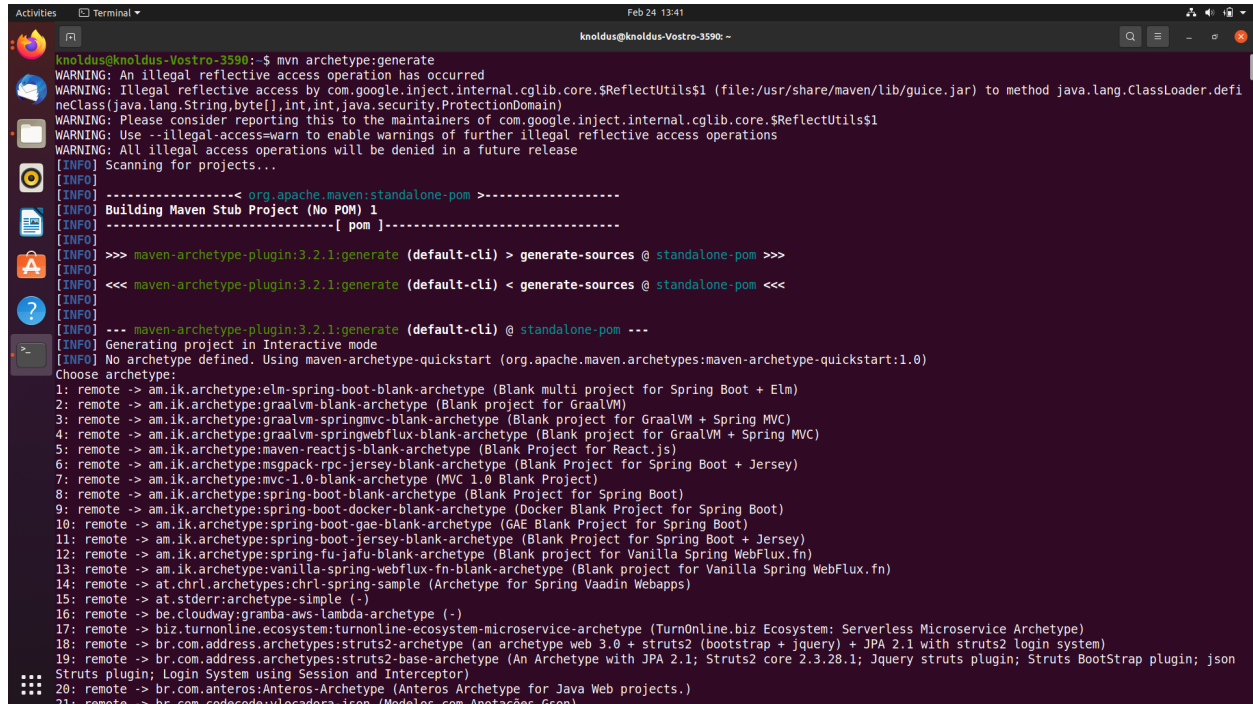


# Maven Assignment

**Step 1:** Create the parent directory by using command:  
**mvn archetype: generate**



```
knoldus@knoldus-Vostro-3590:~$ mvn archetype:generate
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
>>> mvn-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
<<< mvn-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- mvn-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: remote -> am.ik.archetype:elm-spring-boot-blank-archetype (Blank multi project for Spring Boot + Elm)
2: remote -> am.ik.archetype:graalvm-blank-archetype (Blank project for GraalVM)
3: remote -> am.ik.archetype:graalvm-springmvc-blank-archetype (Blank project for GraalVM + Spring MVC)
4: remote -> am.ik.archetype:graalvm-springwebflux-blank-archetype (Blank project for GraalVM + Spring MVC)
5: remote -> am.ik.archetype:maven-reactjs-blank-archetype (Blank Project for React.js)
6: remote -> am.ik.archetype:msgpack-rpc-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
7: remote -> am.ik.archetype:mvc-1.0-blank-archetype (MVC 1.0 Blank Project)
8: remote -> am.ik.archetype:spring-boot-blank-archetype (Blank Project for Spring Boot)
9: remote -> am.ik.archetype:spring-boot-docker-blank-archetype (Docker Blank Project for Spring Boot)
10: remote -> am.ik.archetype:spring-boot-gae-blank-archetype (GAE Blank Project for Spring Boot)
11: remote -> am.ik.archetype:spring-boot-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
12: remote -> am.ik.archetype:spring-fu-jafu-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
13: remote -> am.ik.archetype:vanilla-spring-webflux-fn-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
14: remote -> at.chrl.archetypes:chrl-spring-sample (Archetype for Spring Vaadin Webapps)
15: remote -> at.stderri:archetype-simple (-)
16: remote -> be.cloudway:gramba-aws-lambda-archetype (-)
17: remote -> biz.turnonline.ecosystem:turnonline-ecosystem-microservice-archetype (TurnOnline.biz Ecosystem: Serverless Microservice Archetype)
18: remote -> br.com.address.archetypes:struts2-archetype (an archetype web 3.0 + struts2 (bootstrap + jquery) + JPA 2.1 with struts2 login system)
19: remote -> br.com.address.archetypes:struts2-base-archetype (An Archetype with JPA 2.1; Struts2 core 2.3.28.1; JQuery struts plugin; Struts Bootstrap plugin; json Struts plugin; Login System using Session and Interceptor)
20: remote -> br.com.anteros:Anteros-Archetype (Anteros Archetype for Java Web projects.)
21: remote -> br.com.codercode:ultradora-ison (Modelos com Antares Gson)
```

