

菩提树下的杨过

博客园

首页

新随笔

订阅

管理

公告



注：技术发展日新月异，随着时间推移，无法保证本博客所有内容的正确性，如有误导，请大家见谅！

英文名： Jimmy.Yang

中文名： 杨俊明

一妹儿： yjmyzz#126.com

扣扣： 二妻八舅么舅五零妻

昵称： 菩提树下的杨过

园龄： 11年1个月

荣誉： 推荐博客

粉丝： 3638

关注： 6

+加关注

搜索

 找我看

随笔分类(1410)

- 01.WebForm/MVC(105)
- 02.FrameWork/C#(142)
- 03.WinForm(22)
- 04.JavaScript(85)
- 05.CSS(25)
- 06.Design Pattern(14)
- 07.WPF/Silverlight(118)
- 08.Database(86)
- 09.Open Source(148)
- 10.Delphi(17)
- 11.Ruby(16)
- 12.Flex/Flash/AS3.0(124)
- 13.C/Objective-C(15)
- 15.Java/Scala(288)
- 16.linux/mac(41)
- 17.Hadoop(17)
- 18.Python/ML/AI(31)
- 19.Others(116)

随笔档案(1176)

2018年12月 (11)

随笔-1176 文章-1 评论-2768

maven学习（下）利用Profile构建不同环境的部署包

接上回继续，项目开发好以后，通常要在多个环境部署，象我们公司多达5种环境：本机环境(**local**)、(开发小组内自测的)开发环境(**dev**)、(提供给测试团队的)测试环境(**test**)、预发布环境(**pre**)、正式生产环境(**prod**)，每种环境都有各自的配置参数，比如：数据库连接、远程调用的ws地址等等。如果每个环境build前手动修改这些参数，显然太不fashion.

maven早就考虑到了这些问题，看下面的pom片段：

```
1 <profiles>
2     <profile>
3         <!-- 本地环境 -->
4         <id>local</id>
5         <properties>
6             <db-
7                 url>jdbc:oracle:thin:@localhost:1521:XE</db-url>
8                 <db-username>***</db-username>
9                 <db-password>***</db-password>
10            </properties>
11        </profile>
12        <profile>
13            <!-- 开发环境 -->
14            <id>dev</id>
15            <properties>
16                <db-
17                    url>jdbc:oracle:thin:@172.21.129.51:1521:orcl</db-url>
18                    <db-username>***</db-username>
19                    <db-password>***</db-password>
20                </properties>
21            <!-- 默认激活本环境 -->
22            <activation>
23                <activeByDefault>true</activeByDefault>
24            </activation>
25        </profile>
26    </profiles>
```

profiles节点中，定义了二种环境：local、dev(默认激活dev环境)，可以在各自的环境中添加需要的property值，接下来修改build节点，参考下面的示例：

