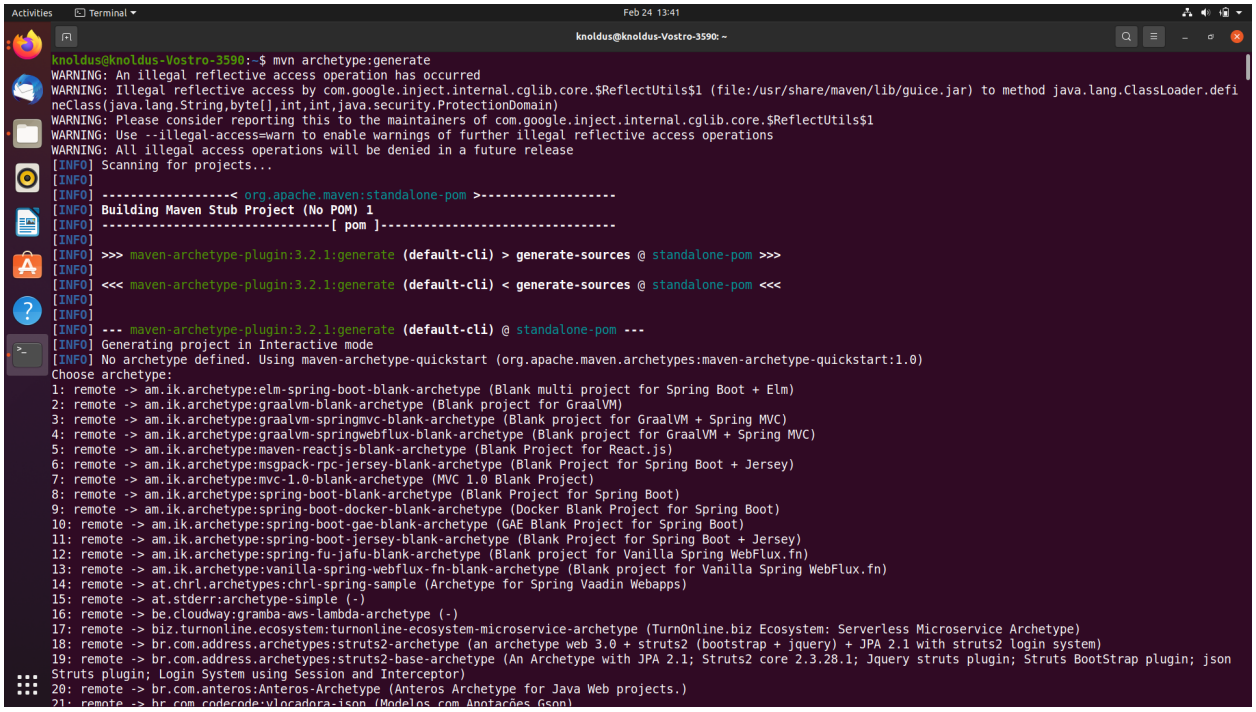
Maven Assignment

Step 1: Create the parent directory by using command:

mvn archetype: generate



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

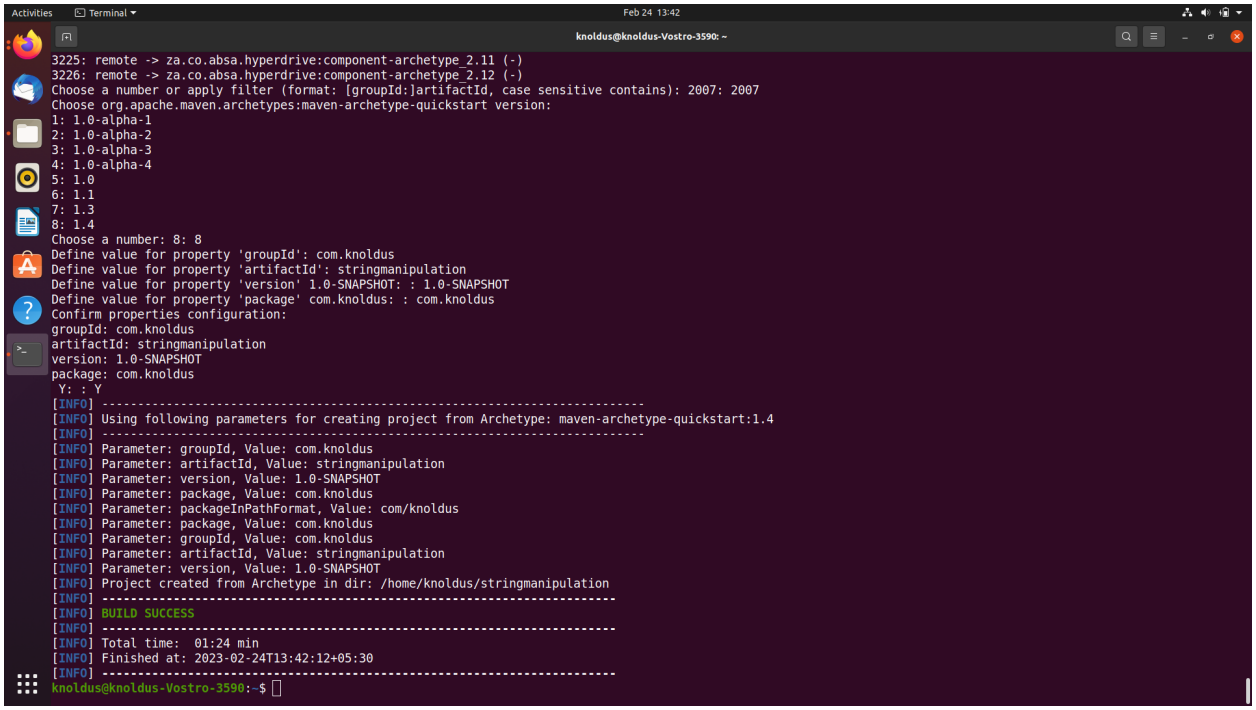
select archetype - choose a template for the project