**Step 2:** Now there will be options visible on the screen among which we need to select the appropriate ones.

**select archetype** - choose a template for the project

**choose pom version** - any version of pom can be selected

**choose group id** - it is a unique id which contains organization name.

**choose archetype** - provide a name for your project

**choose package** - a name of package has to be provided

```
Activities Terminal Feb 24 13:42 knoldus@knoldus-Vostro-3590: ~
3225: remote -> za.co.absa.hyperdrive:component-archetype 2.11 (-)
3226: remote -> za.co.absa.hyperdrive:component-archetype 2.12 (-)
Choose a number or apply filter (format: [groupId]:artifactId, case sensitive contains): 2007: 2007
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Define value for property 'groupId': com.knoldus
Define value for property 'artifactId': stringmanipulation
Define value for property 'version' 1.0-SNAPSHOT: : 1.0-SNAPSHOT
Define value for property 'package' com.knoldus: : com.knoldus
Confirm properties configuration:
groupId: com.knoldus
artifactId: stringmanipulation
version: 1.0-SNAPSHOT
package: com.knoldus
Y: : Y
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: stringmanipulation
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: packageInPathFormat, Value: com/knoldus
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: stringmanipulation
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/knoldus/stringmanipulation
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:24 min
[INFO] Finished at: 2023-02-24T13:42:12+05:30
[INFO] -----
knoldus@knoldus-Vostro-3590: ~ $
```

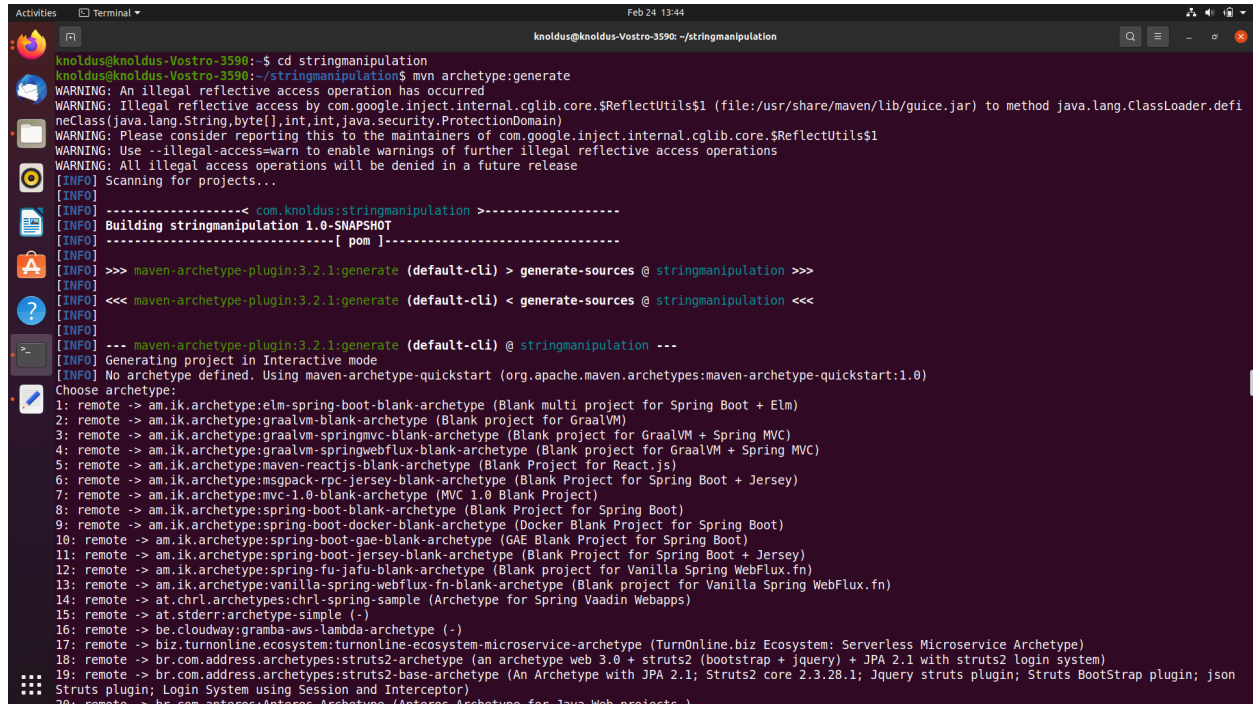
## Step 3: Edit the pom.xml file to add packaging to pom

pom indicates that it is the parent directory

```
Activities Text Editor Feb 24 13:43 pom.xml - /home/knoldus/stringmanipulation
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.knoldus</groupId>
8   <artifactId>stringmanipulation</artifactId>
9   <version>1.0-SNAPSHOT</version>
10  <packaging>pom</packaging>
11  <name>stringmanipulation</name>
12  <!-- FIXME change it to the project's website -->
13  <url>http://www.example.com</url>
14
15  <properties>
16    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17    <maven.compiler.source>1.7</maven.compiler.source>
18    <maven.compiler.target>1.7</maven.compiler.target>
19  </properties>
20
21  <dependencies>
22    <dependency>
23      <groupId>junit</groupId>
24      <artifactId>junit</artifactId>
25      <version>4.11</version>
26      <scope>test</scope>
27    </dependency>
28  </dependencies>
29
30  <build>
31    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
32    <plugins>
33      <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -->
34      <plugin>
35        <artifactId>maven-clean-plugin</artifactId>
36        <version>3.1.0</version>
37      </plugin>
38      <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
39      <plugin>
40        <artifactId>maven-resources-plugin</artifactId>
41        <version>3.0.2</version>
42      </plugin>
43      <plugin>
44        <artifactId>maven-compiler-plugin</artifactId>
45        <version>3.8.0</version>
46      </plugin>
47      <plugin>
48        <artifactId>maven-surefire-plugin</artifactId>
49        <version>2.22.1</version>
50      </plugin>
51      <plugin>
52        <artifactId>maven-jar-plugin</artifactId>
53        <version>3.1.0</version>
54      </plugin>
55    </plugins>
56  </build>
57</project>
```

