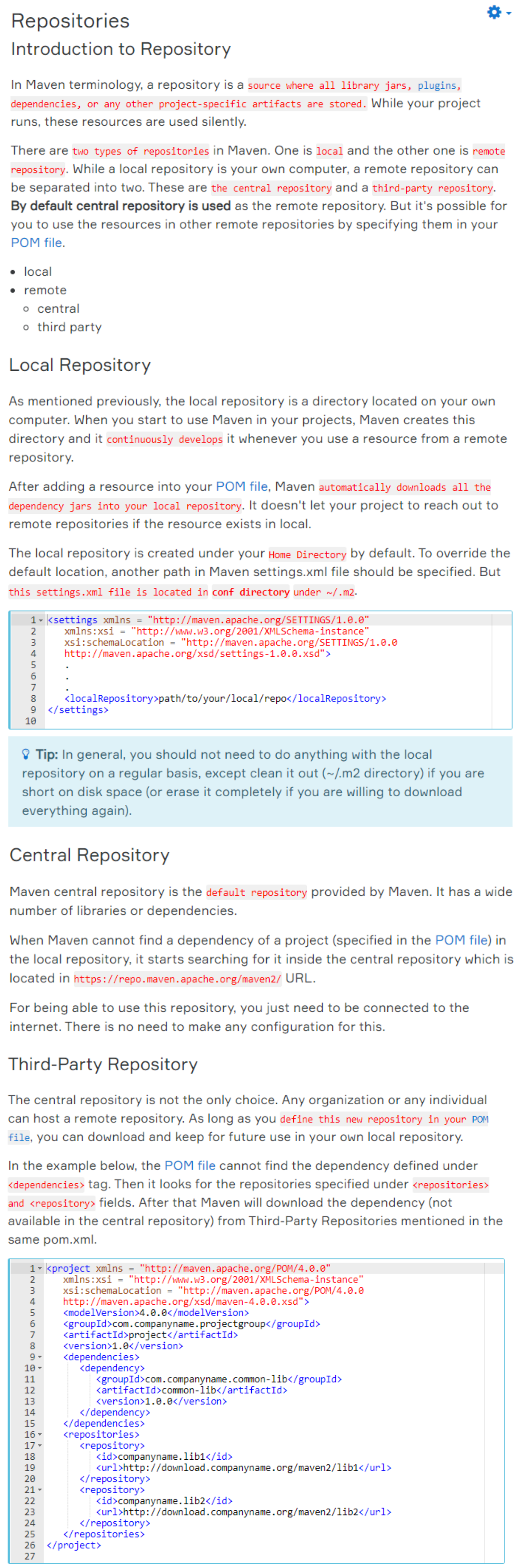


Repositories

Introduction to Repository

In Maven terminology, a repository is a `source where all library jars, plugins, dependencies, or any other project-specific artifacts are stored`. While your project runs, these resources are used silently.

There are `two types of repositories` in Maven. One is `local` and the other one is `remote repository`. While a local repository is your own computer, a remote repository can be separated into two. These are `the central repository` and a `third-party repository`. **By default central repository is used** as the remote repository. But it's possible for you to use the resources in other remote repositories by specifying them in your POM file.

- local
- remote
  - central
  - third party

Local Repository

As mentioned previously, the local repository is a directory located on your own computer. When you start to use Maven in your projects, Maven creates this directory and it `continuously develops` it whenever you use a resource from a remote repository.

After adding a resource into your POM file, Maven `automatically downloads all the dependency jars into your local repository`. It doesn't let your project to reach out to remote repositories if the resource exists in local.

The local repository is created under your `Home Directory` by default. To override the default location, another path in Maven settings.xml file should be specified. But `this settings.xml file is located in` **`conf directory`** `under ~/.m2`.

```
<settings xmlns = "http://maven.apache.org/SETTINGS/1.0.0"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://maven.apache.org/SETTINGS/1.0.0
   http://maven.apache.org/xsd/settings-1.0.0.xsd">
   .
   .
   .
   <localRepository>path/to/your/local/repo</localRepository>
</settings>

```

Tip: In general, you should not need to do anything with the local repository on a regular basis, except clean it out (~/.m2 directory) if you are short on disk space (or erase it completely if you are willing to download everything again).

Central Repository

Maven central repository is the `default repository` provided by Maven. It has a wide number of libraries or dependencies.

When Maven cannot find a dependency of a project (specified in the POM file) in the local repository, it starts searching for it inside the central repository which is located in `https://repo.maven.apache.org/maven2/` URL.

For being able to use this repository, you just need to be connected to the internet. There is no need to make any configuration for this.

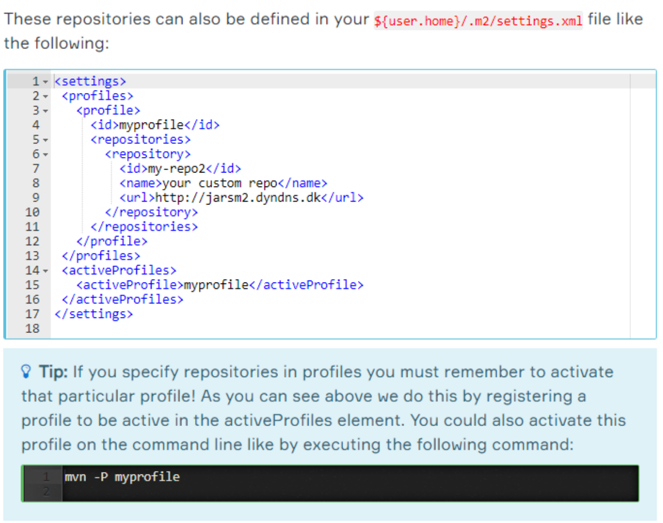
Third-Party Repository

The central repository is not the only choice. Any organization or any individual can host a remote repository. As long as you `define this new repository in your POM file`, you can download and keep for future use in your own local repository.

In the example below, the POM file cannot find the dependency defined under `<dependencies>` tag. Then it looks for the repositories specified under `<repositories> and <repository>` fields. After that Maven will download the dependency (not available in the central repository) from Third-Party Repositories mentioned in the same pom.xml.

```
<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://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   <groupId>com.companyname.projectgroup</groupId>
   <artifactId>project</artifactId>
   <version>1.0</version>
   <dependencies>
      <dependency>
         <groupId>com.companyname.common-lib</groupId>
         <artifactId>common-lib</artifactId>
         <version>1.0.0</version>
      </dependency>
   </dependencies>
   <repositories>
      <repository>
         <id>companyname.lib1</id>
         <url>http://download.companyname.org/maven2/lib1</url>
      </repository>
      <repository>
         <id>companyname.lib2</id>
         <url>http://download.companyname.org/maven2/lib2</url>
      </repository>
   </repositories>
</project>

```



These repositories can also be defined in your `${user.home}/.m2/settings.xml` file like the following:

```
<settings>
 <profiles>
   <profile>
     <id>myprofile</id>
     <repositories>
       <repository>
         <id>my-repo2</id>
         <name>your custom repo</name>
         <url>http://jarsm2.dyndns.dk</url>
       </repository>
     </repositories>
   </profile>
 </profiles>
 <activeProfiles>
   <activeProfile>myprofile</activeProfile>
 </activeProfiles>
</settings>

```

Tip: If you specify repositories in profiles you must remember to activate that particular profile! As you can see above we do this by registering a profile to be active in the activeProfiles element. You could also activate this profile on the command line like by executing the following command:

```
mvn -P myprofile

```