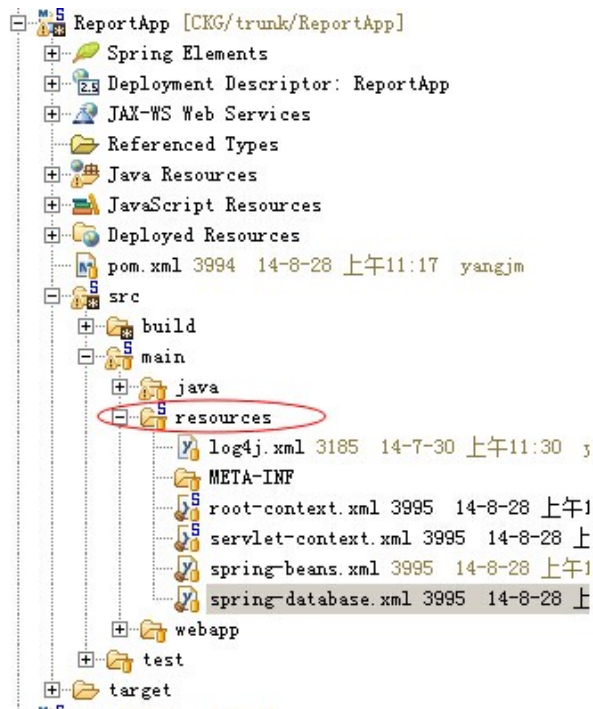
☐

2018年9月 (3)
2018年8月 (5)
2018年7月 (3)
2018年6月 (1)
2018年5月 (4)
2018年4月 (6)
2018年3月 (6)
2018年2月 (1)
2018年1月 (1)
2017年12月 (6)
2017年11月 (5)
2017年10月 (5)
2017年9月 (4)
2017年8月 (9)
2017年7月 (4)
2017年6月 (5)
2017年5月 (3)
2017年4月 (7)
2017年3月 (1)
2017年2月 (2)
2017年1月 (2)
2016年12月 (1)
2016年9月 (1)
2016年6月 (3)
2016年5月 (2)
2016年4月 (1)
2016年3月 (13)
2016年2月 (10)
2016年1月 (15)
2015年12月 (14)
2015年11月 (4)
2015年10月 (5)
2015年9月 (18)
2015年8月 (17)
2015年7月 (6)
2015年6月 (9)
2015年5月 (11)
2015年4月 (6)
2015年3月 (1)
2015年2月 (5)
2015年1月 (9)
2014年12月 (5)
2014年11月 (17)
2014年10月 (15)
2014年9月 (10)
2014年8月 (9)
2014年7月 (14)
2014年5月 (2)
2014年4月 (12)

```
1 <build>
2     <resources>
3         <resource>
4
5 <directory>src/main/resources</directory>
6         <filtering>true</filtering>
7     </resource>
8 </resources>
9     <plugins>
10        <plugin>
11
12 <groupId>org.apache.maven.plugins</groupId>
13        <artifactId>maven-compiler-
14plugin</artifactId>
15        <version>2.5.1</version>
16        <configuration>
17            <source>1.6</source>
18            <target>1.6</target>
19            <encoding>utf-8</encoding>
20        </configuration>
21    </plugin>
22 </plugins>
23 </build>
```

resource节点是关键，它表明了哪个目录下的配置文件（不管是xml配置文件，还是properties属性文件），需要根据profile环境来替换属性值。

通常配置文件放在resources目录下，build时该目录下的文件都会自动copy到class目录下



以上图为例，其中spring-database.xml的内容为：

[View Code](#)

2014年3月 (7)
2014年2月 (8)
2014年1月 (18)
2013年12月 (1)
2013年11月 (8)
2013年10月 (7)
2013年9月 (3)
2013年8月 (5)
2013年7月 (3)
2013年5月 (4)
2013年4月 (1)
2013年2月 (3)
2013年1月 (2)
2012年12月 (8)
2012年11月 (16)
2012年9月 (1)
2012年7月 (1)
2012年6月 (1)
2012年5月 (1)
2012年4月 (1)
2012年3月 (2)
2012年2月 (2)
2012年1月 (4)
2011年12月 (4)
2011年11月 (2)
2011年10月 (4)
2011年9月 (5)
2011年8月 (2)
2011年7月 (1)
2011年6月 (12)
2011年5月 (15)
2011年4月 (16)
2011年3月 (11)
2011年2月 (9)
2011年1月 (8)
2010年12月 (10)
2010年11月 (13)
2010年10月 (10)
2010年9月 (5)
2010年8月 (12)
2010年7月 (32)
2010年6月 (19)
2010年5月 (21)
2010年4月 (30)
2010年3月 (49)
2010年2月 (23)
2010年1月 (22)
2009年12月 (47)
2009年11月 (26)

各属性节点的值，用占位符"**`${属性名}`**"占位，maven在package时，会根据profile的环境自动替换这些占位符为实际属性值。

默认情况下：

maven package

将采用默认激活的profile环境来打包，也可以手动指定环境，比如：

maven package -P dev

将自动打包成dev环境的部署包(注：参数P为大写)