**Step 4:** Enter into the parent directory and create a one more directory using the command

**mvn archetype: generate**

A terminal window showing the execution of the Maven command 'mvn archetype:generate'. The terminal output includes several warnings about illegal reflective access operations. It then shows the scanning for projects, building 'stringmanipulation 1.0-SNAPSHOT', and the Maven archetype plugin starting the generation process. A list of available archetypes is displayed, including options for Spring Boot, GraalVM, and other frameworks. The user is prompted to choose an archetype.

```
knoldus@knoldus-Vostro-3590:~$ cd stringmanipulation
knoldus@knoldus-Vostro-3590:~/stringmanipulation$ mvn archetype:generate
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.knoldus:stringmanipulation >-----
[INFO] Building stringmanipulation 1.0-SNAPSHOT
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ stringmanipulation >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ stringmanipulation <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ stringmanipulation ---
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: remote -> am.ik.archetype:elm-spring-boot-blank-archetype (Blank multi project for Spring Boot + Elm)
2: remote -> am.ik.archetype:graalvm-blank-archetype (Blank project for GraalVM)
3: remote -> am.ik.archetype:graalvm-springmvc-blank-archetype (Blank project for GraalVM + Spring MVC)
4: remote -> am.ik.archetype:graalvm-springwebflux-blank-archetype (Blank project for GraalVM + Spring MVC)
5: remote -> am.ik.archetype:maven-reactjs-blank-archetype (Blank Project for React.js)
6: remote -> am.ik.archetype:msgpack-rpc-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
7: remote -> am.ik.archetype:mvc-1.0-blank-archetype (MVC 1.0 Blank Project)
8: remote -> am.ik.archetype:spring-boot-blank-archetype (Blank Project for Spring Boot)
9: remote -> am.ik.archetype:spring-boot-docker-blank-archetype (Docker Blank Project for Spring Boot)
10: remote -> am.ik.archetype:spring-boot-gae-blank-archetype (GAE Blank Project for Spring Boot)
11: remote -> am.ik.archetype:spring-boot-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
12: remote -> am.ik.archetype:spring-fu-jafu-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
13: remote -> am.ik.archetype:vanilla-spring-webflux-fn-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
14: remote -> at.chrl.archetypes:chrl-spring-sample (Archetype for Spring Vaadin Webapps)
15: remote -> at.stder:archetype-simple (-)
16: remote -> be.cloudway:gramba-aws-lambda-archetype (-)
17: remote -> biz.turnonline.ecosystem:turnonline-ecosystem-microservice-archetype (TurnOnline.biz Ecosystem: Serverless Microservice Archetype)
18: remote -> br.com.address.archetypes:struts2-archetype (an archetype web 3.0 + struts2 (bootstrap + jquery) + JPA 2.1 with struts2 login system)
19: remote -> br.com.address.archetypes:struts2-base-archetype (An Archetype with JPA 2.1; Struts2 core 2.3.28.1; JQuery struts plugin; Struts Bootstrap plugin; json
Struts plugin; Login System using Session and Interceptor)
20: remote -> br.com.ateros.ateros-archetype (Ateros Archetype for Java Web projects)
```

**Step 5:** Now there will be options visible on the screen among which we need to select the appropriate ones.

**select archetype** - choose a template for the project

**choose pom version** - any version of pom can be selected

**choose group id** - it is a unique id which contains organization name.

**choose archetype** - provide a name for your project

**choose package** - a name of package has to be provided

```
Activities Terminal Feb 24 13:45
knoldus@knoldus-Vostro-3590: ~/stringmanipulation

3225: remote -> za.co.absa.hyperdrive:component-archetype 2.11 (-)
3226: remote -> za.co.absa.hyperdrive:component-archetype 2.12 (-)
Choose a number or apply filter (format: [groupId]:artifactId, case sensitive contains): 2007: 2007
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Define value for property 'groupId': com.knoldus
Define value for property 'artifactId': modify-strings
Define value for property 'version': 1.0-SNAPSHOT
Define value for property 'package': com.knoldus
Confirm properties configuration:
groupId: com.knoldus
artifactId: modify-strings
version: 1.0-SNAPSHOT
package: com.knoldus
Y: : Y

[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: modify-strings
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: packageInPathFormat, Value: com/knoldus
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: modify-strings
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/knoldus/stringmanipulation/modify-strings
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:45 min
[INFO] Finished at: 2023-02-24T13:45:35+05:30
[INFO] -----
knoldus@knoldus-Vostro-3590: ~/stringmanipulation$
```

