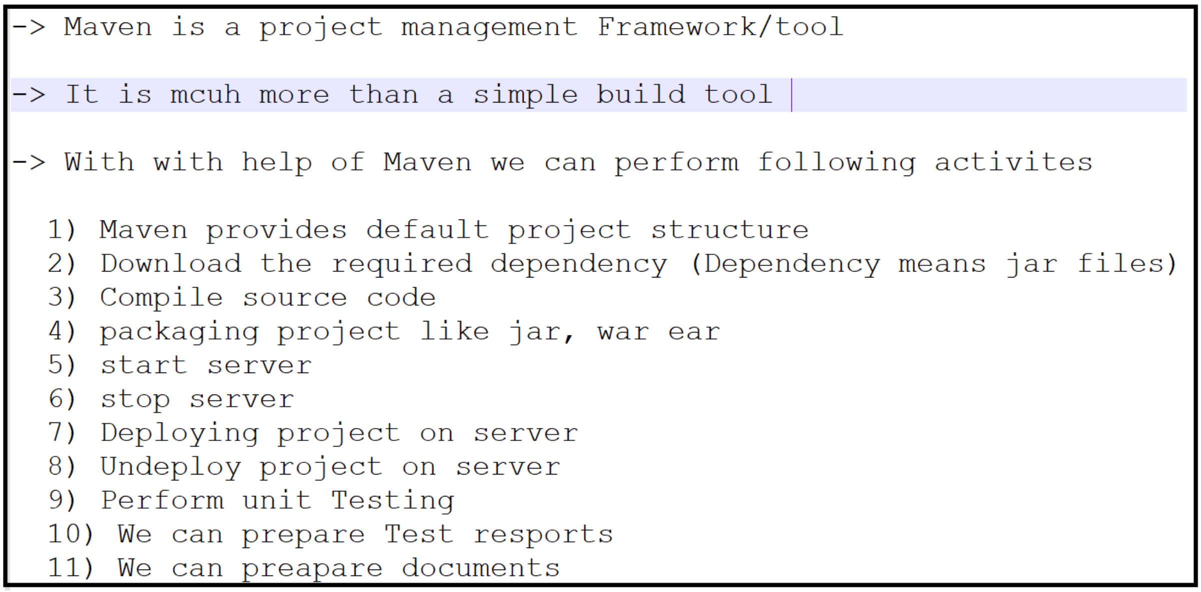
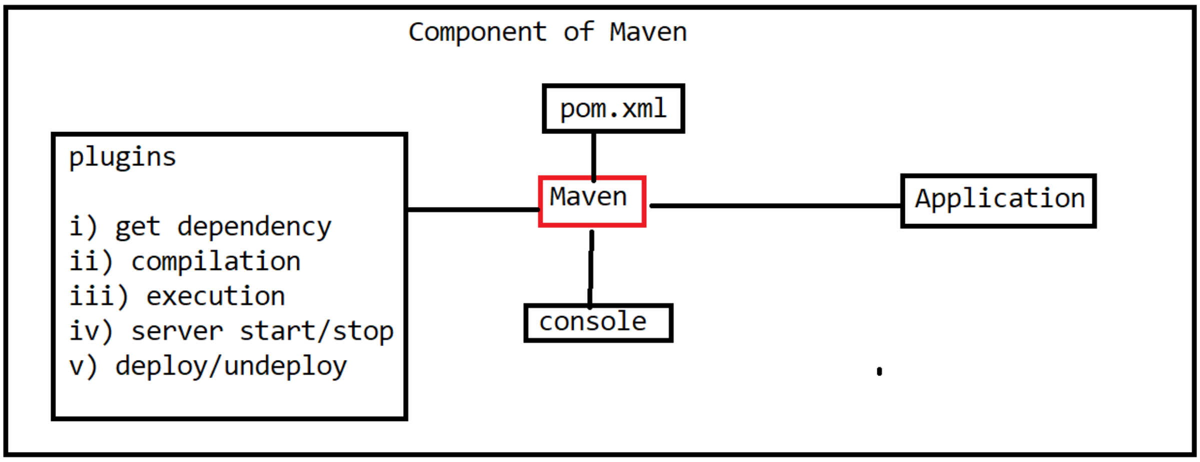


