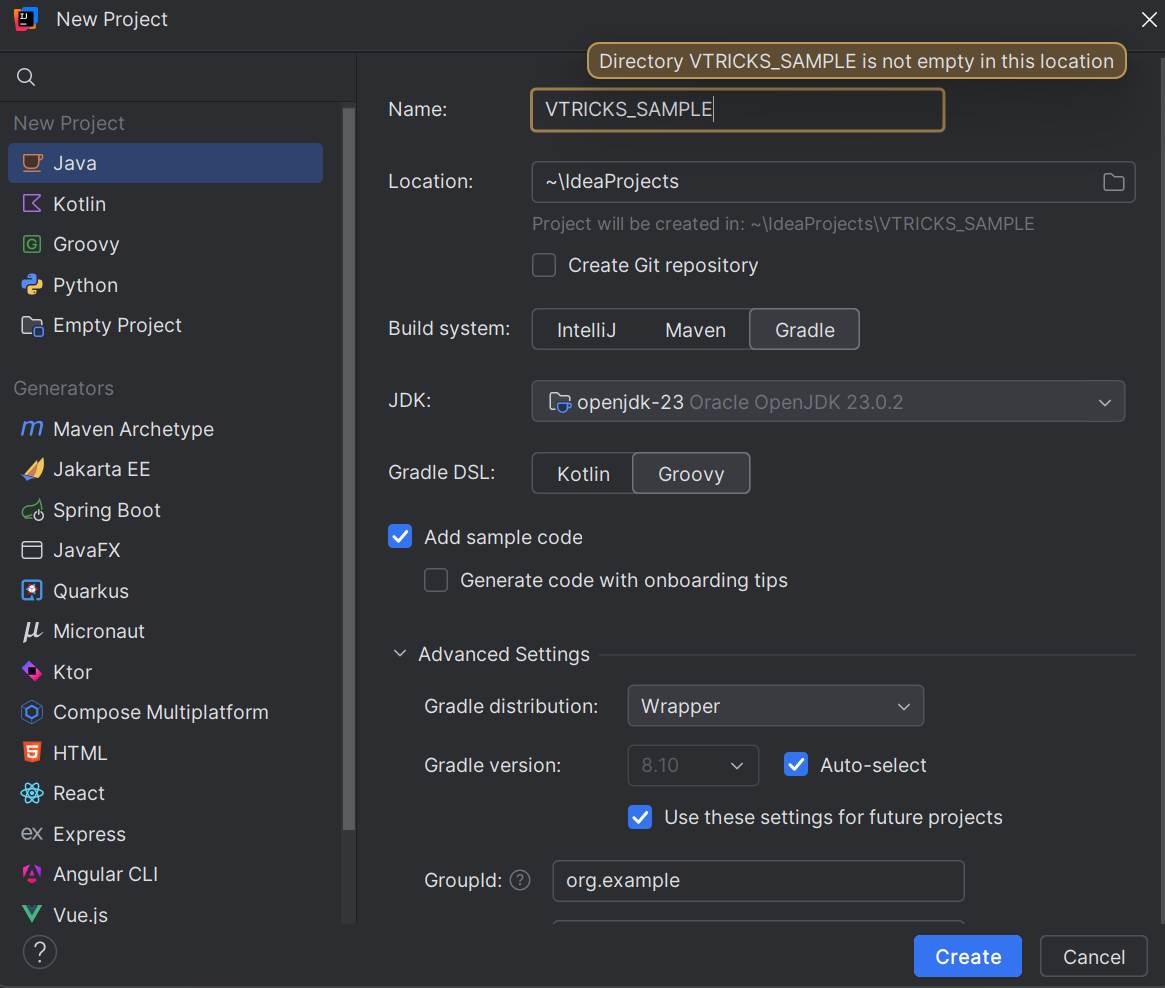
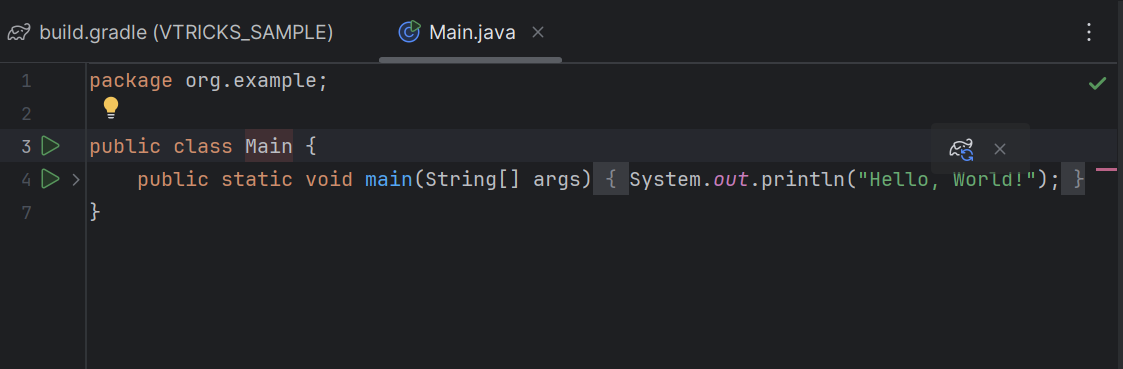
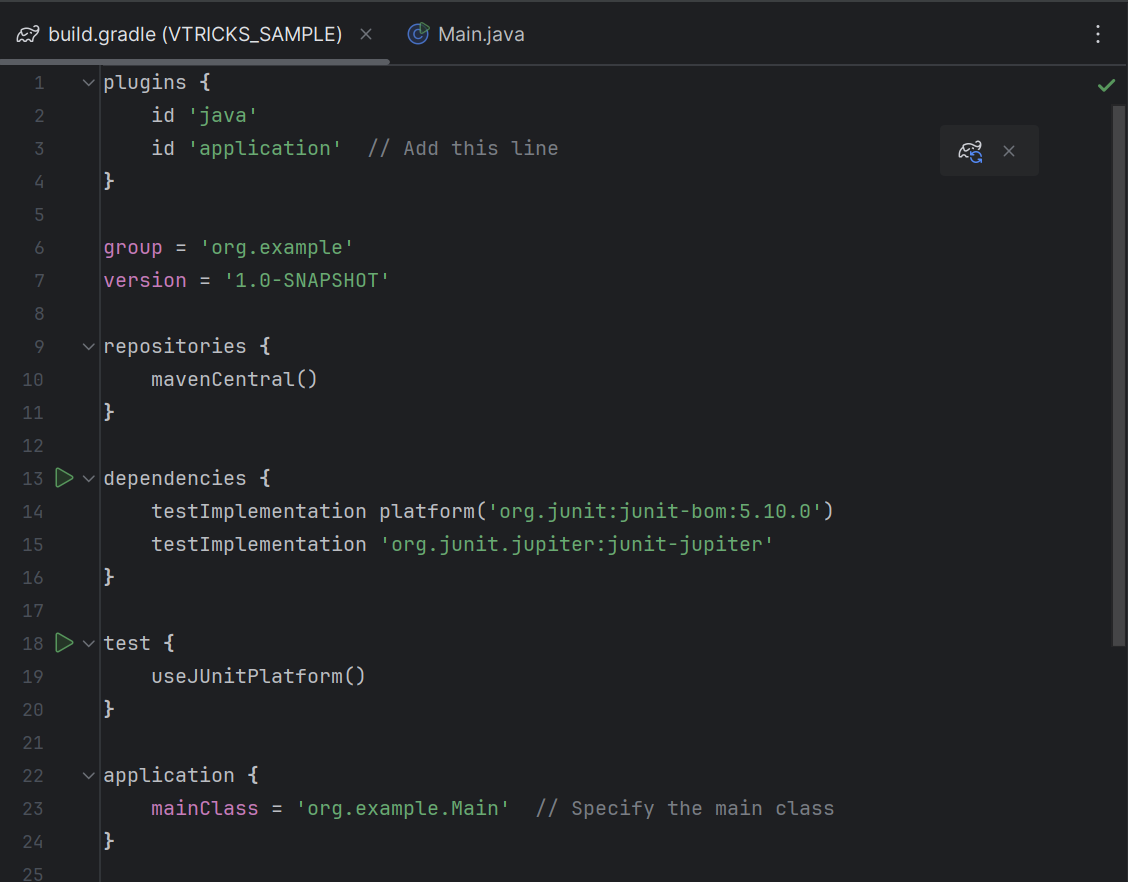
Choose anew project provide name of the project, choose build system as the Gradle and Gradle DSL as Groovy then click on the create.



