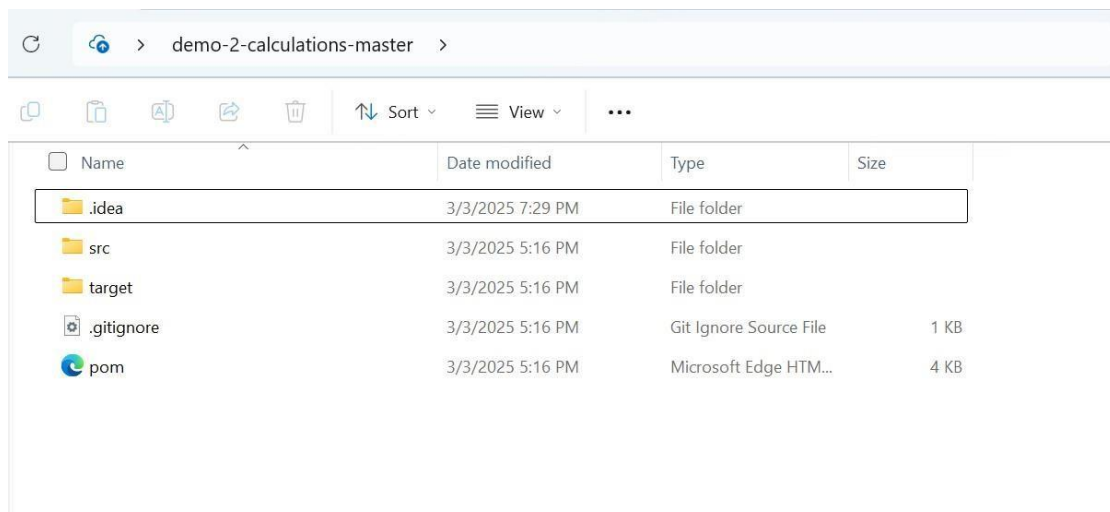


Open the folder that contains the maven project, and there you need to open the command prompt



Open the command prompt here,

```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell\Desktop\demo-2-calculations-master>gradle init
Starting a Gradle Daemon (subsequent builds will be faster)

Found a Maven build. Generate a Gradle build from this? (default: yes) [yes, no] yes

Select build script DSL:
 1: Kotlin
 2: Groovy
Enter selection (default: Kotlin) [1..2] 2

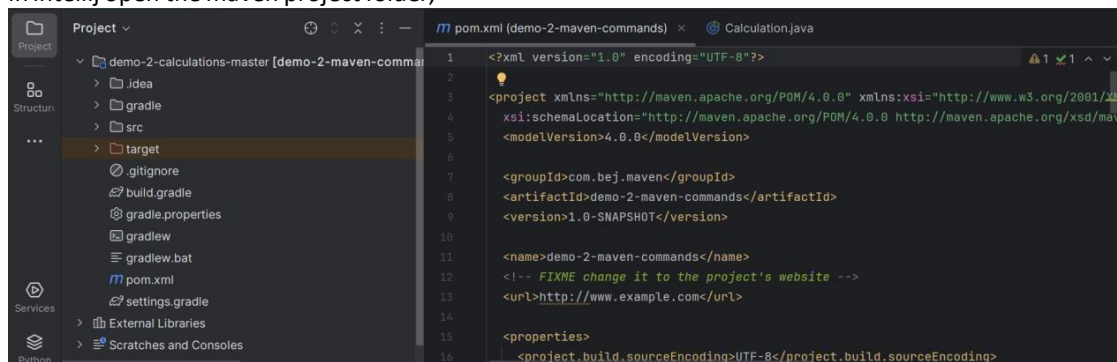
Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no] no

> Task :init
Maven to Gradle conversion is an incubating feature.
For more information, please refer to https://docs.gradle.org/8.13/userguide/migrating_from_maven.html in the Gradle documentation.

BUILD SUCCESSFUL in 1m 28s
1 actionable task: 1 executed
C:\Users\Dell\Desktop\demo-2-calculations-master>
```

Here we need to initialize the gradle, then choose to build gradle project from maven project, and choose the DSL Script as Groovy, then for to build the generate build using new API choose no then the maven to gradle conversion will be successful

In IntelliJ open the maven project folder,



Here we can see the in the maven project the build.gradle fill will be added.

```

application {
    mainClass = 'com.bej.maven.Calculation' // Specify the main class
}

jar {
    manifest {
        attributes(
            'Main-Class': 'com.bej.maven.Calculation' // Specify the main class here
        )
    }
}

```

After that build the file using ./gradlew build

```

BUILD SUCCESSFUL in 2s
1 actionable task: 1 executed
Configuration cache entry stored.
PS C:\Users\Dell\Desktop\demo-2-calculations-master> ./gradlew build
Calculating task graph as no cached configuration is available for tasks: build

BUILD SUCCESSFUL in 6s
7 actionable tasks: 7 executed
Configuration cache entry stored.
PS C:\Users\Dell\Desktop\demo-2-calculations-master> cd build
PS C:\Users\Dell\Desktop\demo-2-calculations-master\build> cd libs
PS C:\Users\Dell\Desktop\demo-2-calculations-master\build\libs> ls

Directory: C:\Users\Dell\Desktop\demo-2-calculations-master\build\libs

Mode                LastWriteTime         Length Name
----                -
-a----             3/3/2025   8:04 PM           1752 demo-2-maven-commands-1.0-SNAPSHOT.jar

```

So, after building test the jar file using the java -jar demo-2-maven-commands-1.0-SNAPSHOT.jar

The jar file will be stored in /demo-2-calculations-master-build/libs folder

```

Services Alt+8
PS C:\Users\Dell\Desktop\demo-2-calculations-master\build\libs> java -jar demo-2-maven-commands-1.0-SNAPSHOT.jar
Enter the numbers:
5
4
Select the operator 1.addition2.subtraction3.multiplication4.division
4
Results: 1
PS C:\Users\Dell\Desktop\demo-2-calculations-master\build\libs>

```

Build.gradle file

```
plugins {  
    id 'java-library'  
    id 'maven-publish'  
    id 'application' // Add this line  
}  
  
repositories  
{  
    mavenLocal()  
    maven {  
        url = uri('https://repo.maven.apache.org/maven2/')  
    }  
}  
  
dependencies {  
    testImplementation libs.org.junit.jupiter.junit.jupiter.api  
}  
  
group = 'org.example'  
version = '1.0-SNAPSHOT'  
description = 'Raga_demo'  
java.sourceCompatibility = JavaVersion.VERSION_17  
  
publishing  
{  
    publications  
    {  
        maven(MavenPublication)  
        { from(components.java)  
        }  
    }  
}  
  
tasks.withType(JavaCompile)
```

```
{ options.encoding = 'UTF-8'
}

tasks.withType(Javadoc)

{ options.encoding = 'UTF-8'
}

application {

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

jar {

    manifest

    { attributes

        (

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

        )

    }

}
```