choose pom version - any version of pom can be selected

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

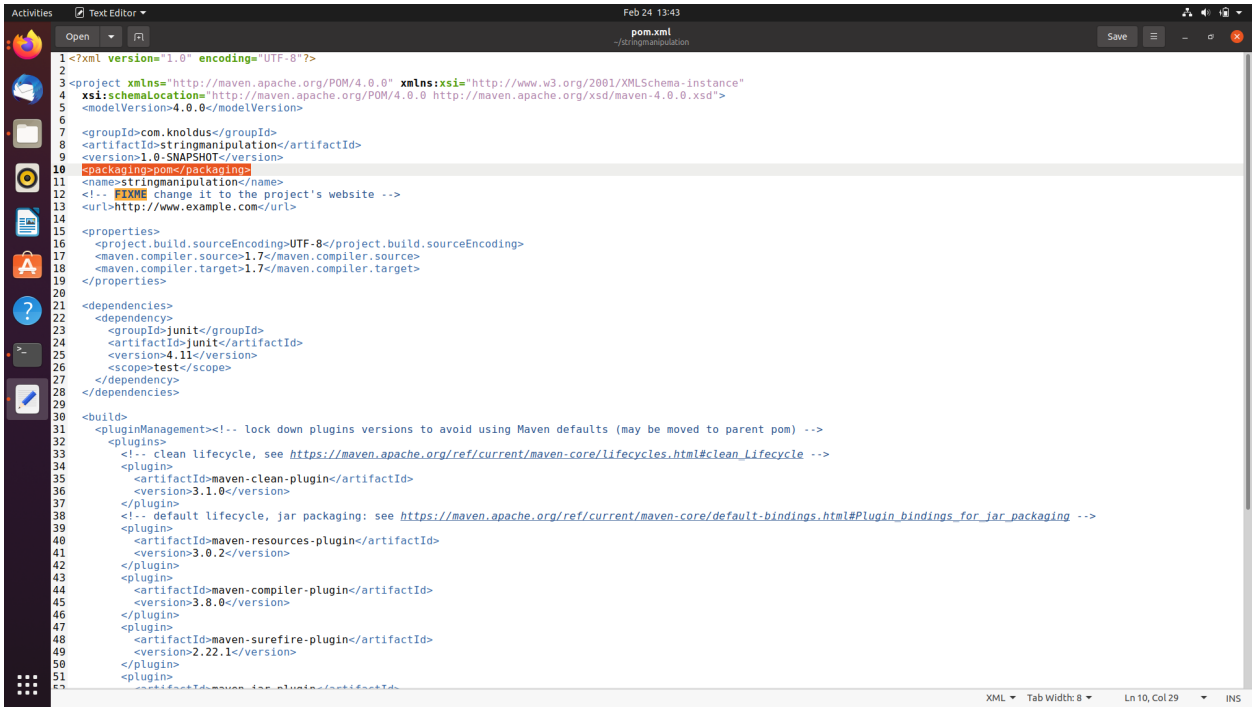
choose archetype - provide a name for your project

