

WebSphere Liberty - 你好，世界應用程式

這裡有一個簡單的範例，展示如何在您的 WebSphere Liberty 伺服器上運行一個“Hello World”應用程式，假設您已經按照之前的指示啟用了 `javaee-8.0` 功能。這個範例使用了一個基本的 JSP (JavaServer Pages) 文件，並通過 `dropins` 目錄進行部署，以簡化操作。

分步指南

1. 創建應用程式目錄和文件 您將通過在 Liberty 伺服器的 `dropins` 目錄中放置一個 JSP 文件來創建一個小型的 Web 應用程式。`dropins` 目錄允許 Liberty 自動檢測和部署應用程式。

- **定位 `dropins` 目錄**：導航到伺服器文件夾中的 `dropins` 目錄。如果您的 Liberty 安裝位於 `/opt/ibm/wlp`，並且您的伺服器名稱為 `myServer`，則路徑為：

```
/opt/ibm/wlp/usr/servers/myServer/dropins
```

將 `/opt/ibm/wlp` 替換為您的實際 Liberty 安裝目錄，並將 `myServer` 替換為您的伺服器名稱。

- **創建一個爆破 WAR 目錄**：在 `dropins` 目錄中，創建一個名為 `myApp.war` 的目錄。`.war` 擴展名告訴 Liberty 將其視為一個 Web 應用程式。使用以下命令：

```
mkdir -p /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war
```

- **創建 `index.jsp` 文件**：在 `myApp.war` 中，創建一個名為 `index.jsp` 的文件，內容如下，以顯示“Hello World!”：

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

您可以使用以下命令直接創建它：

```
echo '<html><body><h2>Hello World!</h2></body></html>' > /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war
```

或者，使用文本編輯器創建 `index.jsp` 並將其保存到該位置。

2. 開始伺服器（如果尚未運行） 如果您的伺服器未運行，您需要啟動它，以便它可以部署和提供應用程式。

- **導航到 `bin` 目錄**：進入 Liberty 安裝中的 `bin` 目錄：

```
cd /opt/ibm/wlp/bin
```

- **啟動伺服器**：以前台模式運行伺服器，以直接查看輸出：

```
./server run myServer
```

或者，在後台運行它：

```
./server start myServer
```

如果伺服器已經運行，則跳過此步驟——Liberty 將自動檢測新應用程序。

3. 驗證應用程序部署

Liberty 將在 dropins 目錄中檢測到 myApp.war 應用程序時自動部署它。

- **檢查控制台輸出：**如果您以前台模式啟動了伺服器，請查找類似以下的消息：

```
[AUDIT ] CWWKT0016I: Web application available (default_host): http://localhost:9080/myApp/  
這確認應用程序已部署並可用。
```

- **檢查日誌（如果在後台運行）：**如果伺服器在後台運行，請在以下位置查找相同的部署消息：

```
/opt/ibm/wlp/usr/servers/myServer/logs/console.log
```

4. 訪問應用程序

現在應用程序已部署，您可以在 Web 瀏覽器中查看它。

- **打開您的瀏覽器：**導航到：

```
http://localhost:9080/myApp/
```

上下文根 /myApp 來自 myApp.war 目錄名稱，index.jsp 默認作為歡迎文件提供。

- **預期結果：**您應該看到一個顯示以下內容的頁面：

```
Hello World!
```

以大字體顯示。

注意事項

- **端口號：**此範例使用端口 9080，這是 Liberty 的默認 HTTP 端口。如果您配置了不同的端口，請相應地調整 URL（例如，http://localhost:YOUR_PORT/myApp/）。
- **權限：**確保您對 dropins 目錄具有寫權限。
- **JSP 支持：**javaee-8.0 功能包括 JSP 支持，因此不需要額外配置。
- **故障排除：**如果頁面未加載，請檢查伺服器日誌 /opt/ibm/wlp/usr/servers/myServer/logs/messages.log 中的錯誤，或者確保伺服器正在運行，並且文件位於正確位置。