**Step 6:** Edit the pom.xml file to add packaging to **pom** as we will create more modules into this.

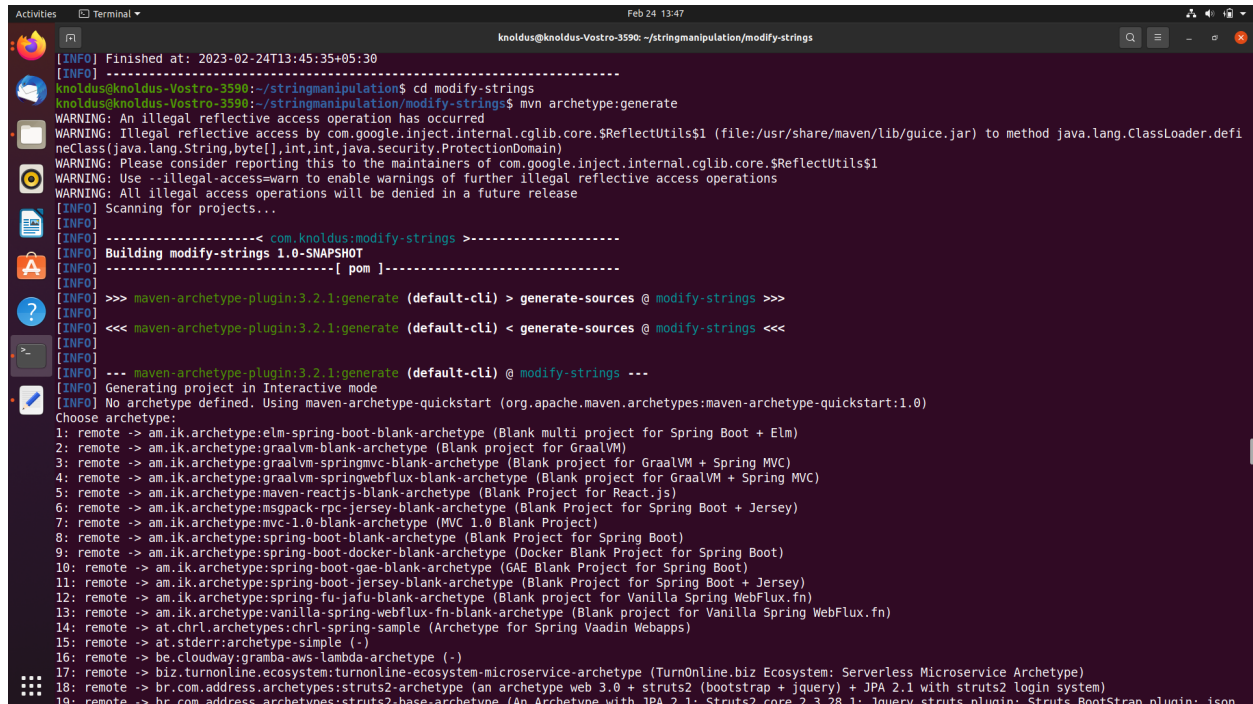
**pom** indicates that it is the parent directory

```
Activities Text Editor Feb 24 13:46
pom.xml
pom.xml

1 <?xml version="1.0" encoding="UTF-8"?>
2 <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://
3 <modelVersion>4.0.0</modelVersion>
4 <parent>
5 <artifactId>stringmanipulation</artifactId>
6 <groupId>com.knoldus</groupId>
7 <version>1.0-SNAPSHOT</version>
8 </parent>
9
10 <groupId>com.knoldus</groupId>
11 <artifactId>modify-strings</artifactId>
12 <version>1.0-SNAPSHOT</version>
13 <packaging>pom</packaging>
14 <name>modify-strings</name>
15 <!-- FIXME change it to the project's website -->
16 <url>http://www.example.com</url>
17
18 <properties>
19 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
20 <maven.compiler.source>1.7</maven.compiler.source>
21 <maven.compiler.target>1.7</maven.compiler.target>
22 </properties>
23
24 <dependencies>
25 <dependency>
26 <groupId>junit</groupId>
27 <artifactId>junit</artifactId>
28 <version>4.11</version>
29 <scope>test</scope>
30 </dependency>
31 </dependencies>
32
33 <build>
34 <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
35 <plugins>
36 <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -->
37 <plugin>
38 <artifactId>maven-clean-plugin</artifactId>
39 <version>3.1.0</version>
40 </plugin>
41 <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
42 <plugin>
43 <artifactId>maven-resources-plugin</artifactId>
44 <version>3.0.2</version>
45 </plugin>
46 <plugin>
47 <artifactId>maven-compiler-plugin</artifactId>
48 <version>3.8.0</version>
49 </plugin>
50 </plugins>
51 </build>
52
53 </project>
```

**Step 7:** Enter into the parent directory and create a two more directory using the command these directories will contain the code files.