最后再给2个实例的运用例子：

1、开发环境与生产环境数据源采用不同方式的问题

本机开发时为了方便，很多开发人员喜欢直接用JDBC直接连接数据库，这样修改起来方便：

```
1 <bean id="dataSource"
class="org.apache.commons.dbcp.BasicDataSource"
2     destroy-method="close">
3     <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver" />
4     <property name="url" value="${db-url}" />
5     <property name="username" value="${db-
username}" />
6     <property name="password" value="${db-
password}" />
7     <property name="defaultAutoCommit"
value="false" />
8     <property name="initialSize" value="2" />
9     <property name="maxActive" value="10" />
10    <property name="maxWait" value="60000" />
11 </bean>
```

而生产环境，通常是在webserver(比如weblogic上)配置一个JNDI数据源，

```
1 <bean id="dataSource"
class="org.springframework.jndi.JndiObjectFactoryBean">
2     <property name="jndiName" value="appDS" />
3 </bean>
```

如果每次发布生产前，都要手动修改，未免太原始，可以通过maven的profile来解决

先把配置文件改成

- 2009年10月 (12)
- 2009年9月 (7)
- 2009年8月 (14)
- 2009年7月 (2)
- 2009年6月 (27)
- 2009年5月 (14)
- 2009年4月 (4)
- 2009年3月 (14)
- 2009年2月 (9)
- 2009年1月 (3)
- 2008年11月 (5)
- 2008年10月 (8)
- 2008年9月 (34)
- 2008年8月 (40)
- 2008年6月 (2)
- 2008年5月 (8)
- 2008年4月 (10)
- 2008年3月 (20)
- 2008年2月 (12)
- 2008年1月 (24)
- 2007年12月 (36)
- 2007年11月 (27)

友情链接

- [cloudgamer](#)
- [JeffreyZhao](#)
- [包建强](#)
- [计算机的潜意识](#)
- [司徒正美](#)
- [万一](#)
- [张逸:晴窗笔记](#)

积分与排名

- 积分 - 1516908
- 排名 - 41

最新评论

- 1. Re:mac:在当前文件夹打开terminal终端
666
--xgqfrms
- 2. Re:ZooKeeper 笔记(3) 实战应用之【统一配置管理】
请问下楼主客户端如何才能自动感知配置变化?
--qing0506
- 3. Re:spring cloud: 使用consul来替换eureka
@ylduB站的小伙伴? ...
--言小川

```
1 <bean id="${db-source-jdbc}"
class="org.apache.commons.dbcp.BasicDataSource"
2     destroy-method="close">
3     <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver" />
4     <property name="url" value="${db-url}" />
5     <property name="username" value="${db-
username}" />
6     <property name="password" value="${db-
password}" />
7     <property name="defaultAutoCommit"
value="false" />
8     <property name="initialSize" value="2" />
9     <property name="maxActive" value="10" />
10    <property name="maxWait" value="60000" />
11 </bean>
12
13 <bean id="${db-source-jndi}"
class="org.springframework.jndi.JndiObjectFactoryBean">
14     <property name="jndiName" value="appDS" />
15 </bean>
```

即用占位符来代替bean的id，然后在pom.xml里类似下面设置

```
1 <profile>
2 <!-- 本机环境 -->
3 <id>local</id>
4 <properties>
5     ...
6     <db-source-jdbc>dataSource</db-source-
jdbc>
7     <db-source-jndi>NONE</db-source-jndi>
8     <db-
url>jdbc:oracle:thin:@172.21.129.51:1521:orcl</db-url>
9     <db-username>mu_fsu</db-username>
10    <db-password>mu_fsu</db-password>
11    ...
12 </properties>
13 <!-- 默认激活本环境 -->
14 <activation>
15     <activeByDefault>true</activeByDefault>
16 </activation>
17 </profile>
18 <profile>
19 <!-- 生产环境 -->
20 <id>pro</id>
21 <properties>
22     ...
```

4. Re:intellij idea 高级用法之：集成 JIRA、UML类图插件、集成SSH、集成FTP、Database管理

666

--0o一泓秋水o0

5. Re:ZooKeeper 笔记(5)

ACL(Access Control List)访问控制列表

赞一个，楼主对zookeeper的acl总结的非常好，很有收获。不过对于超级管理员密码的设置有一个更简便的方法：export ZK_CLASSPATH=/etc/zookeeper/conf/:/us.....

