



链滴

Tomcat Jetty 容器中部署项目的注意事项与区别

作者: [linyiheng](#)

原文链接: <https://ld246.com/article/1503241166520>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>${jsp.version}</version>
  <!--scope>provided</scope-->
  <exclusions>
    <exclusion>
      <artifactId>servlet-api</artifactId>
      <groupId>javax.servlet</groupId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>${servlet.version}</version>
  <!--scope>provided</scope-->
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>${jstl.version}</version>
  <!--scope>provided</scope-->
</dependency>

```

在jetty容器中运行需要打开provided,因为jetty容器中已经自带了servlet,jstl和jsp的实现。但当项目要部署到Tomcat中时,由于Tomcat容器未带有servlet,jstl和jsp的实现所以需要关闭provided,将相关的包代入。

在Jetty环境中若Jetty不设置项目的基本Context则相应的\${ctx},即Context Path无法找到相应的项路径,所以配置Jetty项目时需要配置好相应的Context。

```

<plugin>
  <groupId>org.eclipse.jetty</groupId>
  <artifactId>jetty-maven-plugin</artifactId>
  <version>9.1.6.v20160112</version>
  <configuration>
    <stopPort>9966</stopPort>
    <stopKey>stop-jetty-for-it</stopKey>
    <webAppConfig>
      <contextPath>/${project.build.finalName}</contextPath>
    </webAppConfig>
  </configuration>
  <!--dependencies>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.38</version>
    </dependency>
  </dependencies-->

</plugin>

```

更高版本的mysql-connector-java在配置到jetty时需要被jetty所依赖,但Tomcat不需要类似的操作。