choose package - a name of package has to be provided



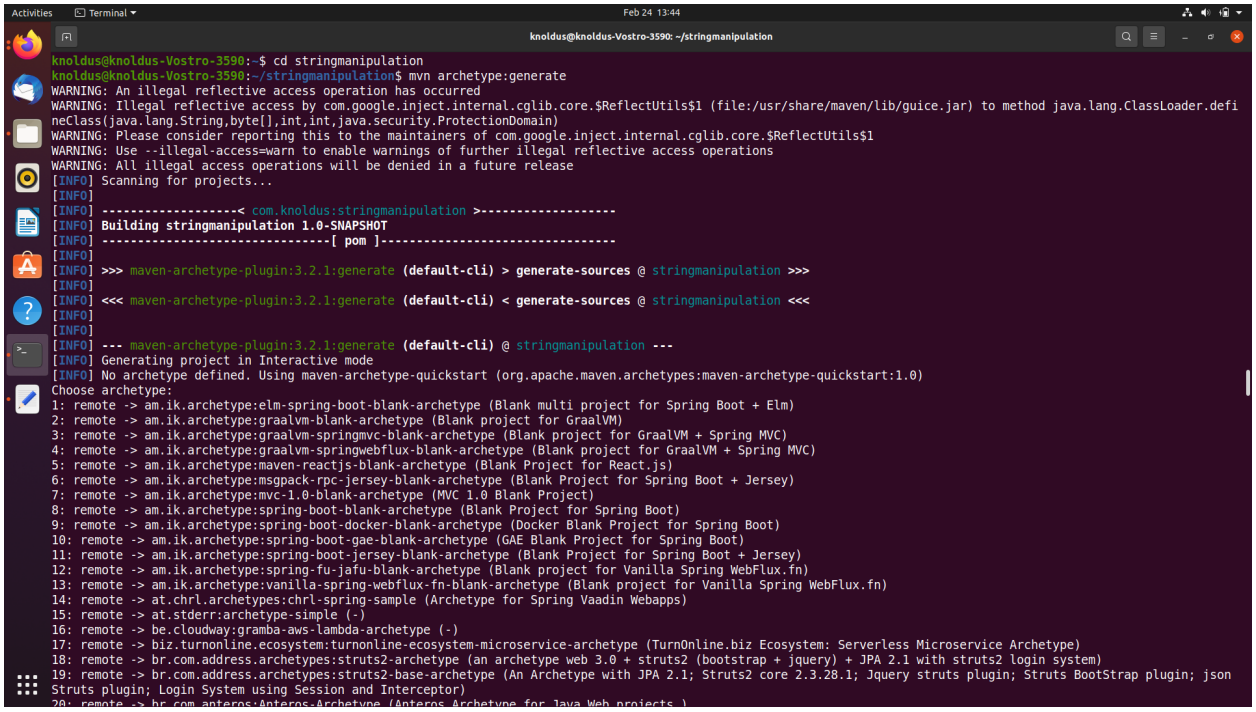
Step 3: Edit the pom.xml file to add packaging to **pom**

pom indicates that it is the parent directory



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

mvn archetype: generate



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

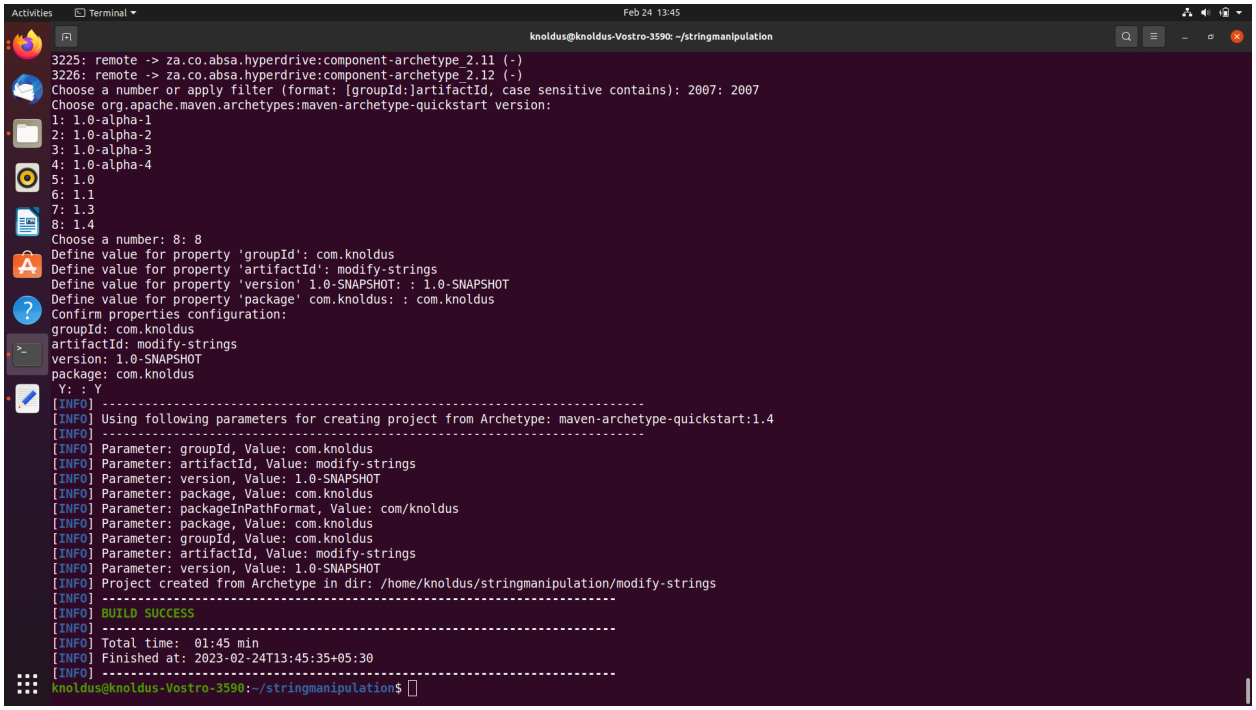
select archetype - choose a template for the project