--CREATE_17

```

23         <db-source-jdbc>NONE</db-source-jdbc>
24         <db-source-jndi>dataSource</db-source-
jndi>
25         ...
26     </properties>
27 </profile>
28 </profiles>

```

这样，mvn clean package -P local打包本地开发环境时，将生成

```

1     <bean id="dataSource"
class="org.apache.commons.dbcp.BasicDataSource"
2         destroy-method="close">
3         <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver" />
4         <property name="url"
value="jdbc:oracle:thin:@172.21.129.***:1521:orcl" />
5         <property name="username" value="***" />
6         <property name="password" value="***" />
7         <property name="defaultAutoCommit"
value="false" />
8         <property name="initialSize" value="2" />
9         <property name="maxActive" value="10" />
10        <property name="maxWait" value="60000" />
11    </bean>
12
13    <bean id="NONE"
class="org.springframework.jndi.JndiObjectFactoryBean">
14        <property name="jndiName" value="appDS" />
15    </bean>

```

而打包生产环境 mvn clean package -P pro时，生成

```

1     <bean id="NONE"
class="org.apache.commons.dbcp.BasicDataSource"
2         destroy-method="close">
3         <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver" />
4         <property name="url" value="${db-url}" />
5         <property name="username" value="${db-
username}" />
6         <property name="password" value="${db-
password}" />
7         <property name="defaultAutoCommit"
value="false" />
8         <property name="initialSize" value="2" />
9         <property name="maxActive" value="10" />

```

```
10         <property name="maxWait" value="60000" />
11     </bean>
12
13     <bean id="dataSource"
14         class="org.springframework.jndi.JndiObjectFactoryBean">
15         <property name="jndiName" value="appDS" />
16     </bean>
```

spring配置的其它跟数据库相关的bean，约定引用dataSource这个名称的bean即可

2、不同webserver环境，依赖jar包，是否打包的问题

weblogic上，允许多个app，把共用的jar包按约定打包成一个war文件，以library的方式部署，然后各应用在WEB-INF/weblogic.xml中，用类似下面的形式

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <weblogic-web-app
3     xmlns="http://www.bea.com/ns/weblogic/90">
4     ...
5     <library-ref>
6         <library-name>my-share-lib</library-name>
7     </library-ref>
8 </weblogic-web-app>
```

指定共享library 的名称即可。这样的好处是，即节省了服务器开销，而且各app打包时，就不必再重复打包这些jar文件，打包后的体积大大减少，上传起来会快很多。

而其它webserver上却未必有这个机制，一般为了方便，我们开发时，往往采用一些轻量级的webserver，比如:tomcat,jetty,jboss之类，正式部署时才发布到weblogic下，这样带来的问题就是，本机打包时，要求这些依赖jar包，全打包到app的WEB-INF/lib下；而生产环境下，各应用的WEB-INF/lib下并不需要这些jar文件，同样还是用profile来搞定，先处理pom.xml，把依赖项改成类似下面的形式：

```
1     <dependency>
2         <groupId>dom4j</groupId>
3         <artifactId>dom4j</artifactId>
4         <version>1.6.1</version>
5         <scope>${jar.scope}</scope>
6     </dependency>
```



即scope这里，用一个占位符来代替，然后profile这样配置



