|  |
| --- |
| 作为一名程序员，在开发的过程中，经常需要面对不同的运行环境（开发环境、测试环境、生产环境、内网环境、外网环境等等），在不同的环境中，相关的配置一般不一样，比如数据源配置、日志文件配置、以及一些软件运行过程中的基本配置。每次在不同环境部署程序时，都需要修改相应的配置文件，使之完成环境的配置。这么做存在一个比较大的问题：每次修改配置非常麻烦，而且配置错误会产生不可预估的影响，比如，在发布生产环境时用的开发环境的配置还好，但如果在开发环境下用生产环境的数据，将会造成生产数据的污染，导致生产环境崩溃。  目前JAVA相关的项目基本都是使用Maven来进行构建。在maven中实现多环境的构建可移植性需要使用profile，通过不同的环境激活不同的profile来达到构建的可移植性。 |

# 项目的父pom中配置profile

<profiles> 建议放在 <build>后

|  |
| --- |
| <!-- 配置文件, 线上使用 mvn 打包添加 -Pproduct 变量 -->  <profiles>  <profile>  <id>development</id>  <properties>  <profiles.active>development</profiles.active>  </properties>  <!-- 默认激活 -->  <activation>  <activeByDefault>true</activeByDefault>  </activation>  </profile>  <profile>  <id>test</id>  <properties>  <profiles.active>test</profiles.active>  </properties>  </profile>  <profile>  <id>product</id>  <properties>  <profiles.active>product</profiles.active>  </properties>  </profile>  </profiles> |

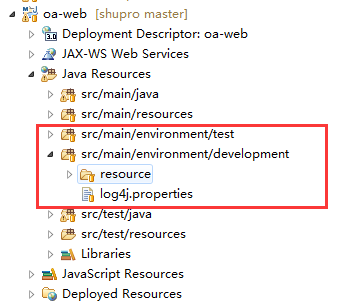
这里定义了三个环境，development（开发环境）、test（测试环境）、product（生产环境），其中开发环境是默认激活的（activeByDefault为true），这样如果在不指定profile时默认是开发环境。

<build>里面的<resources>

|  |
| --- |
| <resources>  <resource>  <directory>src/main/java</directory>  <includes>  <include>\*\*/\*.properties</include>  <include>\*\*/\*.xml</include>  </includes>  </resource>  <resource>  <directory>src/main/resources</directory>  </resource>  <!-- profile配置对应的配置文件路径 -->  <resource>  <directory>src/main/environment/${profiles.active}</directory>  </resource>  </resources> |

这样在激活指定的profile时，会加载指定目录下的配置文件，如当前激活的是development profile，那么这个资源目录就是src/main/environment/development。这样就达到了不同环境加载不同配置的目的。

# 工程目录



src/main/environment/development 这个是资源文件夹，其下的文件实际就是

src/main/resources 里面的，只是将其提取出来利用配置。

不要只建立src/main/environment 资源文件夹，然后新建development 文件夹，这样会读取不到，因为配置的是读取src/main/environment/development这个资源文件夹下的内容。

# 项目编译生成

由于默认的profile是development，所以如果我们不指定profile，那么加载就是开发环境development下的配置文件了。即我们在本地开发测试时，不用关心profile的问题。

部署时，如果想修改配置，方式一：改pom.xml文件将其激活放入指定位置。

方式二：命令加上相应参数，mvn clean install -Pproduct（如果集成eclipse注意去掉mvn）

如果使用eclipse建议使用方式一