## mvn archetype: generate



```
[INFO] Finished at: 2023-02-24T13:45:35+05:30
[INFO] -----
knoldus@knoldus-Vostro-3590:~/stringmanipulation$ cd modify-strings
knoldus@knoldus-Vostro-3590:~/stringmanipulation/modify-strings$ mvn archetype:generate
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO] -----< com.knoldus:modify-strings -----
[INFO] Building modify-strings 1.0-SNAPSHOT
[INFO] -----[ pom ]-----
[INFO] >>> mvn-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ modify-strings >>>
[INFO] <<< mvn-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ modify-strings <<<
[INFO] --- mvn-archetype-plugin:3.2.1:generate (default-cli) @ modify-strings ---
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: remote -> am.ik.archetype:elm-spring-boot-blank-archetype (Blank multi project for Spring Boot + Elm)
2: remote -> am.ik.archetype:graalvm-blank-archetype (Blank project for GraalVM)
3: remote -> am.ik.archetype:graalvm-springmvc-blank-archetype (Blank project for GraalVM + Spring MVC)
4: remote -> am.ik.archetype:graalvm-springwebflux-blank-archetype (Blank project for GraalVM + Spring MVC)
5: remote -> am.ik.archetype:maven-reactjs-blank-archetype (Blank Project for React.js)
6: remote -> am.ik.archetype:msgpack-rpc-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
7: remote -> am.ik.archetype:mvc-1.0-blank-archetype (MVC 1.0 Blank Project)
8: remote -> am.ik.archetype:spring-boot-blank-archetype (Blank Project for Spring Boot)
9: remote -> am.ik.archetype:spring-boot-docker-blank-archetype (Docker Blank Project for Spring Boot)
10: remote -> am.ik.archetype:spring-boot-gae-blank-archetype (GAE Blank Project for Spring Boot)
11: remote -> am.ik.archetype:spring-boot-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
12: remote -> am.ik.archetype:spring-fu-jafu-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
13: remote -> am.ik.archetype:vanilla-spring-webflux-fn-blank-archetype (Blank project for Vanilla Spring WebFlux.fn)
14: remote -> at.chrl.archetypes:chrl-spring-sample (Archetype for Spring Vaadin Webapps)
15: remote -> at.stderi:archetype-simple (-)
16: remote -> be.cloudway:gramba-aws-lambda-archetype (-)
17: remote -> biz.turnonline.ecosystem:turnonline-ecosystem-microservice-archetype (TurnOnline.biz Ecosystem: Serverless Microservice Archetype)
18: remote -> br.com.address.archetypes:struts2-archetype (an archetype web 3.0 + struts2 (bootstrap + jquery) + JPA 2.1 with struts2 login system)
19: remote -> br.com.address.archetypes:struts2-base-archetype (An Archetype with JPA 2.1; Struts2 core 2.3.28.1; Jquery struts plugin; Struts Bootstrap plugin; ison
```

**Step 8:** Now there will be options visible on the screen among which we need to select the appropriate ones.

**select archetype** - choose a template for the project

**choose pom version** - any version of pom can be selected

**choose group id** - it is a unique id which contains organization name.

**choose archetype** - provide a name for your project

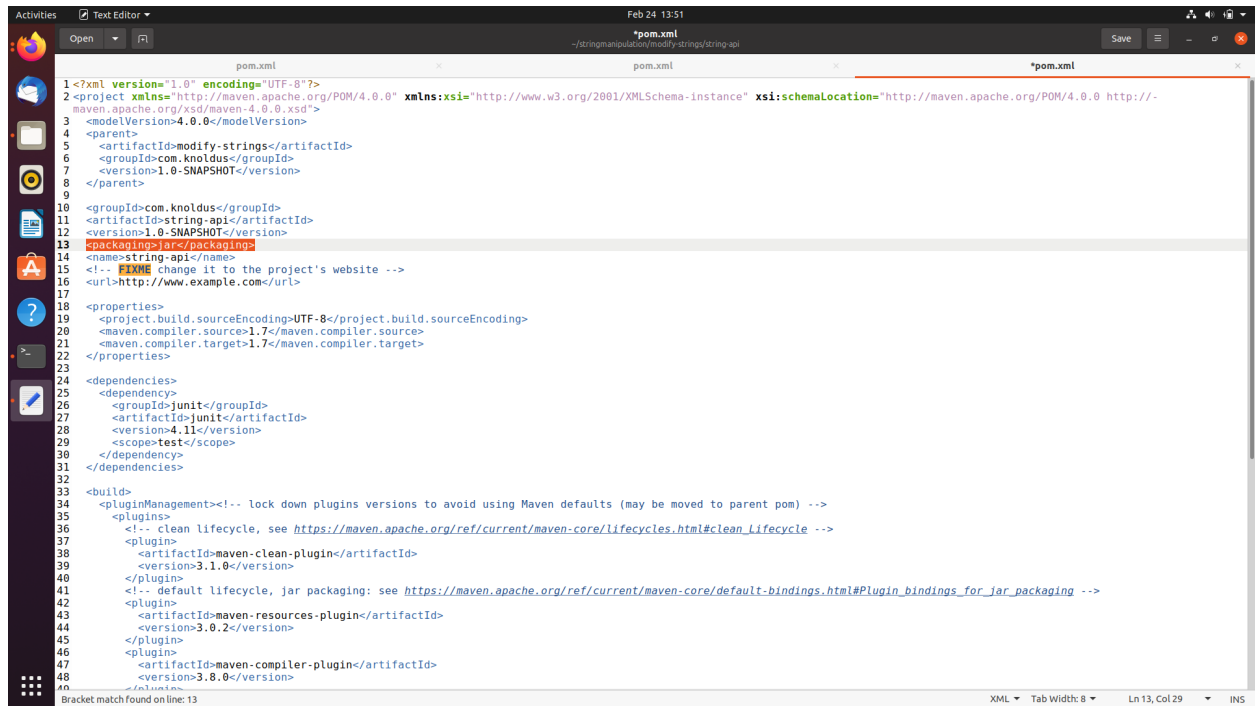
**choose package** - a name of package has to be provided

```
Activities Terminal Feb 24 13:50 knoldus@knoldus-Vostro-3590: ~/stringmanipulation/modify-strings

2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Define value for property 'groupId': com.knoldus
Define value for property 'artifactId': string-impl
Define value for property 'version' 1.0-SNAPSHOT: : 1.0-SNAPSHOT
Define value for property 'package' com.knoldus: : com.knoldus
Confirm properties configuration:
groupId: com.knoldus
artifactId: string-impl
version: 1.0-SNAPSHOT
package: com.knoldus
Y: : Y
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: string-impl
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: packageInPathFormat, Value: com/knoldus
[INFO] Parameter: package, Value: com.knoldus
[INFO] Parameter: groupId, Value: com.knoldus
[INFO] Parameter: artifactId, Value: string-impl
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/knoldus/stringmanipulation/modify-strings/string-impl
[INFO] -----
[INFO] Reactor Summary for modify-strings 1.0-SNAPSHOT:
[INFO]
[INFO] modify-strings ..... SUCCESS [01:09 min]
[INFO] string-api ..... SKIPPED
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:09 min
[INFO] Finished at: 2023-02-24T13:50:23+05:30
[INFO]
knoldus@knoldus-Vostro-3590:~/stringmanipulation/modify-strings$
```

