Maven 快照(SNAPSHOT) Maven 依赖管理

自动化构建定义了这样一种场景: 在一个项目成功构建完成后，其相关的依赖工程即开始构建，这样可以保证其依赖项目的稳定。

比如一个团队正在开发一个项目 bus-core-api， 并且有其他两个项目 app-web-ui 和 app-desktop-ui 依赖于这个项目。

app-web-ui 项目使用的是 bus-core-api 项目的 1.0 快照：

```
<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>app-web-ui</groupId>
<artifactId>app-web-ui</artifactId>
<version>1.0</version>
<packaging>jar</packaging>
<dependencies>
<dependency>
<groupId>bus-core-api</groupId>
<artifactId>bus-core-api</artifactId>
<version>1.0-SNAPSHOT</version>
</dependency>
</dependencies>
</project>
```

app-desktop-ui 项目使用的是 bus-core-api 项目的 1.0 快照：

```
<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>app-desktop-ui</groupId>
<artifactId>app-desktop-ui</artifactId>
<version>1.0</version>
<packaging>jar</packaging>
<dependencies>
<dependency>
<groupId>bus-core-api</groupId>
<artifactId>bus-core-api</artifactId>
<version>1.0-SNAPSHOT</version>
</dependency>
</dependencies>
</project>
```

bus-core-api 项目：

```
<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>bus-core-api</groupId>
<artifactId>bus-core-api</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
</project>
```

现在 app-web-ui 和 app-desktop-ui 项目的团队要求不管 bus-core-api 项目何时变化，他们的构建过程都应当可以启动。

使用快照可以确保最新的 bus-core-api 项目被使用，但要达到上面的要求，我们还需要做一些额外的工作。

可以使用两种方式：

- 在 bus-core-api 项目的 pom 文件中添加一个 post-build 目标操作来启动 app-web-ui 和 app-desktop-ui 项目的构建。
- 使用持续集成（CI）服务器，比如 Hudson，来自行管理构建自动化。

使用 Maven

修改 bus-core-api 项目的 pom.xml 文件。

```
<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>bus-core-api</groupId>
<artifactId>bus-core-api</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<build>
<plugins>
<plugin>
<artifactId>maven-invoker-plugin</artifactId>
<version>1.6</version>
<configuration>
<debug>true</debug>
<pomIncludes>
<pomInclude>app-web-ui/pom.xml</pomInclude>
<pomInclude>app-desktop-ui/pom.xml</pomInclude>
</pomIncludes>
</configuration>
<executions>
<execution>
<id>build</id>
<goals>
<goal>run</goal>
</goals>
</execution>
</executions>
</plugin>
</plugins>
<build>
</project>
```

打开命令控制台，切换到 C:\ > MVN > bus-core-api 目录下，然后执行以下命令。

```
C:\MVN\bus-core-api>mvn clean package -U
```

执行完命令后，Maven 将开始构建项目 bus-core-api。

```
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building bus-core-api
[INFO]    task-segment: [clean, package]
[INFO] ------------------------------------------------------------------------
...
[INFO] [jar:jar {execution: default-jar}]
[INFO] Building jar: C:\MVN\bus-core-ui\target\
bus-core-ui-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
```

bus-core-api 构建成功后，Maven 将开始构建 app-web-ui 项目。

```
[INFO] ------------------------------------------------------------------------
[INFO] Building app-web-ui
[INFO]    task-segment: [package]
[INFO] ------------------------------------------------------------------------
...
[INFO] [jar:jar {execution: default-jar}]
[INFO] Building jar: C:\MVN\app-web-ui\target\
app-web-ui-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
```

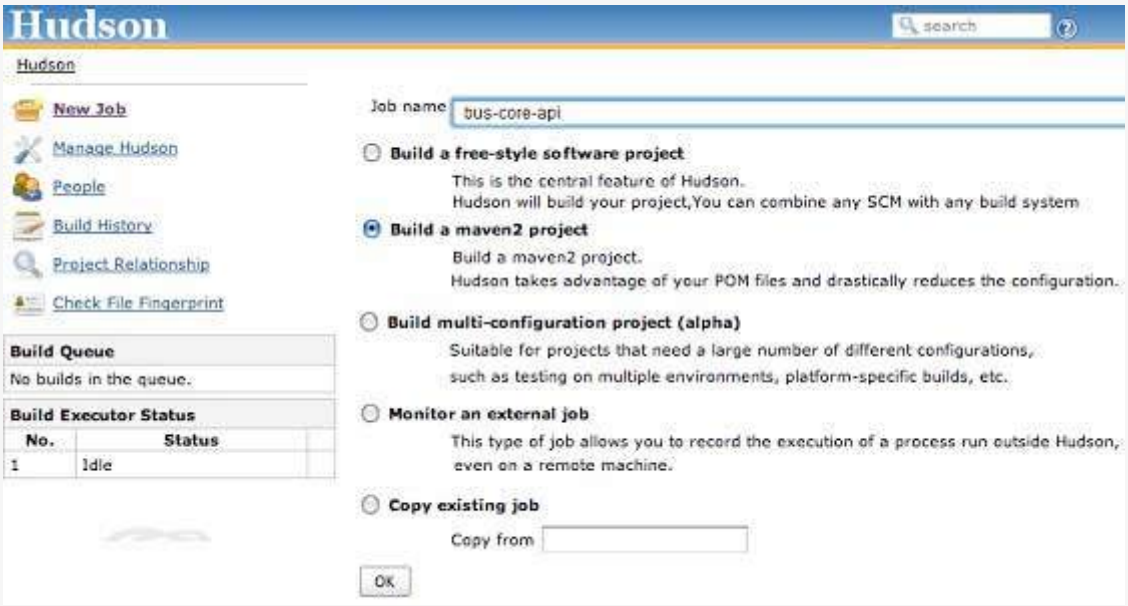
app-web-ui 构建成功后，Maven 将开始构建 app-desktop-ui 项目。

```
[INFO] ------------------------------------------------------------------------
[INFO] Building app-desktop-ui
[INFO]    task-segment: [package]
[INFO] ------------------------------------------------------------------------
...
[INFO] [jar:jar {execution: default-jar}]
[INFO] Building jar: C:\MVN\app-desktop-ui\target\
app-desktop-ui-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
```

```
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
```

使用持续集成服务器（CI）

如果使用 CI 服务器更，我们每次的一个新项目，比如说实例中的 app-mobile-ui，添加为依赖 bus-core-api 项目时，开发者就不需要更新 bus-core-api 项目的 pom。Hudson 将会借助 Maven 的依赖管理功能实现工程的自动化创建。



Hudson 把每个项目构建当成一次任务。在一个项目的代码提交到 SVN （或者任何映射到 Hudson 的代码管理工具）后，Hudson 将开始项目的构建任务，并且一旦此构建任务完成，Hudson 将自动启动其他依赖的构建任务（其他依赖项目的构建）。在上面的例子中，当 bus-core-ui 源代码在 SVN 更新后，Hudson 开始项目构建。一旦构建成功，Hudson 自动地查找依赖的项目，然后开始构建 app-web-ui 和 app-desktop-ui 项目。

← Maven 快照(SNAPSHOT) Maven 依赖管理 →