choose pom version - any version of pom can be selected

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

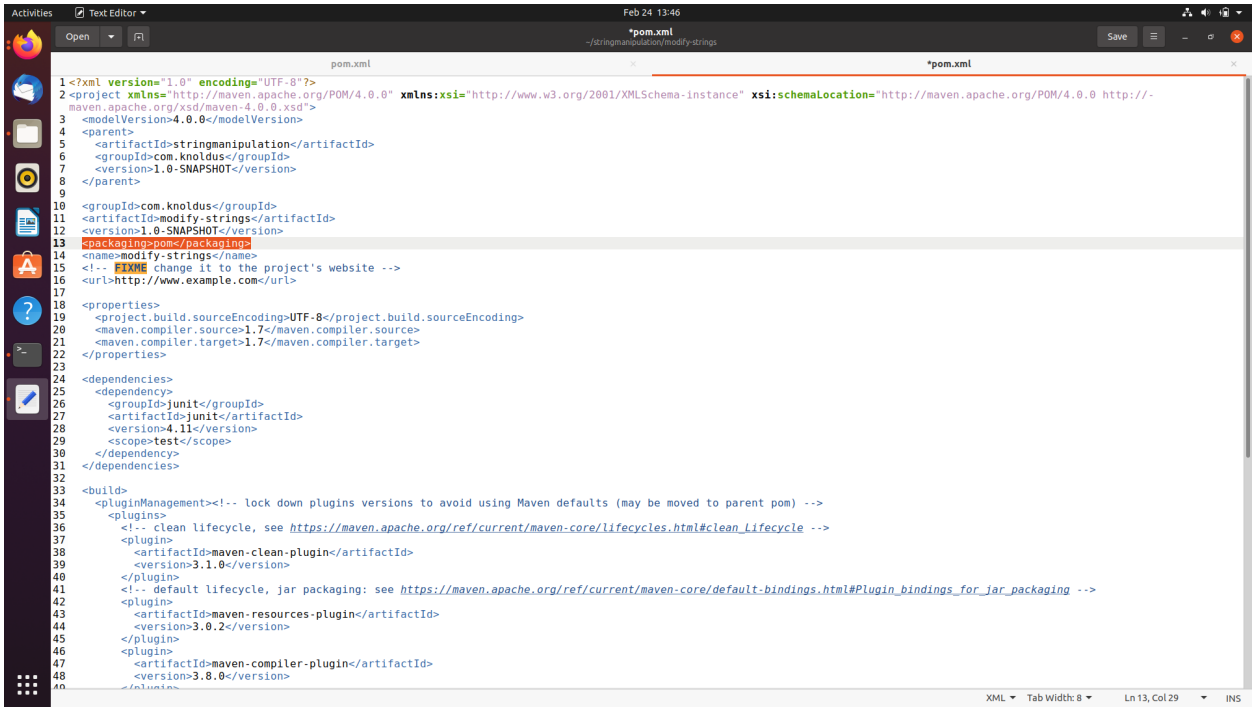
choose archetype - provide a name for your project

choose package - a name of package has to be provided



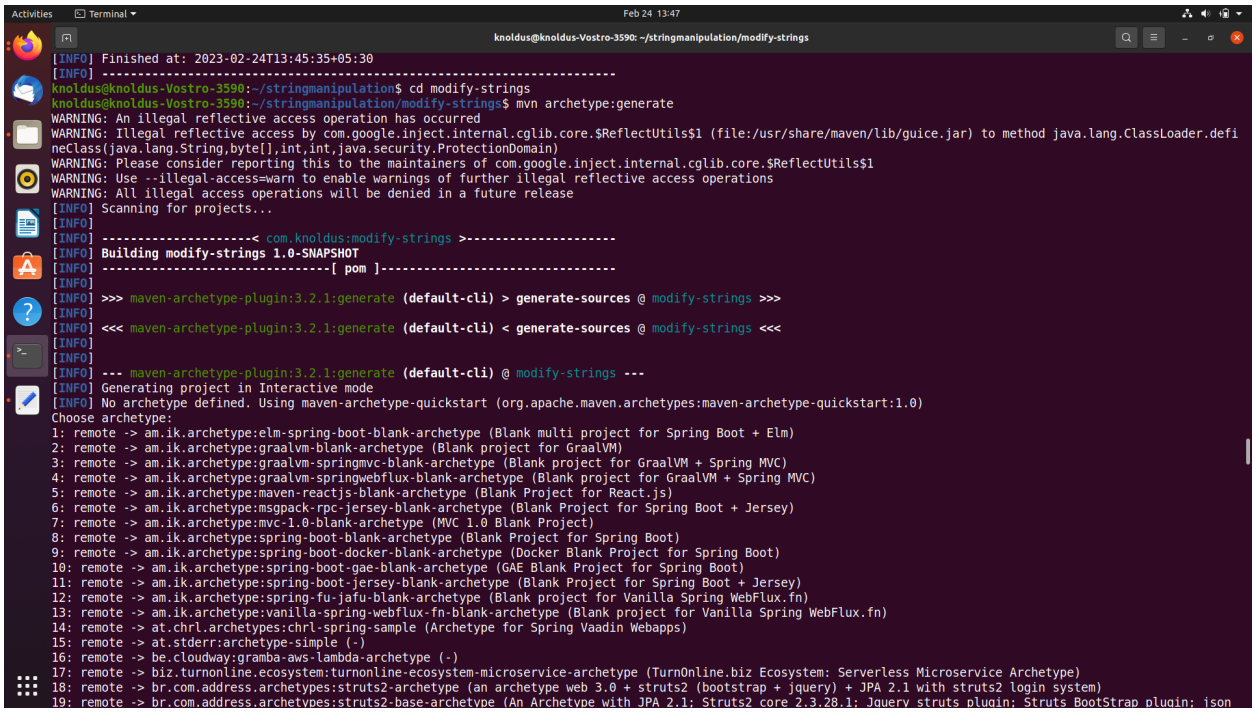
Step 6: Edit the pom.xml file to add packaging to **pom** as we will create more modules into this.

