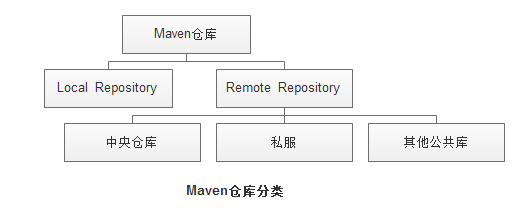
Maven的Mirror和Repository 的详细讲解

## 1 Repository（仓库）



### 1.1 Maven仓库主要有2种：

* remote repository：相当于公共的仓库，大家都能访问到，一般可以用URL的形式访问
* local repository：存放在本地磁盘的一个文件夹，例如，windows上默认是C:\Users\｛用户名｝\.m2\repository目录

### 1.2 Remote Repository主要有3种：

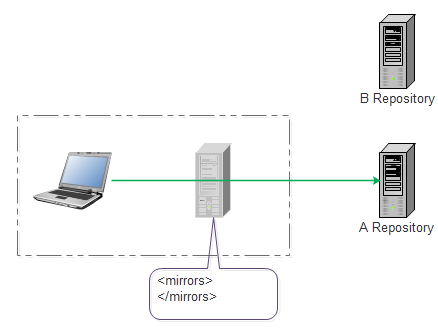
* 中央仓库：http://repo1.maven.org/maven2/
* 私服：内网自建的maven repository，其URL是一个内部网址
* 其他公共仓库：其他可以互联网公共访问maven repository，例如 jboss repository等

repository里存放的都是各种jar包和maven插件。当向仓库请求插件或依赖的时候，会先检查local repository，如果local repository有则直接返回，否则会向remote repository请求，并缓存到local repository。也可以把做的东西放到本地仓库，仅供本地使用；或上传到远程仓库，供大家使用。

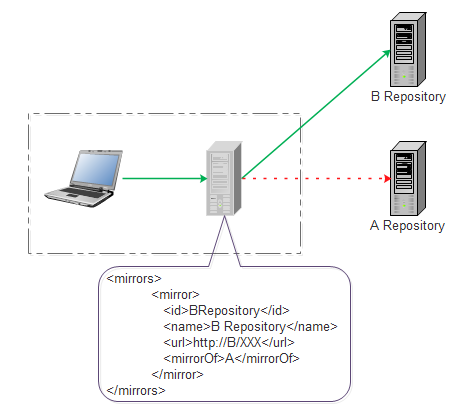
## 2 Mirror

mirror相当于一个拦截器，它会拦截maven对remote repository的相关请求，把请求里的remote repository地址，重定向到mirror里配置的地址。

### 2.1 没有配置mirror：



### 2.2 配置mirror：



此时，B Repository被称为A Repository的镜像。

如果仓库X可以提供仓库Y存储的所有内容，那么就可以认为X是Y的一个镜像。换句话说，任何一个可以从仓库Y获得的构件，都胡够从它的镜像中获取。

### 2.3 <mirrorOf></mirrorOf>

<mirrorOf></mirrorOf>标签里面放置的是要被镜像的Repository ID。为了满足一些复杂的需求，Maven还支持更高级的镜像配置：

* <mirrorOf>\*</mirrorOf>

            匹配所有远程仓库。

* <mirrorOf>repo1,repo2</mirrorOf>

            匹配仓库repo1和repo2，使用逗号分隔多个远程仓库。

* <mirrorOf>\*,!repo1</miiroOf>

            匹配所有远程仓库，repo1除外，使用感叹号将仓库从匹配中排除。

## 3 Repository与Mirror

### 3.1 定义

其实，mirror表示的是两个Repository之间的关系，在maven配置文件（setting.xml)里配置 了<mirrors><mirror>..........</mirror></mirrors>， 即定义了两个Repository之间的镜像关系。

### 3.2 目的

配置两个Repository之间的镜像关系，一般是出于访问速度和下载速度考虑。

例如， 有一个项目，需要在公司和住所都编码，并在项目pom.xml配置了A Maven库。在公司，是电信网络，访问A库很快，所以maven管理依赖和插件都从A库下载；在住所，是网通网络，访问A库很慢，但是访问B库很快。这 时，在住所的setting.xml里，只要配置一 下<mirrors><mirror>....</mirror></mirrors>，让B库成为A 库的mirror，即可不用更改项目pom.xml里对于A库的相关配置。

如果该镜像仓库需要认证，则配置setting.xml中的<server></server>即可。

### 3.3  注意

需要注意的是，由于镜像仓库完全屏蔽了被镜像仓库，当镜像仓库不稳定或者停止服务的时候，Maven仍将无法访问被镜像仓库，因而将无法下载构件。

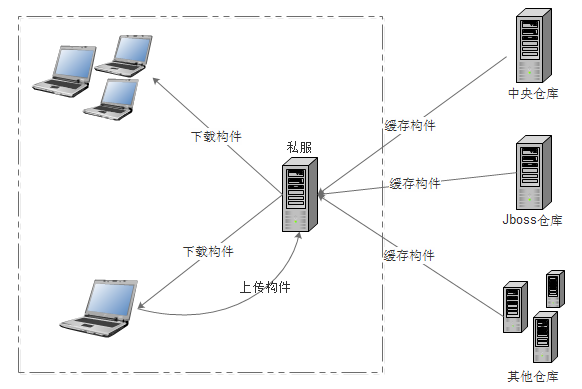
## 4 私服

私服是一种特殊的远程Maven仓库，它是架设在局域网内的仓库服务，私服一般被配置为互联网远程仓库的镜像，供局域网内的Maven用户使用。

当Maven需要下载构件的时候，先向私服请求，如果私服上不存在该构件，则从外部的远程仓库下载，同时缓存在私服之上，然后为Maven下载请求提供下载服务，另外，对于自定义或第三方的jar可以从本地上传到私服，供局域网内其他maven用户使用。

优点主要有：

1. 节省外网宽带
2. 加速Maven构建
3. 部署第三方构件
4. 提高稳定性、增强控制：原因是外网不稳定
5. 降低中央仓库的负荷：原因是中央仓库访问量太大

Maven仓库示意图