-> Maven is a project management Framework/tool

-> It is mcuh more than a simple build tool

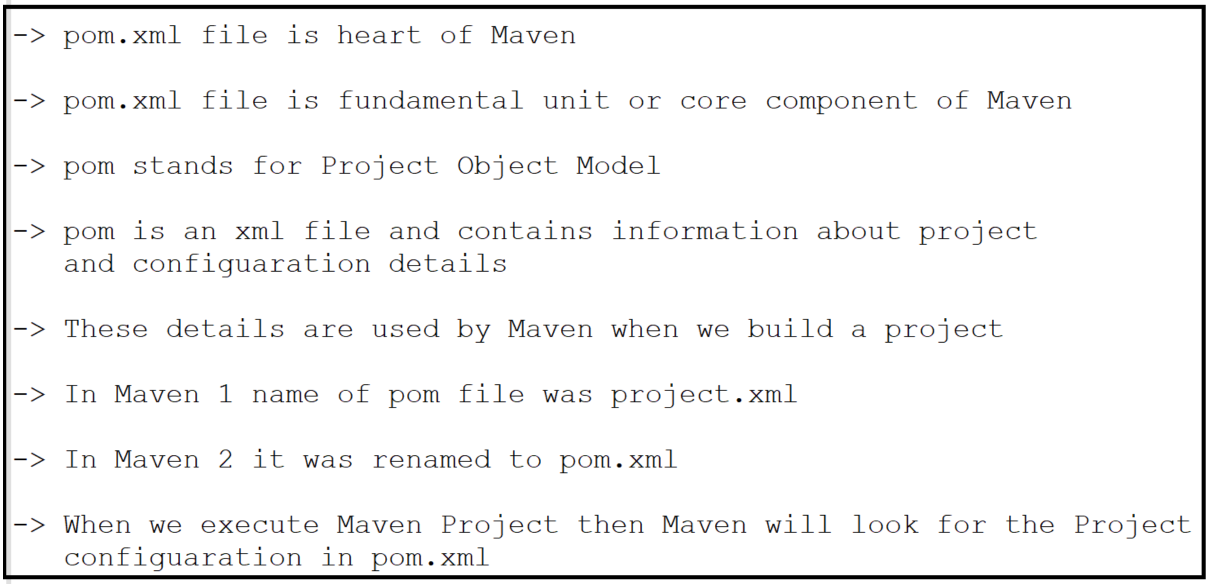
-> With with help of Maven we can perform following activites

1) Maven provides default project structure
2) Download the required dependency (Dependency means jar files)
3) Compile source code
4) packaging project like jar, war ear
5) start server
6) stop server
7) Deploying project on server
8) Undeploy project on server
9) Perform unit Testing
10) We can prepare Test resports
11) We can preapare documents



Component of Maven

pom.xml file



-> pom.xml file is heart of Maven

-> pom.xml file is fundamental unit or core component of Maven

-> pom stands for Project Object Model

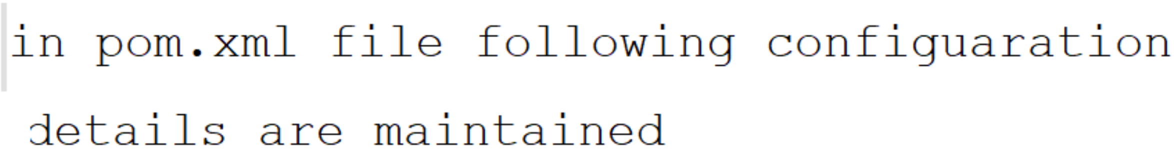
-> pom is an xml file and contains information about project and configuaration details