這個簡單的範例展示了如何在 WebSphere Liberty 伺服器上使用 dropins 目錄進行自動部署來部署和運行“Hello World”應用程序。享受您的第一個 Liberty 應用程序！

這裡是一個解釋，當在 WebSphere Liberty 上部署“Hello World”應用程序時，它是如何工作的：

1. Dropins 目錄

WebSphere Liberty 提供了一個方便的功能，通過其 `dropins` 目錄部署應用程序。這是一個特殊的文件夾，您可以簡單地將應用程序放置在其中（作為 WAR（Web Application Archive）文件或爆破 WAR 目錄），Liberty 將自動檢測並部署它。與傳統部署方法不同，您不需要手動在 `server.xml` 文件中配置應用程序。當 Liberty 開始或在 `dropins` 文件夾中注意到變化（例如添加新應用程序）時，它會自動啟動部署過程。

2. 使用爆破 WAR 目錄

在這個範例中，應用程序作為一個名為 `myApp.war` 的爆破 WAR 目錄進行部署，而不是一個單一打包的 WAR 文件。爆破 WAR 只是一個包含 Web 應用程序的所有內容（例如 HTML、JSP 文件和其他資源）的文件夾，以未壓縮的形式。Liberty 將這個目錄視為 WAR 文件，將其部署為一個完全功能的 Web 應用程序。這種方法在開發過程中特別有用，因為您可以直接編輯文件（例如調整 HTML 或 JSP），而不需要重新打包所有內容到 WAR 文件中。

3. JSP 文件

這個“Hello World”應用程序的核心是一個名為 `index.jsp` 的文件，這是一個 JavaServer Page (JSP)。這個文件包含基本的 HTML 以在屏幕上顯示“Hello World!”，並且可以包含 Java 代碼（雖然在這種情況下保持簡單）。當您訪問應用程序時，Liberty 會動態地將 JSP 編譯為一個 servlet——一個小型的 Java 程序，生成網頁並將其提供給您的瀏覽器。

4. 啟用 Java EE 功能

為了使所有這些工作，Liberty 依賴於其配置文件 `server.xml` 中啟用的特定功能。這裡啟用了 `javaee-8.0` 功能，該功能提供對 JSP、Servlet 和 Java Enterprise Edition (EE) 8 平台的其他組件的支持。這個功能確保 Liberty 加載了必要的庫和設置，以平滑地運行應用程序。

5. 自動部署過程

一旦您將 `myApp.war` 目錄放入 `dropins` 文件夾並啟動 Liberty（或如果它已經運行），伺服器將自動檢測並部署應用程序。您將在 Liberty 的輸出中看到日誌消息，指出應用程序已啟動並可在特定 URL 上使用。這種無需手動操作的部署過程使得快速部署和測試應用程序變得簡單。

6. 訪問應用程序：上下文根

您可以訪問應用程序的 URL 由其 **上下文根** 確定，這只是 WAR 文件或目錄的名稱。在這種情況下，由於目錄名為 `myApp.war`，上下文根成為 `/myApp`。默認情況下，Liberty 在端口 9080 上運行，因此訪問應用程序的完整 URL 是 `http://localhost:9080/myApp/`。

7. 提供歡迎文件

當您在瀏覽器中訪問 `http://localhost:9080/myApp/` 時，Liberty 需要確定顯示什麼。它查找一個 **歡迎文件**——當沒有請求特定文件時要提供的默認頁面。在這種情況下，`index.jsp` 位於 `myApp.war` 目錄的根目錄中，並被默認識別為歡迎文件。Liberty 編譯並提供 `index.jsp`，然後在您的屏幕上顯示 “Hello World!”。

總結

這個 “Hello World” 應用程序展示了 WebSphere Liberty 如何簡化 Web 應用程序的部署。通過利用 `dropins` 目錄進行自動部署，支持爆破 WAR 以便於編輯，並使用 `javaee-8.0` 功能以實現 Java EE 兼容性，Liberty 讓您能夠快速部署和測試簡單的 Web 應用程序。結果是一個輕量級、開發者友好的方式來在 `http://localhost:9080/myApp/` 查看 “Hello World!”，這對於學習或原型設計來說是完美的。