

Spring Boot 2.7.18 en WebSphere Liberty Profile

Al usar Spring Boot 2.7.18 para desplegar una aplicación en WebSphere Liberty Profile (WLP) según se describe en la guía proporcionada, la buena noticia es que no se requieren cambios en la clase `DemoApplication` ni en el uso de `SpringBootServletInitializer`. Spring Boot 2.7.18 es una versión de parche dentro de la serie 2.7.x, lo que significa que incluye correcciones de errores y mejoras menores, pero no introduce cambios disruptivos que afecten la configuración estándar para el despliegue de WAR descrita en la guía. A continuación, explicaré por qué es así y confirmaré que la configuración sigue siendo completamente compatible.

Entendiendo la Configuración

La guía configura una aplicación Spring Boot para ser desplegada como un archivo WAR en WLP mediante:

1. **Extender** `SpringBootServletInitializer`:

- La clase `DemoApplication` extiende `SpringBootServletInitializer` para habilitar el despliegue en un contenedor de servlets externo como WLP. Esta clase es necesaria porque, aunque Spring Boot predetermina ejecutarse como un JAR independiente con un servidor incrustado, el despliegue en WLP requiere integración con su contenedor de servlets.
- El método `configure` se sobrescribe para especificar las fuentes de la aplicación, asegurando que la aplicación se inicie correctamente en el contenedor externo.

Aquí está el código de la guía:

```
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. Empaquetar como un Archivo WAR:

- El `pom.xml` especifica `<packaging>war</packaging>` y marca la dependencia `spring-boot-starter-tomcat` como `<scope>provided</scope>` para excluir el servidor Tomcat incrustado, confiando en el contenedor de servlets de WLP en su lugar.

3. Desplegar en WLP:

- El archivo WAR se coloca en el directorio `dropins` de WLP para el despliegue automático, y la característica `javaee-8.0` de WLP proporciona soporte para Servlet 4.0, que es compatible con los requisitos de Spring Boot.

Por Qué No Se Necesitan Cambios con Spring Boot 2.7.18

Spring Boot 2.7.18 es parte de la serie 2.7.x, y los cambios significativos en los mecanismos de despliegue o APIs generalmente ocurren entre versiones principales (por ejemplo, 2.x a 3.x), no dentro de versiones menores o de parche. Aquí hay un análisis detallado:

1. Compatibilidad con `SpringBootServletInitializer`

- **Propósito:** `SpringBootServletInitializer` sigue siendo la forma estándar de configurar una aplicación Spring Boot para el despliegue de WAR en la serie 2.x. Se integra con el contenedor de servlets externo proporcionando un gancho para configurar el contexto de la aplicación.
- **Estabilidad en 2.7.18:** No hay depreciaciones ni reemplazos para `SpringBootServletInitializer` en Spring Boot 2.7.18. Los cambios importantes, como el cambio a Jakarta EE (reemplazando las APIs de Java EE), ocurren en Spring Boot 3.x, que también requiere Java 17. Dado que 2.7.18 se mantiene dentro de la serie 2.x y utiliza Java EE, la implementación existente en `DemoApplication` sigue siendo válida y sin cambios.

2. Compatibilidad del Contenedor de Servlets

- **Requisitos de Spring Boot:** Spring Boot 2.7.x requiere Servlet 3.1 o superior. La guía utiliza WLP con la característica `javaee-8.0`, que incluye Servlet 4.0, una especificación aún más nueva. Esto asegura la compatibilidad total.
- **Sin Cambios en 2.7.18:** Las versiones de parche como 2.7.18 no alteran la compatibilidad de los servlets ni introducen nuevos requisitos que afecten cómo `SpringBootServletInitializer` interactúa con WLP.

3. Configuración de Dependencias y Empaquetado

- **Tomcat como provided:** La guía establece correctamente `spring-boot-starter-tomcat` como `<scope>provided</scope>` en `pom.xml`, asegurando que el Tomcat incrustado se excluya del archivo WAR. Esto es una práctica estándar para los despliegues de WAR en contenedores externos y sigue sin cambios en 2.7.18.
- **Configuración de Maven:** El tipo de empaquetado (`war`) y la configuración de dependencias son consistentes con las convenciones de Spring Boot 2.7.x, y no se necesitan actualizaciones específicas para 2.7.18.

4. Especificidades de Despliegue en WLP

- **Directorio dropins:** El mecanismo de despliegue `dropins` de WLP no se ve afectado por la versión de Spring Boot, ya que depende de la estructura del archivo WAR y las especificaciones de servlets, ambas compatibles.
- **Raíz del Contexto y Puerto:** La raíz del contexto (por ejemplo, `/myapp`) y el puerto predeterminado (por ejemplo, `9080`) se determinan por WLP y el nombre del archivo WAR, no por la versión de Spring Boot, por lo que permanecen iguales.

5. Nuevas Características o Depreciaciones Potenciales

- **Consideración de Notas de la Versión:** Las notas de la versión de Spring Boot 2.7.x (incluyendo 2.7.18) generalmente cubren actualizaciones de dependencias, correcciones de seguridad y mejoras menores, pero ninguna afecta el despliegue de WAR ni `SpringBootServletInitializer`. Por ejemplo, no se introdujeron nuevas anotaciones ni métodos de configuración en 2.7.18 para reemplazar el enfoque existente.
- **Contraste con 3.x:** En contraste, Spring Boot 3.x adopta Jakarta EE y podría requerir ajustes (por ejemplo, diferentes clases base o APIs de servlets actualizadas), pero 2.7.18 no aborda esos cambios.

Conclusión

Para Spring Boot 2.7.18, la clase `DemoApplication` y el uso de `SpringBootServletInitializer` tal como se muestra en la guía no requieren ninguna modificación. La configuración es completamente compatible con:

- El empaquetado y despliegue de WAR en WLP.
- La característica `javaee-8.0` en WLP (Servlet 4.0).
- Los requisitos de servlet y despliegue de Spring Boot 2.7.x.

Puede proceder con los pasos y el código exactos de la guía, utilizando Spring Boot 2.7.18 en lugar de una versión genérica 2.7.x, y esperar que la aplicación se despliegue y ejecute correctamente en WLP, devolviendo “Hello World!” en `http://localhost:9080/myapp/`. No se necesitan configuraciones adicionales, cambios de propiedades ni ajustes de código.