-> These details are used by Maven when we build a project

-> In Maven 1 name of pom file was project.xml

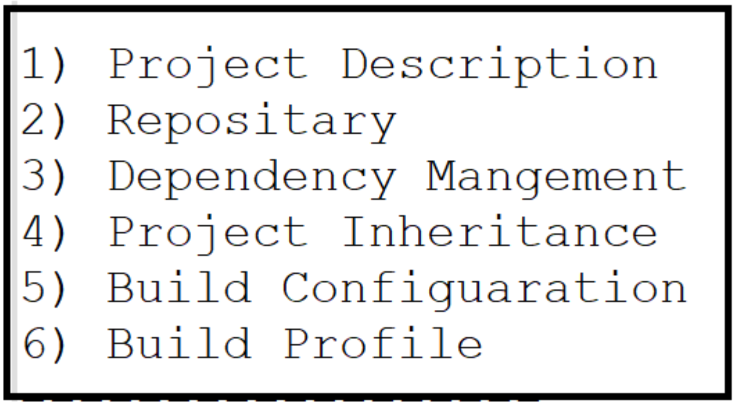
-> In Maven 2 it was renamed to pom.xml

-> When we execute Maven Project then Maven will look for the Project configuaration in pom.xml

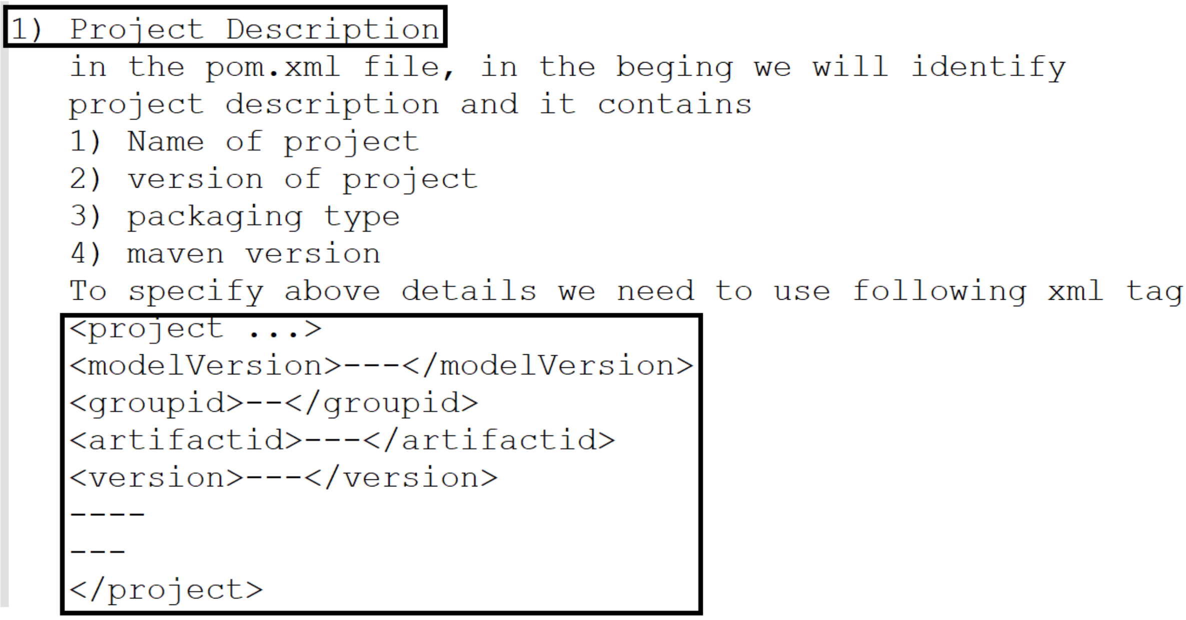


in pom.xml file following configuaration

details are maintained



1) Project Description
2) Repositary
3) Dependency Mangement
4) Project Inheritance
5) Build Configuaration
6) Build Profile



