

Spring Boot 2.7.18 in WebSphere Liberty Profile

當使用 Spring Boot 2.7.18 將應用程式部署到 WebSphere Liberty Profile (WLP) 時，根據提供的指南，好消息是不需要對 `DemoApplication` 類別或使用 `SpringBootServletInitializer` 進行任何更改。Spring Boot 2.7.18 是 2.7.x 系列中的一個補丁版本，這意味著它包含錯誤修復和次要改進，但不會引入影響標準 WAR 部署配置的重大變更。以下，我將解釋為什麼會這樣，並確認設置仍然完全兼容。

了解設置

指南配置 Spring Boot 應用程式以 WAR 文件的形式部署到 WLP，方法如下：

1. 擴展 `SpringBootServletInitializer`：

- `DemoApplication` 類別擴展 `SpringBootServletInitializer` 以啟用到外部 Servlet 容器（如 WLP）的部署。這個類別是必要的，因為雖然 Spring Boot 默認以嵌入式伺服器運行作為獨立的 JAR，但部署到 WLP 需要與其 Servlet 容器集成。
- 覆蓋 `configure` 方法以指定應用程式來源，確保應用程式在外部容器中正確引導。

以下是指南中的代碼：

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

@SpringBootApplication
public class DemoApplication extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(DemoApplication.class);
    }

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

2. 打包為 WAR 文件：

- pom.xml 指定 `<packaging>war</packaging>` 並將 `spring-boot-starter-tomcat` 依賴項標記為 `<scope>provided</scope>`，以排除嵌入式 Tomcat 伺服器，改為依賴 WLP 的 Servlet 容器。

3. 部署到 WLP：

- WAR 文件放置在 WLP 的 `dropins` 目錄中以進行自動部署，並且 WLP 的 `javaee-8.0` 功能提供 Servlet 4.0 支持，這與 Spring Boot 的要求兼容。
-

為什麼在 Spring Boot 2.7.18 中不需要更改

Spring Boot 2.7.18 是 2.7.x 系列的一部分，部署機制或 API 的重大變更通常發生在主要版本之間（例如 2.x 到 3.x），而不是在次要或補丁版本中。以下是詳細分析：

1. 兼容性 `SpringBootServletInitializer`

- **用途：** `SpringBootServletInitializer` 仍然是 2.x 系列中配置 Spring Boot 應用程式進行 WAR 部署的標準方法。它通過提供設置應用程式上下文的鉤子來與外部 Servlet 容器集成。
- **2.7.18 中的穩定性：** Spring Boot 2.7.18 中沒有 `SpringBootServletInitializer` 的廢棄或替代品。重大變更，例如轉向 Jakarta EE（取代 Java EE API），發生在 Spring Boot 3.x 中，這也需要 Java 17。由於 2.7.18 仍在 2.x 系列內並使用 Java EE，`DemoApplication` 中的現有實現仍然有效且未變。

2. Servlet 容器兼容性

- **Spring Boot 要求：** Spring Boot 2.7.x 需要 Servlet 3.1 或更高版本。指南使用 WLP 並啟用 `javaee-8.0` 功能，這包括 Servlet 4.0——一個更新的規範。這確保了完全兼容。
- **2.7.18 中沒有變化：** 如 2.7.18 這樣的補丁版本不會改變 Servlet 兼容性或引入新需求，這會影響 `SpringBootServletInitializer` 如何與 WLP 互動。

3. 依賴和打包配置

- **Tomcat 作為 provided：** 指南在 `pom.xml` 中正確地將 `spring-boot-starter-tomcat` 設置為 `<scope>provided</scope>`，確保嵌入式 Tomcat 從 WAR 文件中排除。這是外部容器 WAR 部署的標準做法，並在 2.7.18 中保持不變。
- **Maven 配置：** 打包類型 (`war`) 和依賴項設置與 Spring Boot 2.7.x 規範一致，並且不需要特定於 2.7.18 的更新。

4. WLP 部署特定

- **dropins 目錄：**WLP 的 dropins 部署機制不受 Spring Boot 版本的影響，因為它依賴於 WAR 文件結構和 Servlet 規範，這些都保持兼容。
- **上下文根和端口：**上下文根（例如 /myapp）和默認端口（例如 9080）由 WLP 和 WAR 文件名決定，而不是 Spring Boot 的版本，因此保持不變。

5. 可能的新功能或廢棄

- **釋放說明考慮：**Spring Boot 2.7.x 釋放說明（包括 2.7.18）通常涵蓋依賴項更新、安全修復和次要增強，但沒有影響 WAR 部署或 `SpringBootServletInitializer`。例如，2.7.18 中沒有引入新的註釋或配置方法來取代現有方法。
 - **與 3.x 對比：**相反，Spring Boot 3.x 採用 Jakarta EE 並可能需要調整（例如不同的基礎類或更新的 Servlet API），但 2.7.18 不會過渡到這些變更。
-

結論

對於 Spring Boot 2.7.18，指南中所示的 `DemoApplication` 類別和使用 `SpringBootServletInitializer` 不需要任何修改。配置完全兼容：

- WAR 打包和部署到 WLP。
- WLP 中的 javaee-8.0 功能（Servlet 4.0）。
- Spring Boot 2.7.x 的 Servlet 和部署需求。

您可以按照指南中的確切步驟和代碼進行操作——使用 Spring Boot 2.7.18 而不是通用的 2.7.x 版本——並期望應用程式在 WLP 上成功部署和運行，並在 `http://localhost:9080/myapp/` 返回 “Hello World!”。不需要額外的配置、屬性更改或代碼調整。