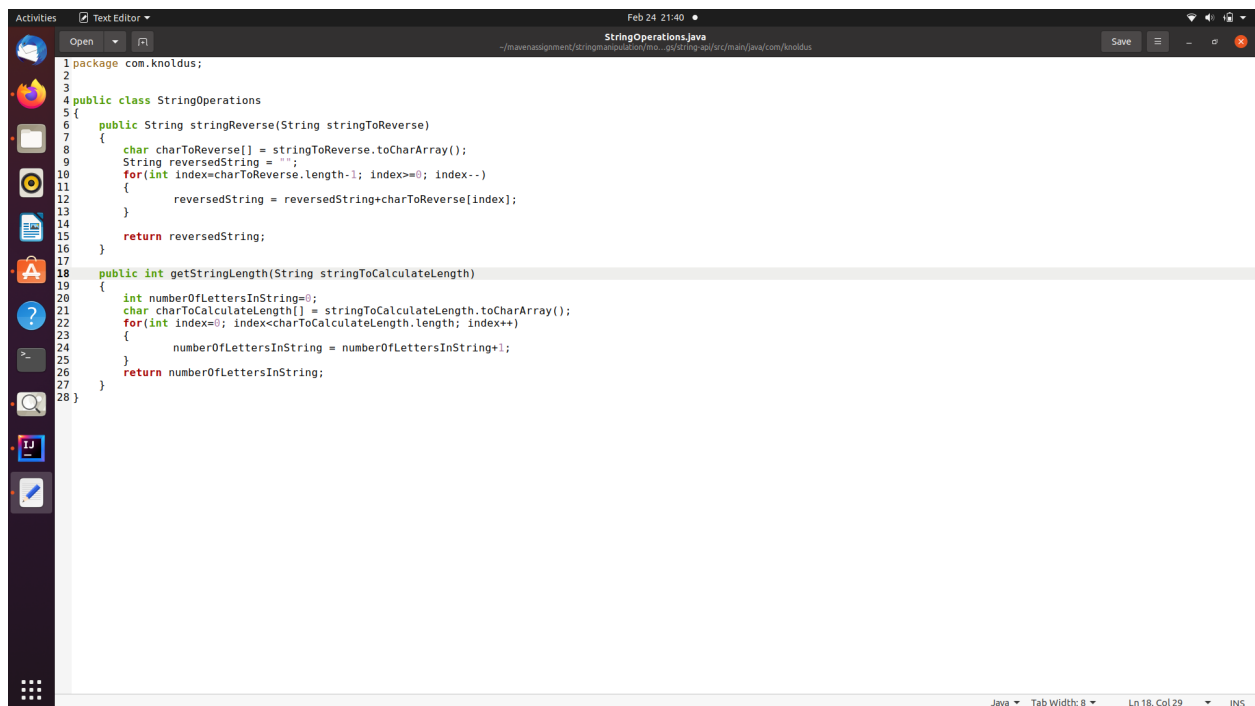
**Step 9:** Edit the pom.xml files of these modules to add packaging to jar.

**jar** - Projects with this packaging type produce a compressed zip file with the *.jar* extension. It may include pure Java classes, interfaces, resources, and metadata files.