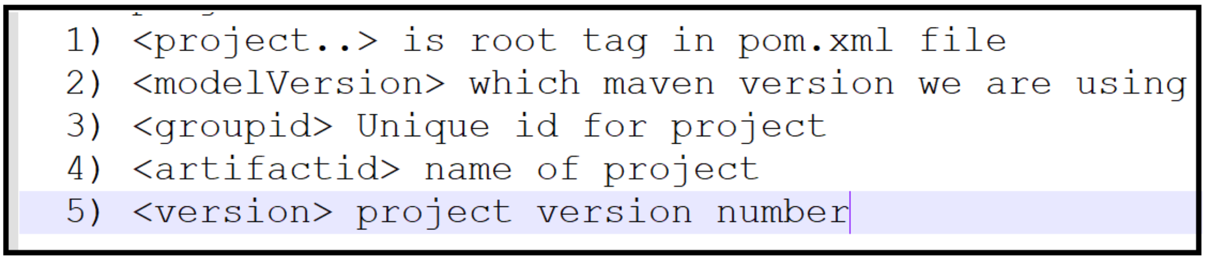
1) Project Description

in the pom.xml file, in the beging we will identify project description and it contains

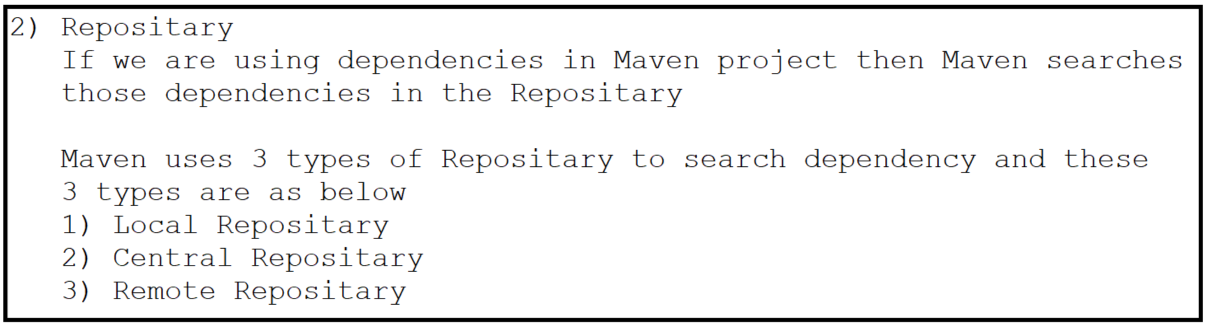
1) Name of project
2) version of project
3) packaging type
4) maven version

To specify above details we need to use following xml tag

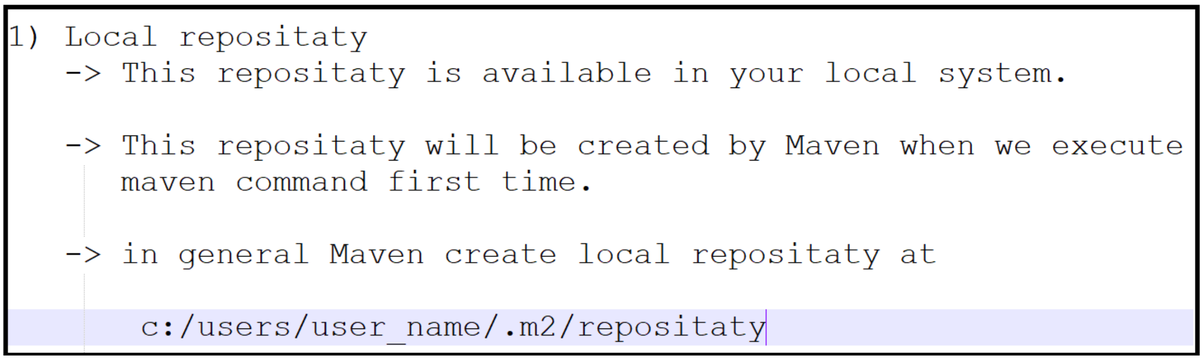
```
<project ...>
<modelVersion>---</modelVersion>
<groupid>--</groupid>
<artifactid>---</artifactid>
<version>---</version>
----
---
</project>
```



