file include this code:

```
package com.example;

import com.azure.core.credential.TokenCredential;
import com.azure.core.management.AzureEnvironment;
import com.azure.core.management.profile.AzureProfile;
import com.azure.identity.DefaultAzureCredentialBuilder;
import com.azure.resourcemanager.AzureResourceManager;
import com.azure.core.management.Region;

public class Main {
```



```

public static void main(String[] args) {

    //=====
    // Azure SDK for Java: Azure Authentication
    //=====

    AzureProfile profile = new AzureProfile(AzureEnvironment.AZURE);
    TokenCredential credential = new DefaultAzureCredentialBuilder()
        .authorityHost(profile.getEnvironment().getActiveDirectoryEndpoint())
        .build();
    AzureResourceManager azureResourceManager = AzureResourceManager
        .authenticate(credential, profile)
        .withDefaultSubscription();

    //=====
    // Azure SDK for Java: Create a resource group
    //=====

    final String rgName = "myFirstResourceGroup";
    System.out.println("Creating a resource group with name: " + rgName);

    azureResourceManager.resourceGroups().define(rgName)
        .withRegion(Region.US_WEST)
        .create();
}
}

```

4. Load the Azure SDK for Java dependencies/libraries

For more info about the dependencies see:

Azure Identity: <https://github.com/Azure/azure-sdk-for-java/tree/main/sdk/identity/azure-identity>

Azure ResourceManager: <https://github.com/Azure/azure-sdk-for-java/tree/main/sdk/resourcemanager>

Edit the **pom.xml** file and add the Azure SDK dependencies/libraries

```

<?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 http://maven.apache.org/xsd/mav
<modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>

```

```
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
</properties>

<dependencies>
  <!-- Azure SDK Dependencies -->
  <dependency>
    <groupId>com.azure</groupId>
    <artifactId>azure-identity</artifactId>
    <version>1.11.1</version>
  </dependency>
  <dependency>
    <groupId>com.azure.resourcemanager</groupId>
    <artifactId>azure-resourcemanager</artifactId>
    <version>2.33.0</version>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <!-- Maven JAR Plugin Configuration -->
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>3.1.0</version>
      <configuration>
        <archive>
          <manifest>
            <mainClass>com.example.Main</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>

    <plugin>
      <!-- Exec Maven Plugin Configuration -->
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>exec-maven-plugin</artifactId>
      <version>3.1.0</version>
      <configuration>
        <mainClass>com.example.Main</mainClass>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>java</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

```
</project>
```

5. Build and run the application

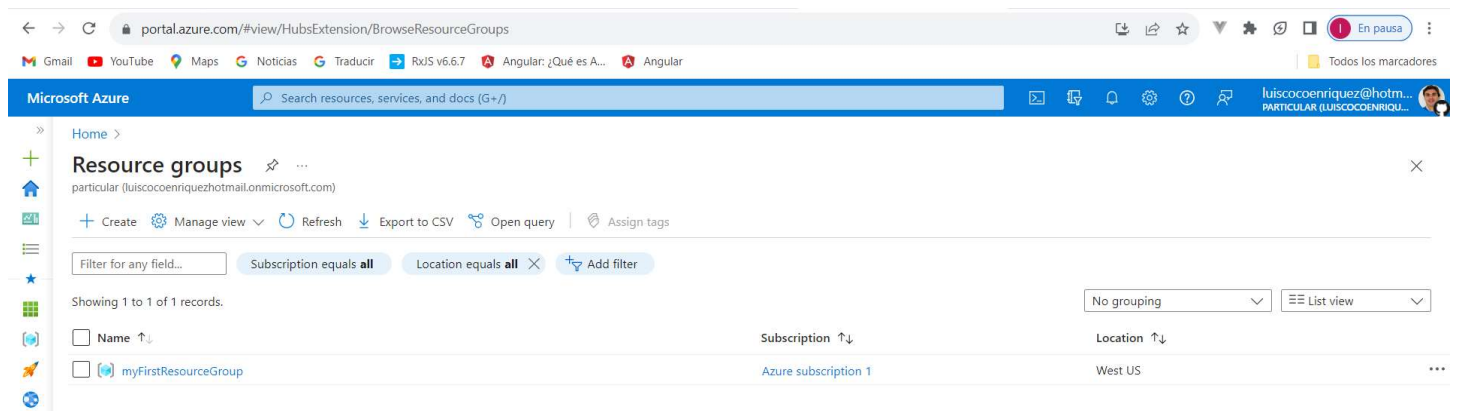
Open the `main.java` file in VSCode and then compile the application with the command:

```
mvn clean install
```

and then run the application with the command:

```
mvn exec:java
```

6. Open Azure portal and see the new ResourceGroup



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information. The main content area is titled 'Resource groups' and shows a list of resource groups. The table below displays the following information:

| Name | Subscription | Location |
|----------------------|----------------------|----------|
| myFirstResourceGroup | Azure subscription 1 | West US |