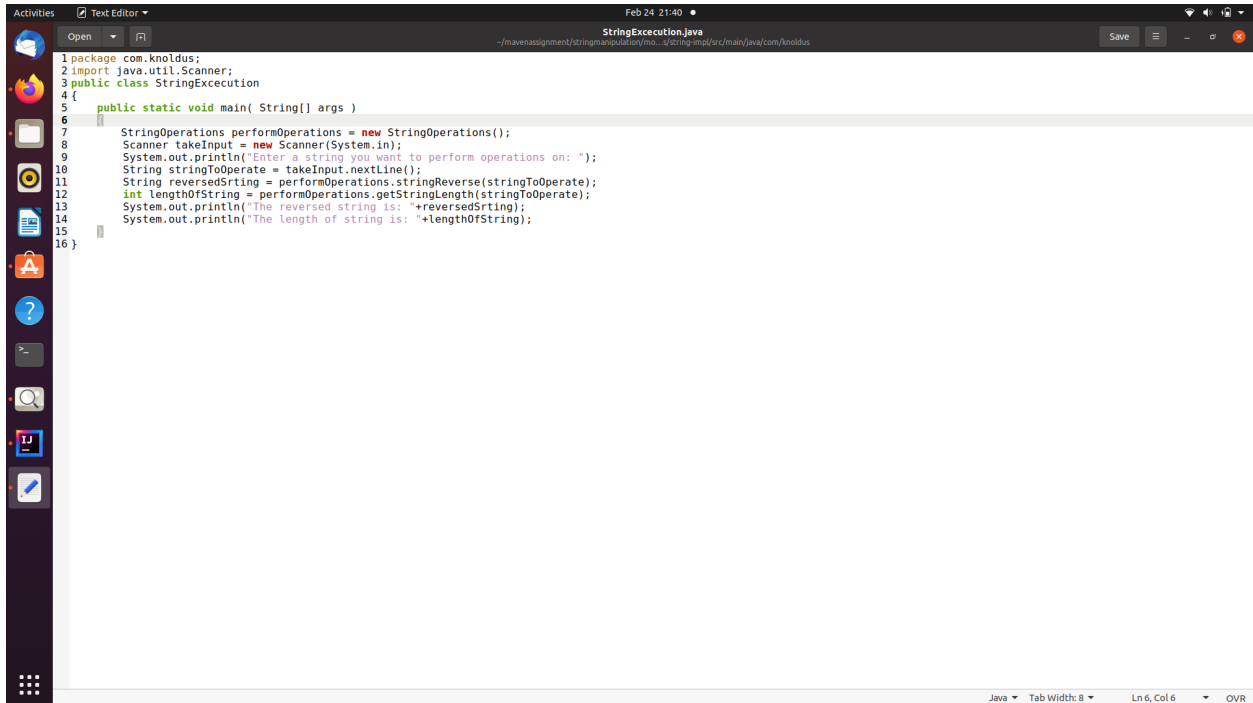
```
1<?xml version="1.0" encoding="UTF-8"?>
2<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
3  <modelVersion>4.0.0</modelVersion>
4  <parent>
5    <artifactId>modify-strings</artifactId>
6    <groupId>com.knoldus</groupId>
7    <version>1.0-SNAPSHOT</version>
8  </parent>
9
10 <groupId>com.knoldus</groupId>
11 <artifactId>string-api</artifactId>
12 <version>1.0-SNAPSHOT</version>
13 <packaging>jar</packaging>
14 <name>string-api</name>
15 <!-- FIXME change it to the project's website -->
16 <url>http://www.example.com</url>
17
18 <properties>
19   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
20   <maven.compiler.source>1.7</maven.compiler.source>
21   <maven.compiler.target>1.7</maven.compiler.target>
22 </properties>
23
24 <dependencies>
25   <dependency>
26     <groupId>junit</groupId>
27     <artifactId>junit</artifactId>
28     <version>4.11</version>
29     <scope>test</scope>
30   </dependency>
31 </dependencies>
32
33 <build>
34   <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
35   <plugins>
36     <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_lifecycle -->
37     <plugin>
38       <artifactId>maven-clean-plugin</artifactId>
39       <version>3.1.0</version>
40     </plugin>
41     <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
42     <plugin>
43       <artifactId>maven-resources-plugin</artifactId>
44       <version>3.0.2</version>
45     </plugin>
46     <plugin>
47       <artifactId>maven-compiler-plugin</artifactId>
48       <version>3.8.0</version>
49     </plugin>
50   </plugins>
51 </build>
```

**Step 10:** Now add a java file into **string-api>src>main>java>com>Knoldus** containing code of the methods.



```
1package com.knoldus;
2
3
4public class StringOperations
5{
6    public String stringReverse(String stringToReverse)
7    {
8        char charToReverse[] = stringToReverse.toCharArray();
9        String reversedString = "";
10        for(int index=charToReverse.length-1; index>=0; index--)
11        {
12            reversedString = reversedString+charToReverse[index];
13        }
14        return reversedString;
15    }
16
17    public int getStringLength(String stringToCalculateLength)
18    {
19        int numberOfLettersInString=0;
20        char charToCalculateLength[] = stringToCalculateLength.toCharArray();
21        for(int index=0; index<charToCalculateLength.length; index++)
22        {
23            numberOfLettersInString = numberOfLettersInString+1;
24        }
25        return numberOfLettersInString;
26    }
27
28 }
```