You will be getting a default main.java file



And build.gradle file



After adding the gradle file we need to run ./gradlew clean then we need to run the ./gradlew run commands then we will be having a jar file craeted at target folder



Need to run for the java -jar jar file

Build.gradle file

plugins {

id 'java'

id 'application' // Add this line

}

group = 'org.example'

version = '1.0-SNAPSHOT'

repositories {

mavenCentral()

}

dependencies {

testImplementation platform('org.junit:junit-bom:5.10.0')

testImplementation 'org.junit.jupiter:junit-jupiter'

}

test {

useJUnitPlatform()

}

application {

mainClass = 'org.example.Main' // Specify the main class

}

jar {

manifest {

attributes(

'Main-Class': 'org.example.Main' // Specify the main class here as well

)

}

}

**Plugins**:

* 1. java: For Java projects.
  2. application: To specify the mainClass for running the app.

**Group & Version**:

* 1. Sets group to org.example.
  2. Sets version to 1.0-SNAPSHOT.

**Repositories**:

* 1. Uses **Maven Central** to fetch dependencies.

**Dependencies**:

* 1. Adds **JUnit 5** (junit-bom:5.10.0) for testing.

**Test Configuration**:

* 1. Uses JUnit 5 for running tests.

**Application Plugin**:

* 1. Defines the mainClass as org.example.Main.

**JAR Configuration**:

* 1. Sets the Main-Class attribute in the **JAR manifest** to enable running it via java -jar.

### ****Build & Run Instructions****

#### ****1. Build the Project****

gradle build

#### ****2. Run the Application****

gradle run

#### ****3. Create an Executable JAR****

gradle jar

#### ****4. Run the JAR File****

java -jar build/libs/project-name-1.0-SNAPSHOT.jar