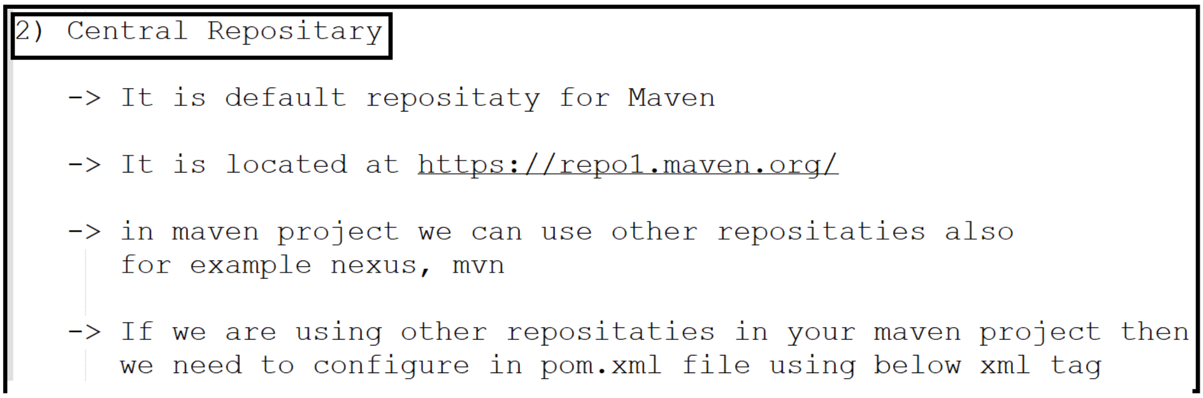
1) <project..> is root tag in pom.xml file
2) <modelVersion> which maven version we are using
3) <groupid> Unique id for project
4) <artifactid> name of project
5) <version> project version number



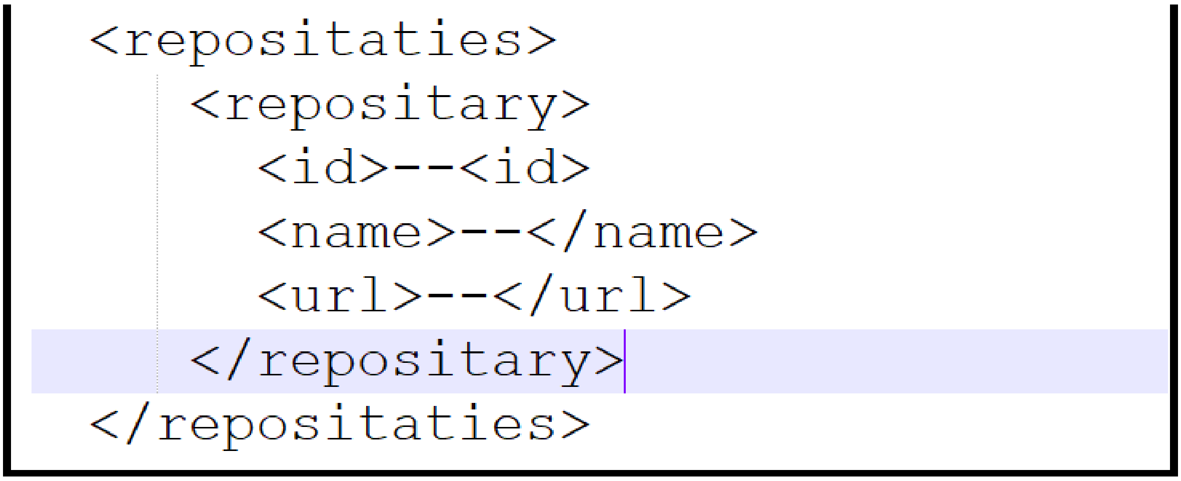
```
2) Repositary
   If we are using dependencies in Maven project then Maven searches
   those dependencies in the Repositary

   Maven uses 3 types of Repositary to search dependency and these
   3 types are as below
   1) Local Repositary
   2) Central Repositary
   3) Remote Repositary
```