pom indicates that it is the parent directory



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



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

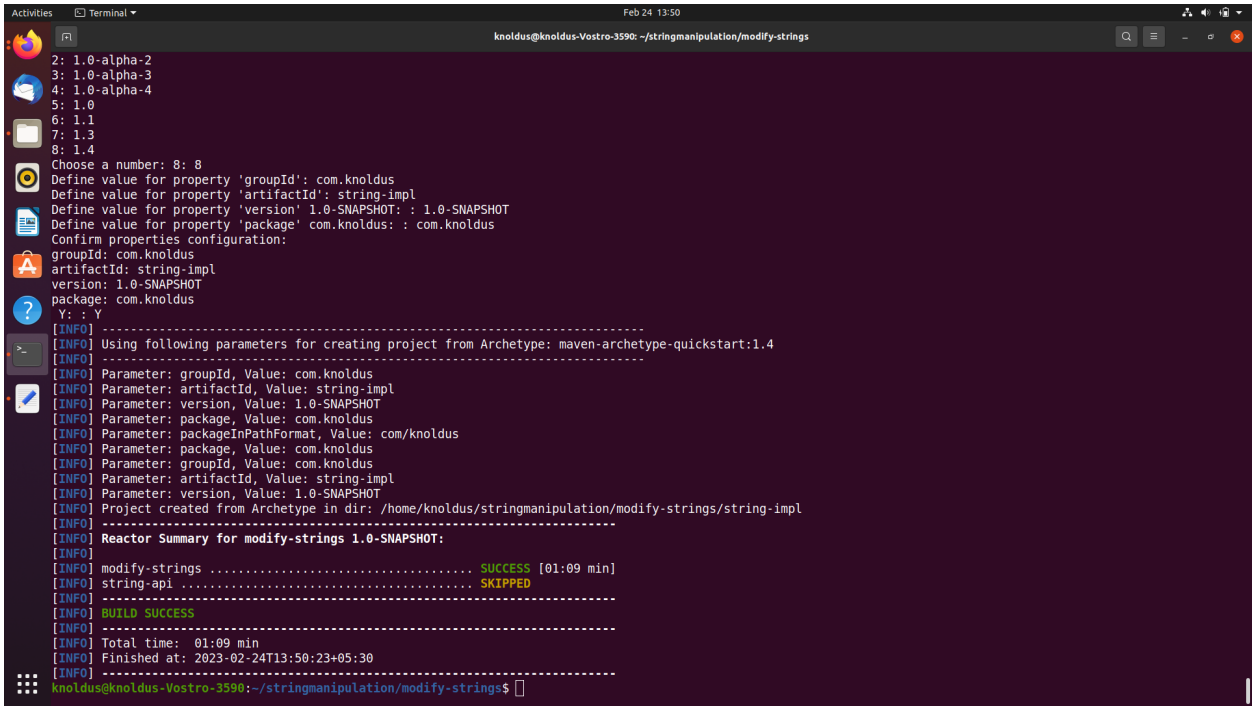
select archetype - choose a template for the project