```
1      <profile>
2          <!-- 本机环境 -->
3          <id>local</id>
4          <properties>
5              <jar.scope>compile</jar.scope>
6              ...
7          </properties>
8          <!-- 默认激活本环境 -->
9          <activation>
10             <activeByDefault>true</activeByDefault>
11          </activation>
12      </profile>
13      <profile>
14          <!-- 生产环境 -->
15          <id>pro</id>
16          <properties>
17              <jar.scope>provided</jar.scope>
18              ...
19          </properties>
20      </profile>
```

在maven里，如果一个依赖项的scope是provided，表示由容器提供，打包时将不会打包进最终的package里，所以这样配置后，生产环境打包时，依赖项的scope全变成了provided，即不打包进war文件，而本机环境下，因为scope是compile，所以会打包到war里

作者: [菩提树下的杨过](#)

出处: <http://yjmyzz.cnblogs.com>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [15.Java/Scala](#)

标签: [java](#), [maven](#)

好文要顶

关注我

收藏该文



[菩提树下的杨过](#)

关注 - 6

粉丝 - 3638

荣誉: [推荐博客](#)

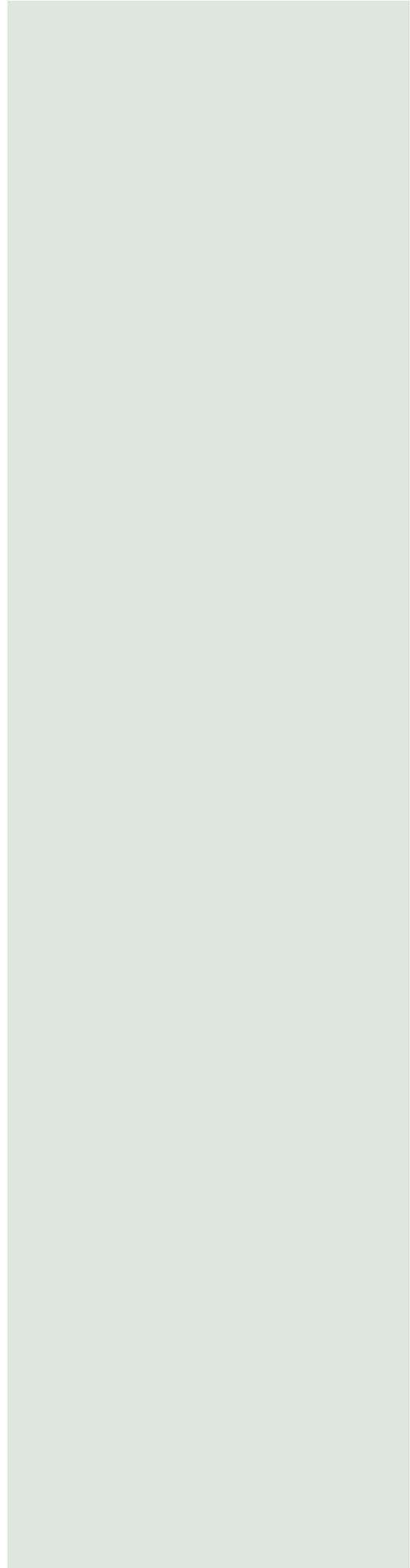
[+加关注](#)

17

推荐

0

反对



« 上一篇: [JSP中的Servlet及Filter](#)
» 下一篇: ["org.jboss.netty.internal.LoggerConfigurator".DESCRIBED is already registered 的解决办法](#)
posted @ 2014-08-28 10:49 菩提树下的杨过 阅读(47285) 评论(4)
[编辑](#) [收藏](#)

评论列表

- #1楼 2015-07-22 16:31 飞翔的茶叶

写得太好了。

支持(0) 反对(0)
- #2楼 2015-08-04 17:21 暗夜心慌方

nice

支持(0) 反对(0)
- #3楼 2016-04-11 16:34 yayad

楼主写的很详细，顶

支持(0) 反对(0)
- #4楼 2016-09-29 11:10 wanggang6688

写得很精辟，我再学习你其它的博客，相信也会有收获

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！

- 相关博文：
- [精度最高的圆周率](#)
 - [你好吗？](#)
 - [再谈单元测试](#)
 - [暑假开始了，第一天的值班。](#)
 - [健康的水果春捲](#)

最新新闻：

- 辣番茄可能即将出现，研究者表示这是为了大量生产辣椒素
 - LG化学加码电池业务：追投72亿元在中国建新厂
 - 嫦娥四号着陆器月午工作正常 地形地貌相机顺利完成360度环拍
 - 压垮互联网从业者的那根稻草：用命搏一个未来
 - 90后女孩入选马云乡村教师奖：把优质资源带给学生
- » 更多新闻...