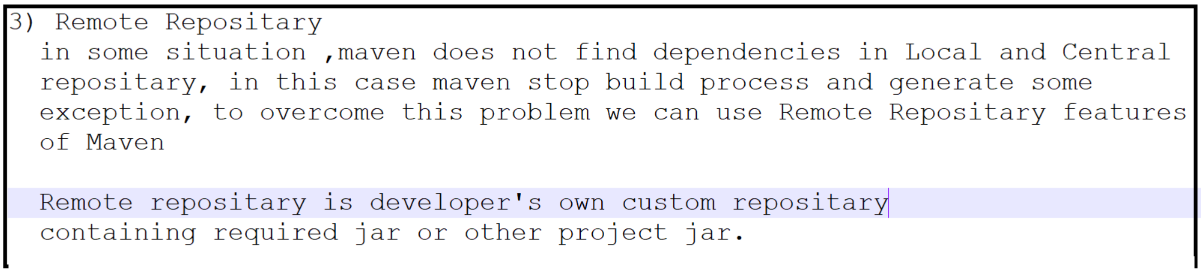
```
1) Local repositaty
   -> This repositaty is available in your local system.

   -> This repositaty will be created by Maven when we execute
      maven command first time.

   -> in general Maven create local repositaty at

       c:/users/user_name/.m2/repositaty
```



```
2) Central Repositary

   -> It is default repositaty for Maven

   -> It is located at https://repo1.maven.org/

   -> in maven project we can use other repositaties also
      for example nexus, mvn

   -> If we are using other repositaties in your maven project then
      we need to configure in pom.xml file using below xml tag
```

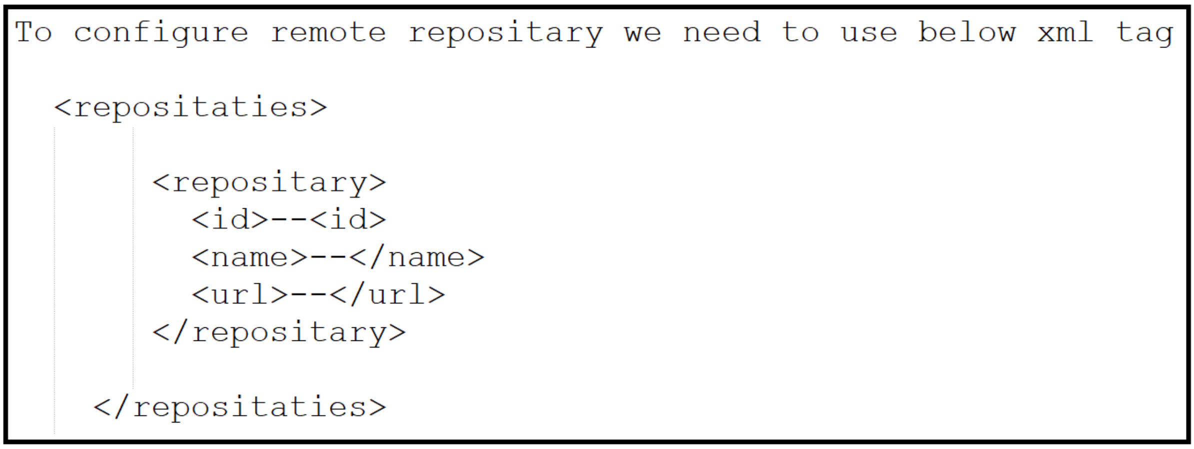


```
<repositaties>
   <repositary>
     <id>--<id>
     <name>--</name>
     <url>--</url>
   </repositary>
</repositaties>
```