choose pom version - any version of pom can be selected

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

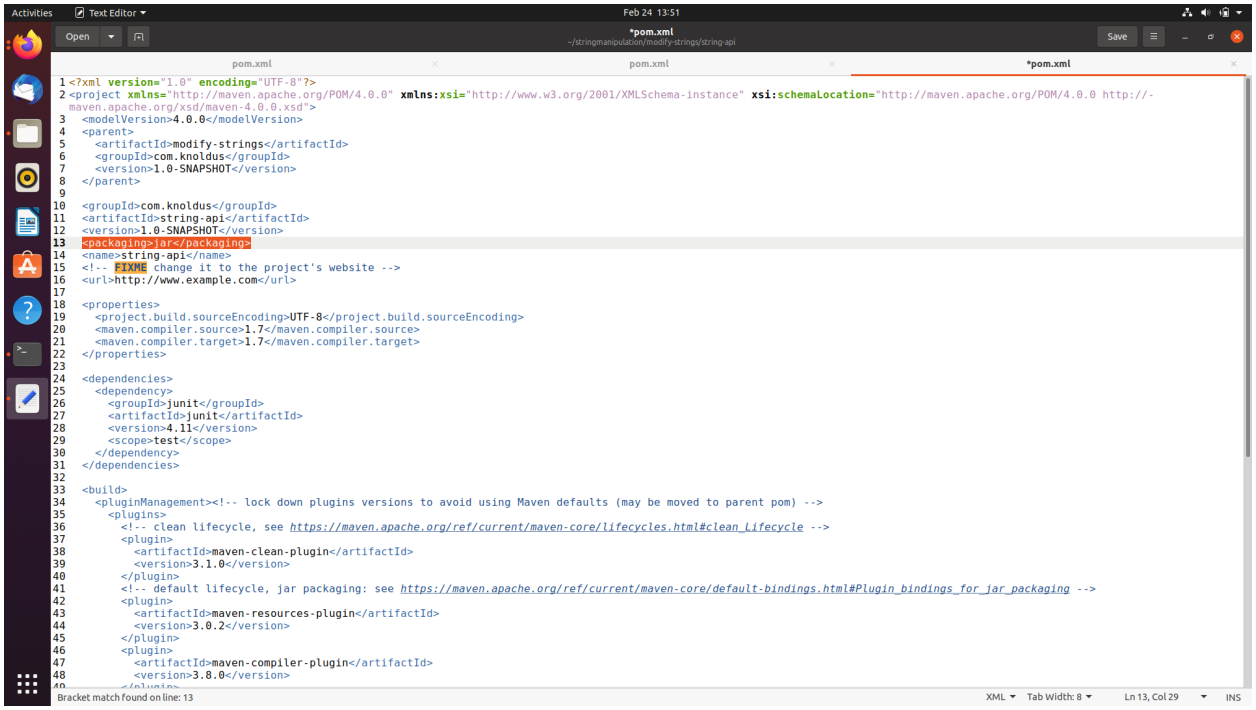
choose archetype - provide a name for your project

choose package - a name of package has to be provided

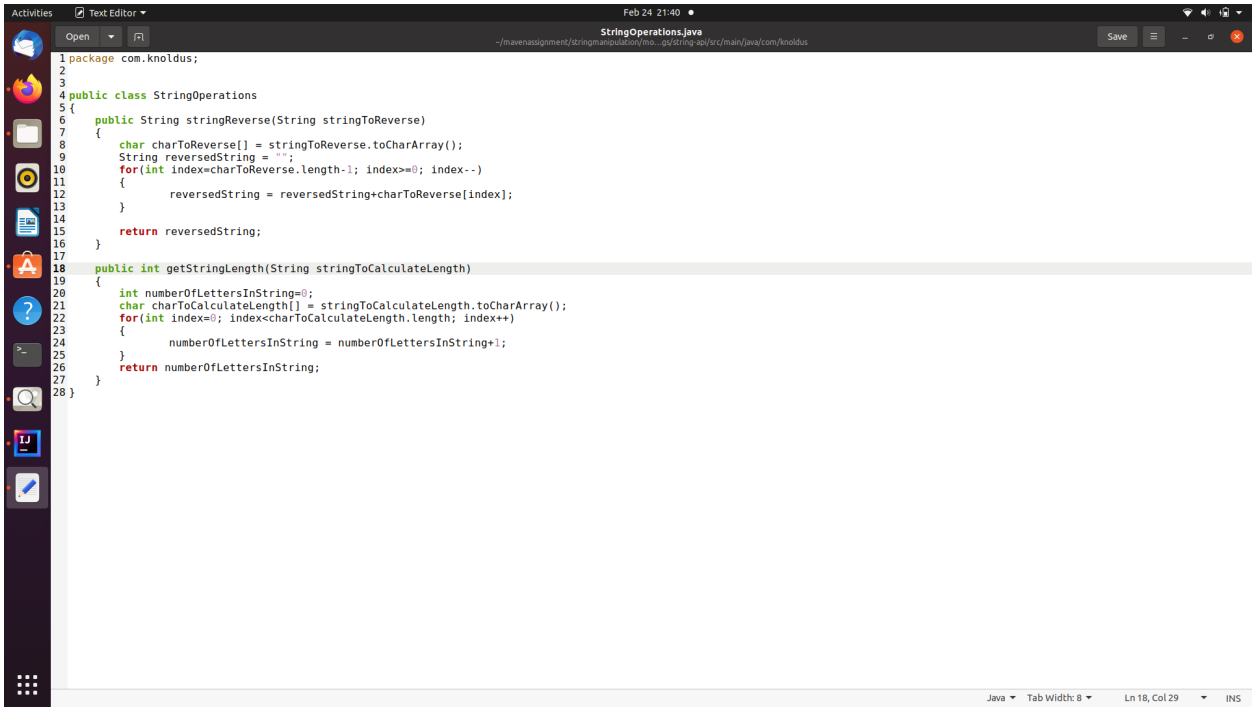


Step 9: Edit the pom.xml files of these modules to add packaging to jar.

