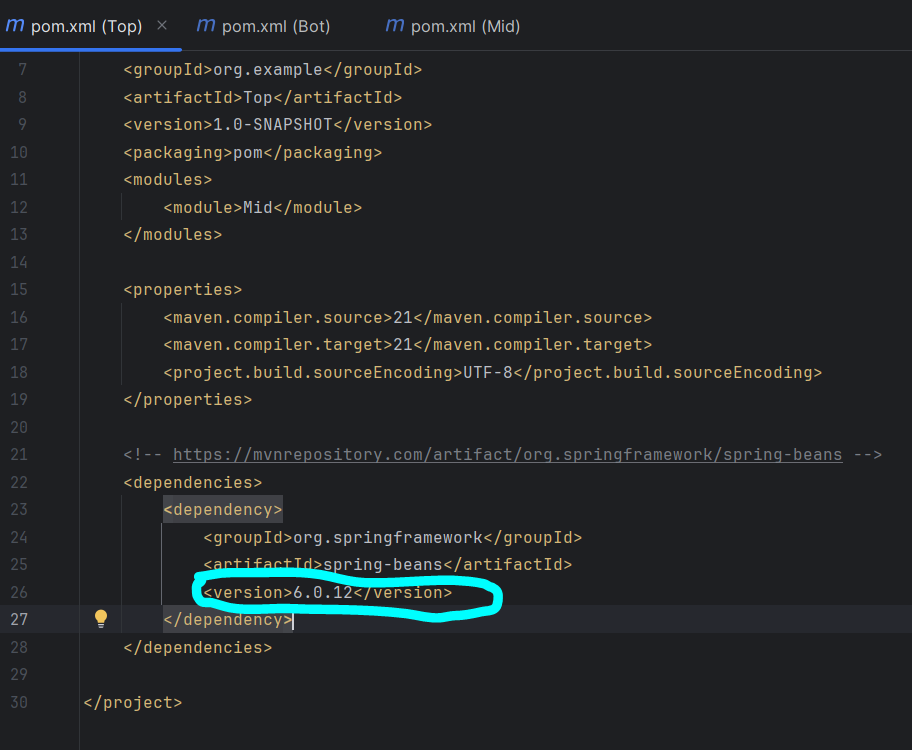


建立完top/mid/bot3级结构（新建top project，然后再new module建立mid/bot，选好父类）

直接依赖：在当前项目中通过依赖配置建立的依赖关系

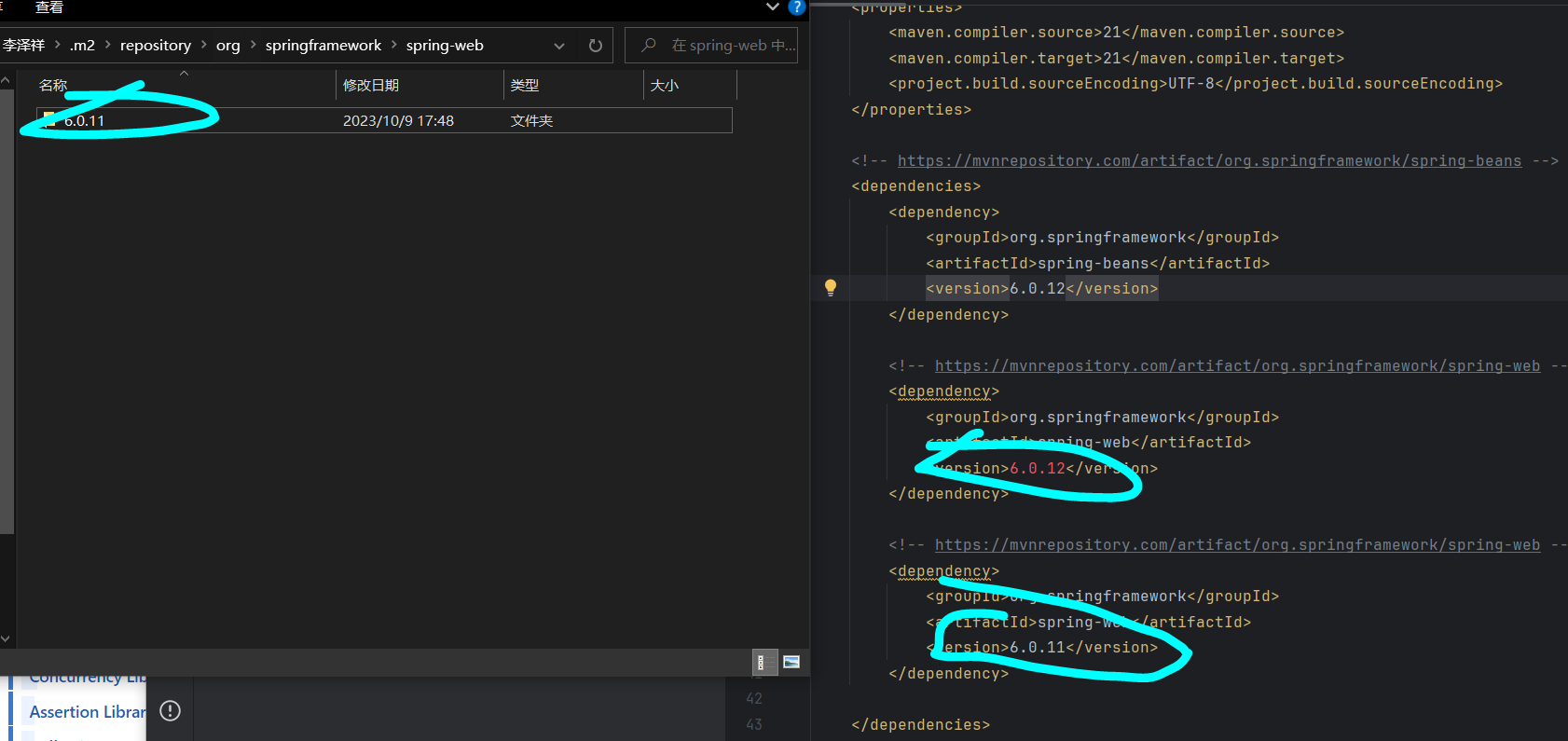
间接依赖：被依赖的资源如果依赖其他资源，当前项目则间接依赖其他资源

在此项目中，可以推断出top直接依赖mid，间接依赖bot。mid直接依赖bot



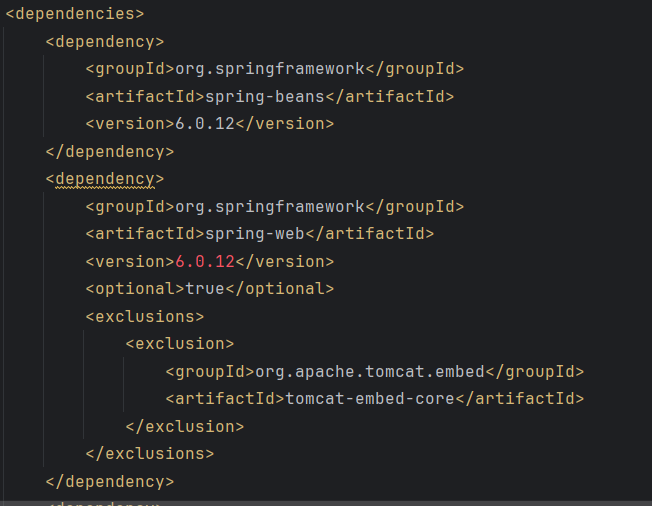
对3个pom分别配置springframework6.0.12、6.0.11、6.0.10版本

此处是路径优先

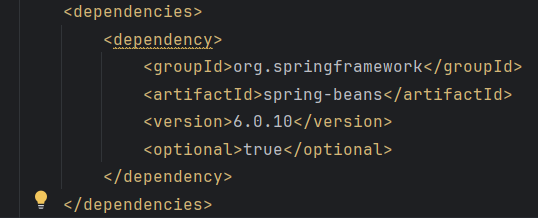


此处是特殊优先，同级后声明的有效

而声明优先只使用top-mid-bot无法展现，在此举个例子来阐释：假如结构变为top-mid1/mid2-bot。在mid1中存在依赖A-1.0.0，而mid2中存在依赖A-1.0.1，top依赖mid1和mid2（并且先声明mid1），那么最后生效的便是mid1中的A-1.0.0



对top的Pom.xml进行修改，将spring-web变为可选依赖并排除其中的tomcat



对mid文件中的spring-beans设为可选依赖



对bot排除"spring-beans"依赖的"spring-context"传递依赖

可选依赖项（optional）只会在需要它们的情况下被引入，否则它们不会包含在构建路径中。排除依赖项（exclusion）将在直接依赖之间移除指定的依赖项。

将"spring-web"依赖项设置为可选依赖项，如果没有其他组件或模块需要使用"spring-web"，它将不会被引入进来。这种设置可以减少项目的依赖项数量，提高构建的效率和可维护性。

在"Bot"项目的POM文件中，排除了"spring-beans"的传递依赖"spring-context"。这意味着在构建项目时，"Bot"项目将不会包含"spring-context"依赖项，即使它是由于"spring-beans"间接引入的。这种设置可以解决依赖项冲突或版本不兼容的问题。