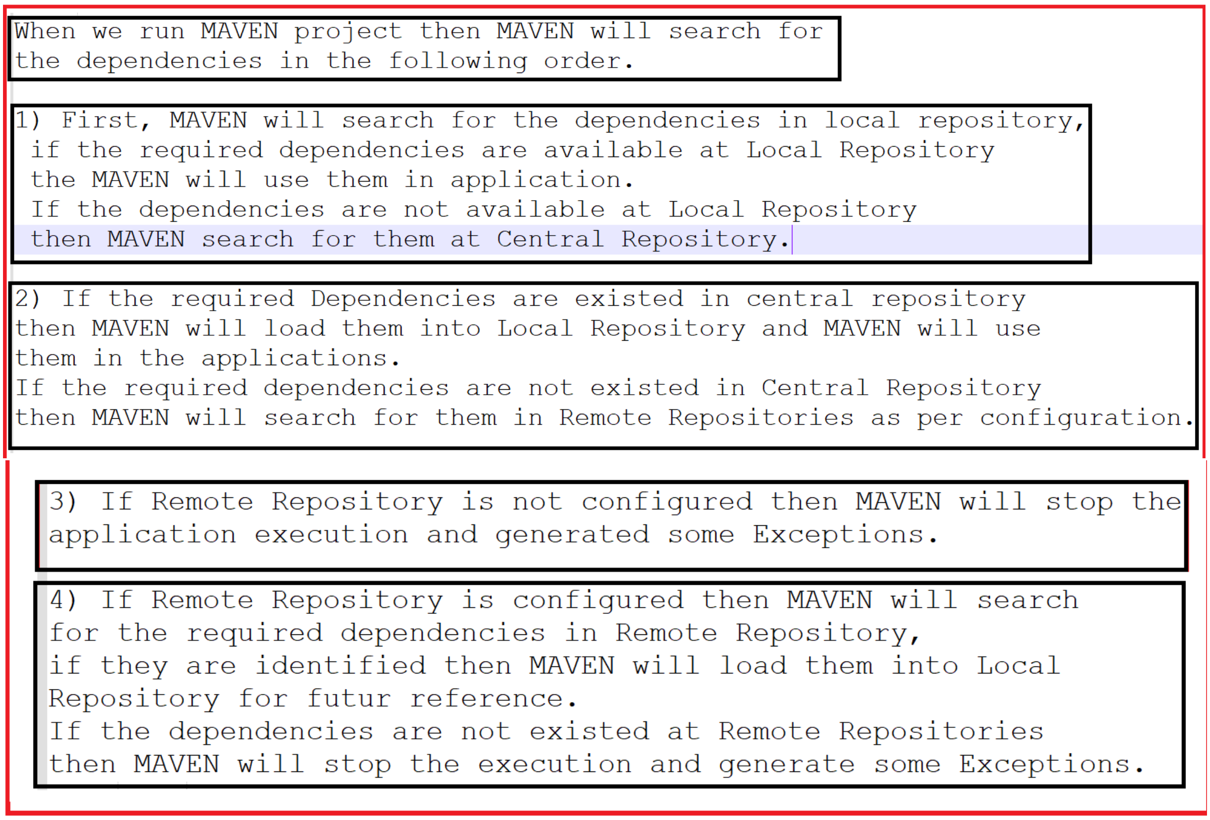
3) Remote Repositary
in some situation ,maven does not find dependencies in Local and Central repositary, in this case maven stop build process and generate some exception, to overcome this problem we can use Remote Repositary features of Maven

Remote repositary is developer's own custom repositary containing required jar or other project jar.



To configure remote repositary we need to use below xml tag

```
<repositaties>

    <repositary>
      <id>--<id>
      <name>--</name>
      <url>--</url>
    </repositary>

  </repositaties>
```



When we run MAVEN project then MAVEN will search for the dependencies in the following order.

1) First, MAVEN will search for the dependencies in local repository,
if the required dependencies are available at Local Repository
the MAVEN will use them in application.
If the dependencies are not available at Local Repository
then MAVEN search for them at Central Repository.

2) If the required Dependencies are existed in central repository
then MAVEN will load them into Local Repository and MAVEN will use
them in the applications.
If the required dependencies are not existed in Central Repository
then MAVEN will search for them in Remote Repositories as per configuration.

3) If Remote Repository is not configured then MAVEN will stop the application execution and generated some Exceptions.

4) If Remote Repository is configured then MAVEN will search
for the required dependencies in Remote Repository,
if they are identified then MAVEN will load them into Local
Repository for futur reference.
If the dependencies are not existed at Remote Repositories
then MAVEN will stop the execution and generate some Exceptions.