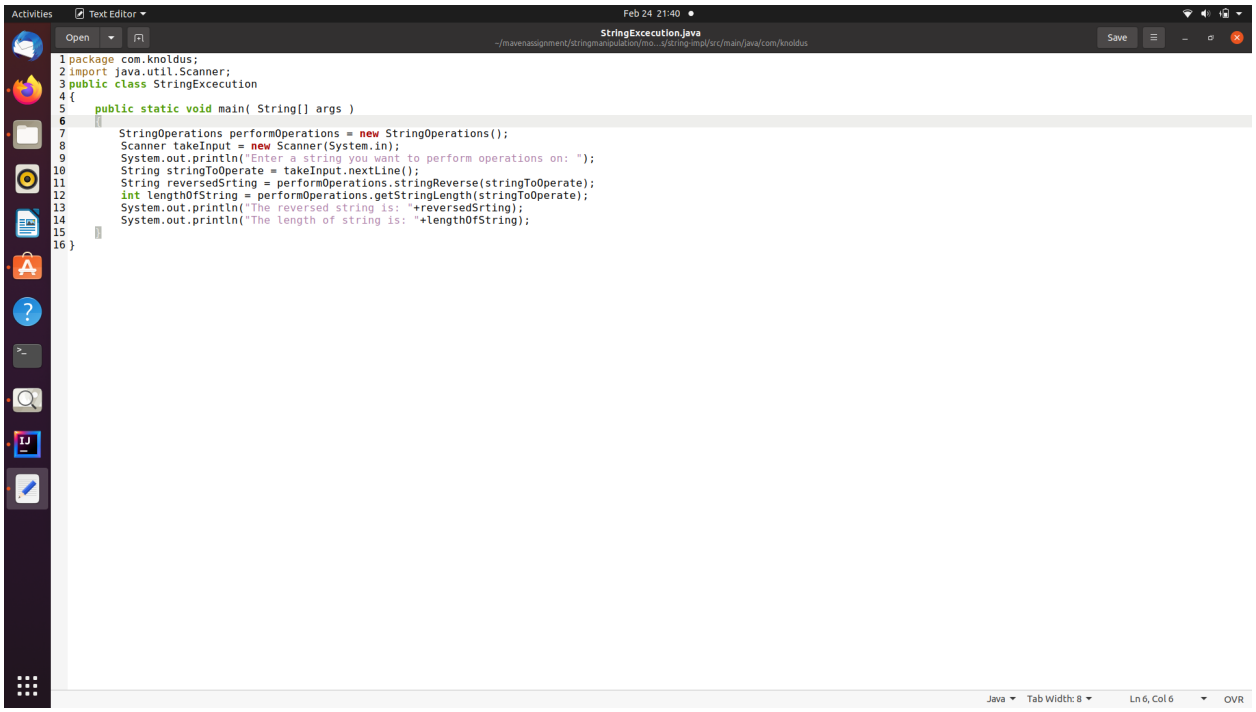
jar - Projects with this packaging type produce a compressed zip file with the *.jar* extension. It may include pure Java classes, interfaces, resources, and metadata files.



