

Java Servlet 應用程式與 Maven

這是一個完整的指南，介紹如何在 WebSphere Liberty Profile (WLP) 上使用 Maven 來構建項目並使用 Liberty Maven 插件來部署簡單的 Java Servlet 應用程序。這個範例創建了一個基本的 Servlet，當訪問時返回 “Hello World!”，然後使用 Maven 將其打包成 WAR 文件並部署到 WLP。我們假設您已經設置了 WLP 並啟用了 servlet-4.0 功能，因為這提供了必要的 Servlet 支持。

分步指南

1. 創建 Maven 項目結構

首先設置標準的 Maven Web 應用程序結構。您可以手動創建這個結構，或者使用 Maven 來生成它。

- **目錄結構:**

```
SimpleServletApp/
  pom.xml
  src/
    main/
      java/
        com/
          example/
            HelloServlet.java
    webapp/
      WEB-INF/
        web.xml
```

- **可選：使用 Maven 生成:** 運行以下命令來創建結構，然後根據需要進行調整：

```
mvn archetype:generate -DgroupId=com.example -DartifactId=simple-servlet-app -DarchetypeArtifactId=maven-
```

這將創建一個基本的 Web 應用程序結構，您將在後續步驟中進行修改。

2. 編寫 Servlet 代碼

在 src/main/java/com/example/ 中創建一個名為 HelloServlet.java 的文件，內容如下：

```
package com.example;

import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class HelloServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().write("Hello World!");
    }
}

```

- 說明：**這個 Servlet 回應 HTTP GET 請求，返回 “Hello World!” 以純文本格式。它使用簡單的 `doGet` 方法，並避免使用註解以與顯式的 `web.xml` 配置兼容。

3. 創建 `web.xml` 部署描述符

在 `src/main/webapp/WEB-INF/` 中創建一個名為 `web.xml` 的文件，內容如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
          version="4.0">

    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>com.example.HelloServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>

```

- 說明：** `web.xml` 文件定義了 `HelloServlet` 類並將其映射到 `/hello` URL 模式。這是必要的，因為我們沒有使用 `@WebServlet` 註解。

4. 配置 Maven `pom.xml`

在 `SimpleServletApp/` 目錄中創建或更新 `pom.xml`，內容如下：

```

<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>com.example</groupId>
    <artifactId>simple-servlet-app</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <dependencies>
        <!-- Servlet API (provided by WLP) -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>4.0.1</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <!-- Maven WAR Plugin to build the WAR file -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.3.1</version>
                <configuration>
                    <finalName>myapp</finalName>
                </configuration>
            </plugin>
            <!-- Liberty Maven Plugin for deployment -->
            <plugin>
                <groupId>io.openliberty.tools</groupId>
                <artifactId>liberty-maven-plugin</artifactId>

```

```

<version>3.3.4</version>
<configuration>
    <installDirectory>/opt/ibm/wlp</installDirectory>
    <serverName>myServer</serverName>
    <appsDirectory>dropins</appsDirectory>
    <looseApplication>false</looseApplication>
    <stripVersion>true</stripVersion>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

- **說明：**

- **坐標**：使用 groupId、artifactId 和 version 定義項目，並將 packaging 設置為 war 以供 Web 應用程序使用。
- **屬性**：將 Java 8 設置為源和目標版本。
- **依賴項**：包含 Servlet API，範圍為 provided，因為它在運行時由 WLP 提供。
- **Maven WAR 插件**：使用 <finalName> 將 WAR 文件名配置為 myapp.war。
- **Liberty Maven 插件**：配置將應用程序部署到 /opt/ibm/wlp 的 Liberty 伺服器，伺服器名稱為 myServer，部署到 dropins 目錄。

5. 构建項目

從 SimpleServletApp/ 目錄中使用 Maven 构建 WAR 文件：

```
mvn clean package
```

- **結果**：這將編譯 Servlet，將其與 web.xml 打包到 target/myapp.war，並準備部署。

6. 在 WebSphere Liberty 上部署和運行

確保您的 Liberty 伺服器 (myServer) 已經設置並啟用了 servlet-4.0 功能。檢查您的 server.xml，確保包含以下內容：

```

<featureManager>
    <feature>servlet-4.0</feature>
</featureManager>

```

使用 Liberty Maven 插件部署和運行應用程序：

```
mvn liberty:run
```

- **發生的事情：**

- 如果伺服器尚未運行，則在前台啟動 Liberty 伺服器。
- 自動將 myapp.war 部署到 dropins 目錄。
- 保持伺服器運行，直到停止。

- **驗證部署：**查找類似以下的日誌消息：

```
[AUDIT    ] CWWKT0016I: Web application available (default_host): http://localhost:9080/myapp/
```

日誌通常位於 /opt/ibm/wlp/usr/servers/myServer/logs/console.log °

7. 訪問應用程序

打開瀏覽器並導航到：

```
http://localhost:9080/myapp/hello
```

- **預期輸出：**

Hello World!

- **URL 解析：**

- 9080：WLP 的默認 HTTP 端口。
- /myapp：來自 WAR 文件名 (myapp.war) 的上下文根。
- /hello：來自 web.xml 的 URL 模式。

8. 停止伺服器

由於 mvn liberty:run 在前台運行伺服器，因此可以通過在終端中按 Ctrl+C 來停止它。

注意事項

- **先決條件：**

- 必須在系統上安裝並配置 Maven。
- 必須在 /opt/ibm/wlp 安裝 Liberty，並且伺服器實例 myServer 必須存在。如果您的設置不同（例如 /usr/local/wlp 或 defaultServer），請調整 pom.xml 中的 installDirectory 和 serverName。

- 必須在 server.xml 中啟用 servlet-4.0 功能。

- **替代部署：**

- 要單獨構建和部署：

```
mvn clean package  
mvn liberty:deploy
```

如果需要，手動啟動伺服器：

```
/opt/ibm/wlp/bin/server start myServer
```

- **端口配置：**如果您的 Liberty 伺服器使用不同的 HTTP 端口，請檢查 server.xml 中的 <httpEndpoint>，並相應地調整 URL。
- **上下文根：**重命名 pom.xml 中的 <finalName> (例如 <finalName>app</finalName>)，將上下文根更改為 /app。
- **故障排除：**

- 如果部署失敗，請驗證 pom.xml 中的 installDirectory 和 serverName。
 - 檢查 /opt/ibm/wlp/usr/servers/myServer/logs/messages.log 中的錯誤。
 - 確保使用 Java 8 或更高版本，與 maven.compiler 設置匹配。

摘要

本指南展示了如何：1. 使用 Maven 設置項目，並創建簡單的 Servlet (HelloServlet.java) 和 web.xml。2. 配置 pom.xml，包括 Servlet API、Maven WAR 插件和 Liberty Maven 插件。3. 使用 mvn clean package 將應用程序打包成 myapp.war。4. 使用 mvn liberty:run 將其部署和運行在 WLP 上。5. 訪問 <http://localhost:9080/myapp/hello> 以獲取 “Hello World!”。

這提供了一個簡潔的、基於 Maven 的方法來開發和部署 Servlet 應用程序到 WebSphere Liberty Profile。