**Step 11:** Now add a java file into **string-impl>src>main>java>com>Knoldus** containing code of the main method.

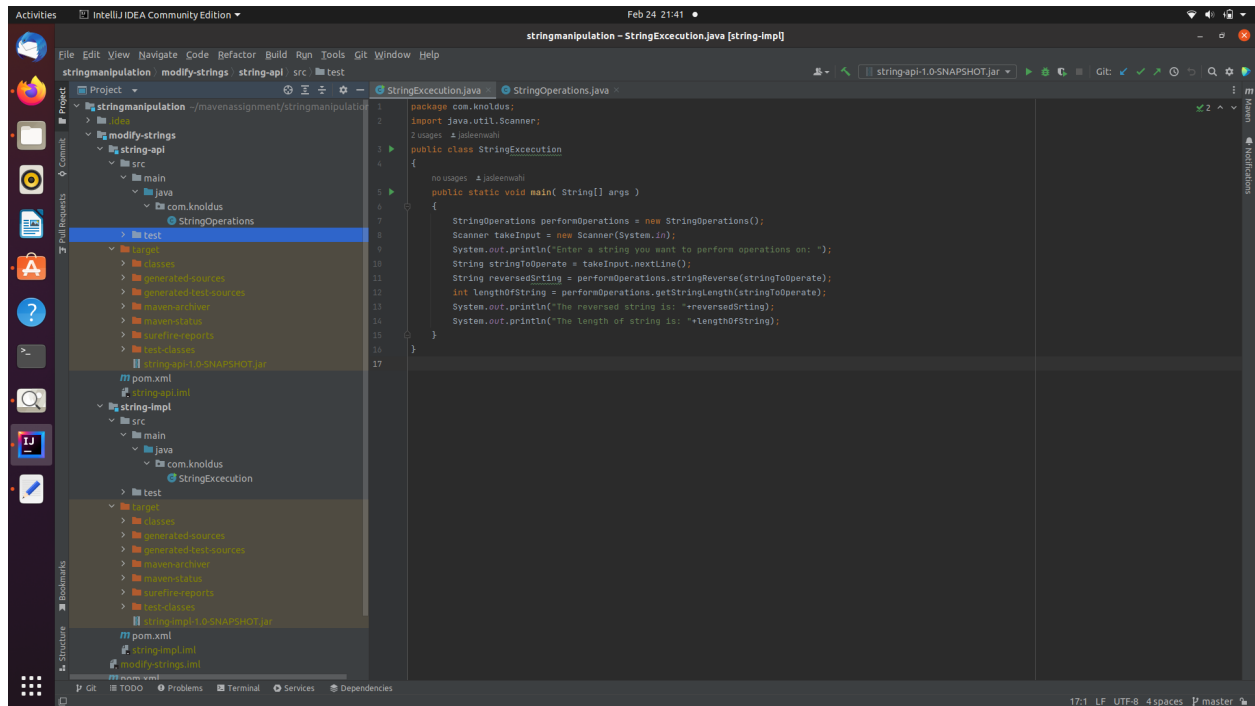
A screenshot of a text editor window titled "StringExcecution.java" showing Java code. The code defines a package "com.knoldus", imports "java.util.Scanner", and defines a public class "StringExcecution" with a main method. The main method creates a Scanner, prompts for input, reverses the string, and prints the reversed string and its length. The code is as follows:

```
1 package com.knoldus;
2 import java.util.Scanner;
3 public class StringExcecution
4 {
5     public static void main( String[] args )
6     {
7         StringOperations performOperations = new StringOperations();
8         Scanner takeInput = new Scanner(System.in);
9         System.out.println("Enter a string you want to perform operations on: ");
10        String stringToOperate = takeInput.nextLine();
11        String reversedSrting = performOperations.stringReverse(stringToOperate);
12        int lengthOfString = performOperations.getStringLength(stringToOperate);
13        System.out.println("The reversed string is: "+reversedSrting);
14        System.out.println("The length of string is: "+lengthOfString);
15    }
16 }
```

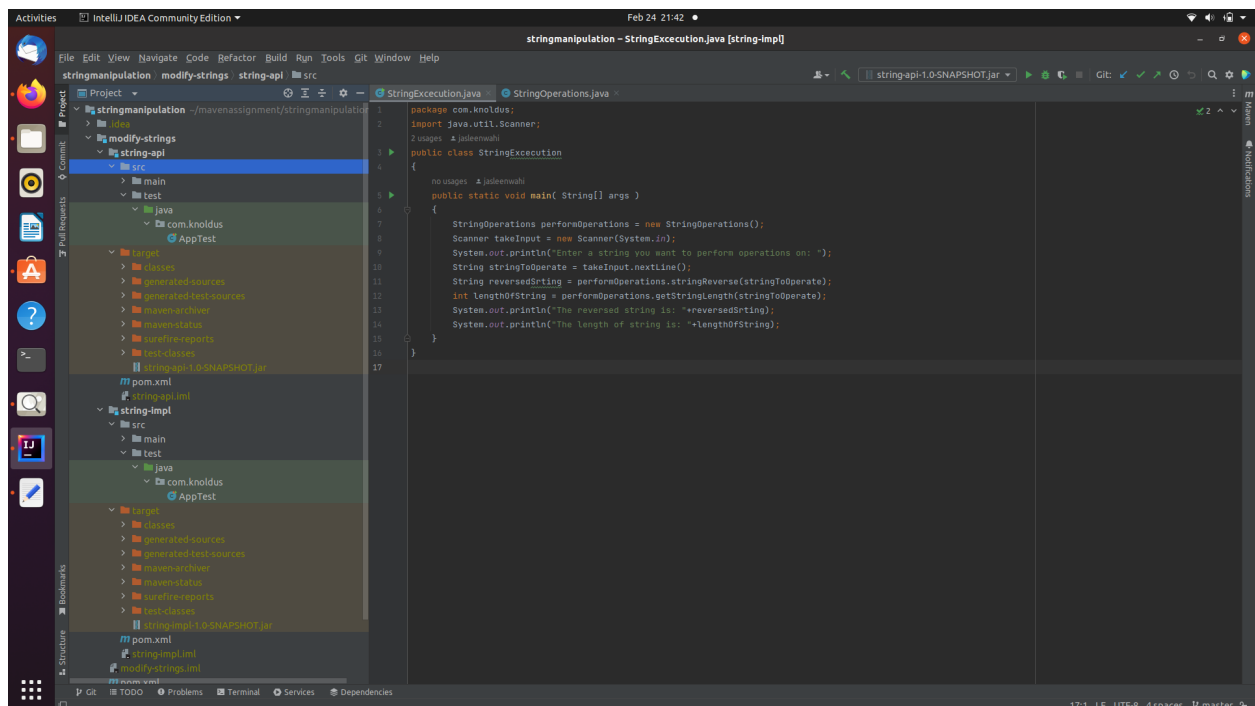
The editor has a sidebar on the left with various icons and a status bar at the bottom showing "Java", "Tab Width: 8", "Ln 6, Col 6", and "OVR".

**Step 12:** Open project in intelij and mark java folders inside string-api and string-impl src folder as **source root**.





**Step 13: mark java folder inside test folder as test source root.**



**Step 14:** Finally execute the file containing main method to see the output.