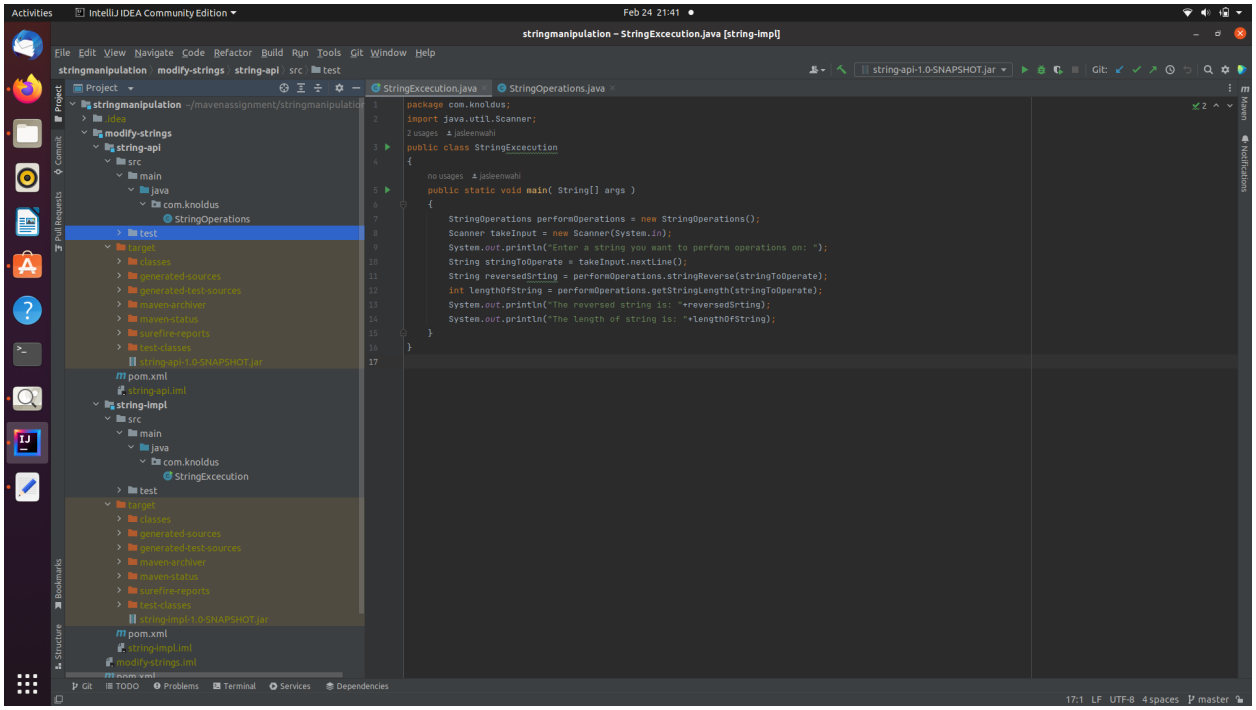
Step 10: Now add a java file into **string-api>src>main>java>com>Knoldus** containing code of the methods.



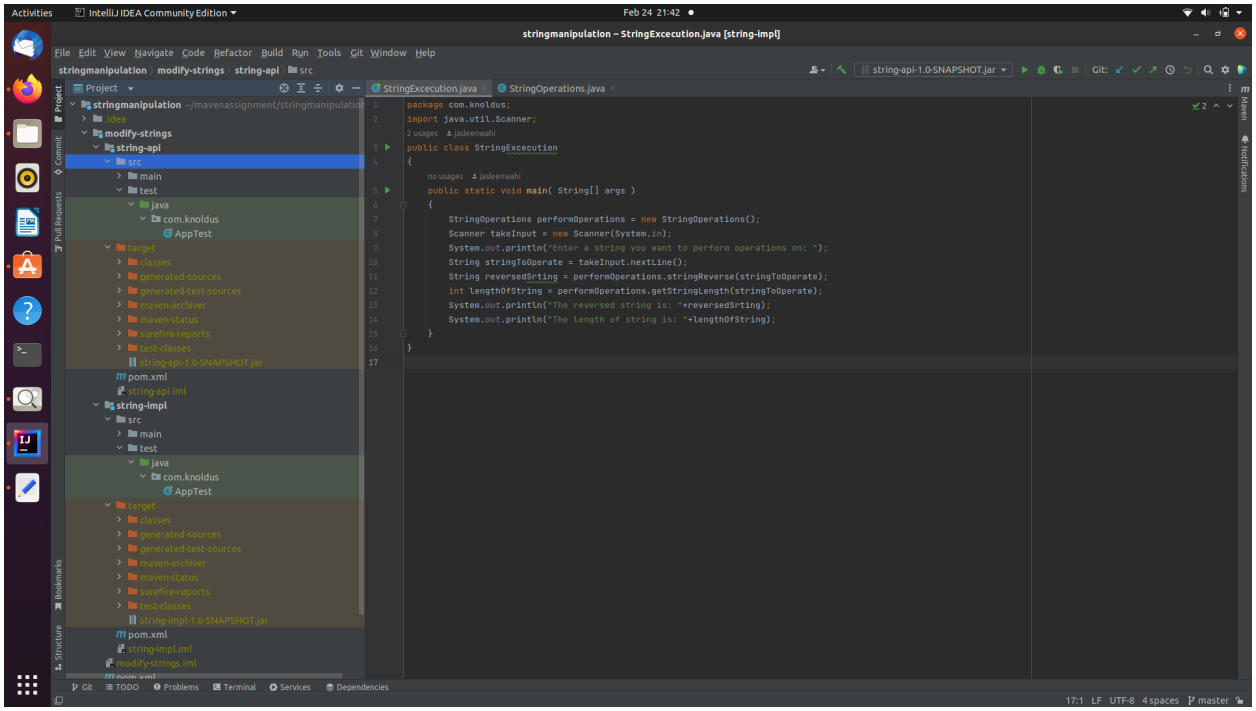
Step 11: Now add a java file into **string-impl>src>main>java >com>Knoldus** containing code of the main method.



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.