

Maven 实现环境隔离

一、前言

通常，一个项目在本地开发完成后，需要提交到测试环境进行测试，测试完成后最终放到生成环境运行。但是，这 3 个运行环境的配置必然有所不同。当每次提交到不同的环境都需要修改项目的配置，这些操作显得非常繁琐且容易出错。

本文将介绍使用 Maven 实现环境隔离的小技巧来避免人工修改出错，解决上述问题。

二、实现方式

演示环境：IDEA

2.1 修改 pom.xml

1. 在 pom.xml 文件的 build 节点下设置 resources 节点:

```
<build>
  <resources>
    <resource>
      <directory>src/main/resources</directory>
      <filtering>true</filtering>
    </resource>
  </resources>
</build>
```

1. 在 pom.xml 文件中设置 3 个 profile 节点，分别对应 dev（开发环境）、test（测试环境）和 pro（生产环境）。这 3 个 profile 节点定义不同环境的需要的配置信息。

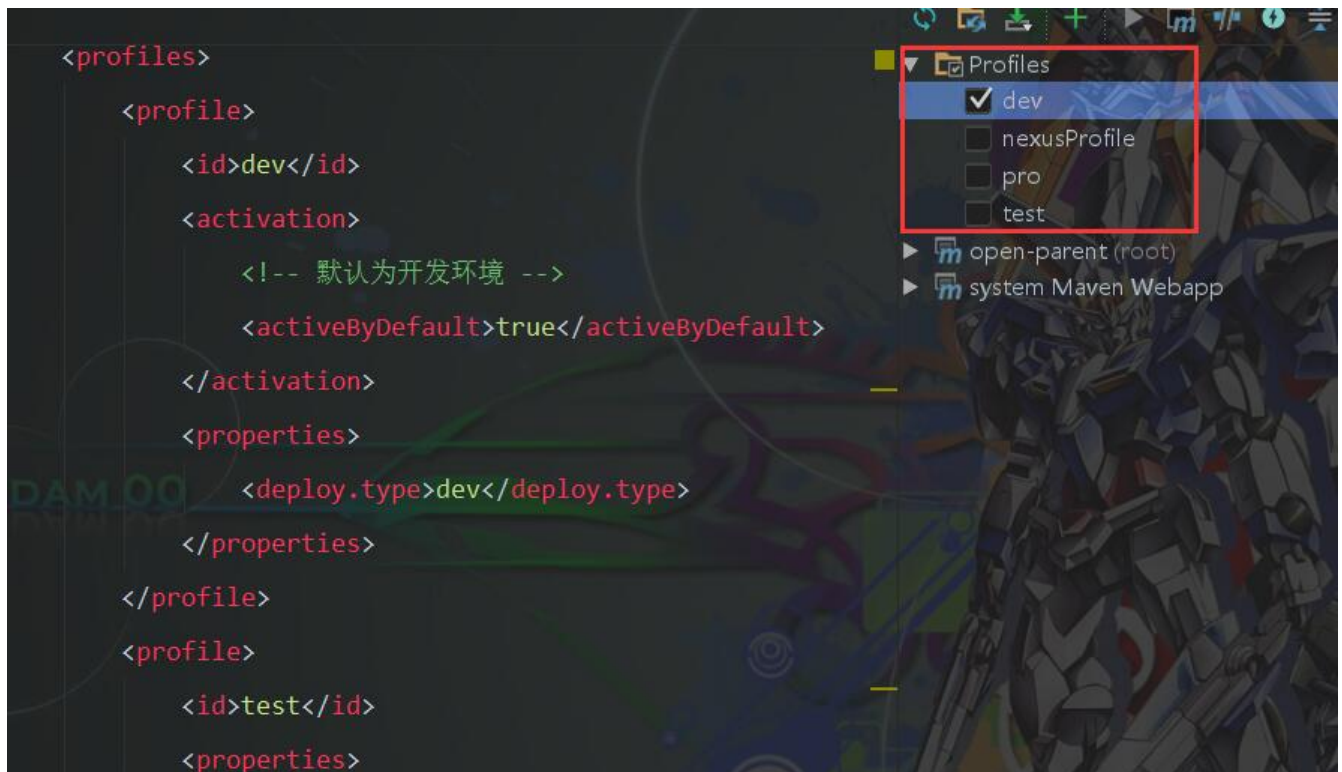
```
<profiles>
  <profile>
    <id>dev</id>
    <activation>
      <!-- 默认为开发环境 -->
      <activeByDefault>true</activeByDefault>
    </activation>
    <properties>

<env.jdbc.url>jdbc:mysql://127.0.0.1:3380/open_system_dev</env.jdbc.url>
      <env.jdbc.username>root</env.jdbc.username>
      <env.jdbc.password>tiger</env.jdbc.password>
    </properties>
  </profile>
  <profile>
    <id>test</id>
    <properties>

<env.jdbc.url>jdbc:mysql://127.0.0.1:3380/open_system_test</env.jdbc.url>
      <env.jdbc.username>root</env.jdbc.username>
      <env.jdbc.password>tiger</env.jdbc.password>
    </properties>
  </profile>
  <profile>
    <id>pro</id>
    <properties>

<env.jdbc.url>jdbc:mysql://127.0.0.1:3380/open_system_pro</env.jdbc.url>
      <env.jdbc.username>root</env.jdbc.username>
      <env.jdbc.password>tiger</env.jdbc.password>
    </properties>
  </profile>
</profiles>
```

此时，点击 IDEA 右侧的 Maven Project 可以看到 Profiles 节点，选择对应的 profile 即可切换运行环境，如下图：



2.2 修改 properties 文件

db.properties

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=${env.jdbc.url}?characterEncoding=utf-8&allowMultiQueries=true&serverTimezone=UTC
jdbc.username=${env.jdbc.username}
jdbc.password=${env.jdbc.password}
```

使用在 pom.xml 文件中定义的变量替换配置文件中的值。

三、测试

我们点击 IDEA 的 Maven Project，选择 dev 环境。使用 Maven 自带的打包工具打包项目，查看其打包后的配置文件信息。如下图：

