

# Spring Boot 2.7.18 dans WebSphere Liberty Profile

Lors de l'utilisation de Spring Boot 2.7.18 pour déployer une application sur WebSphere Liberty Profile (WLP) comme décrit dans le guide fourni, la bonne nouvelle est qu'aucun changement n'est nécessaire pour la classe `DemoApplication` ou l'utilisation de `SpringBootServletInitializer`. Spring Boot 2.7.18 est une version corrective dans la série 2.7.x, ce qui signifie qu'elle inclut des correctifs de bugs et des améliorations mineures mais n'introduit pas de changements cassants qui affecteraient la configuration standard pour le déploiement WAR décrite dans le guide. Ci-dessous, je vais expliquer pourquoi c'est le cas et confirmer que la configuration reste entièrement compatible.

---

## Compréhension de la Configuration

Le guide configure une application Spring Boot pour être déployée en tant que fichier WAR sur WLP en :

### 1. **Étendant** `SpringBootServletInitializer` :

- La classe `DemoApplication` étend `SpringBootServletInitializer` pour permettre le déploiement sur un conteneur de servlets externe comme WLP. Cette classe est nécessaire car, bien que Spring Boot par défaut s'exécute en tant que JAR autonome avec un serveur intégré, le déploiement sur WLP nécessite une intégration avec son conteneur de servlets.
- La méthode `configure` est remplacée pour spécifier les sources de l'application, garantissant que l'application démarre correctement dans le conteneur externe.

Voici le code du guide :

```
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. Emballage en tant que fichier WAR :

- Le `pom.xml` spécifie `<packaging>war</packaging>` et marque la dépendance `spring-boot-starter-tomcat` comme `<scope>provided</scope>` pour exclure le serveur Tomcat intégré, en s'appuyant à la place sur le conteneur de servlets de WLP.

## 3. Déploiement sur WLP :

- Le fichier WAR est placé dans le répertoire `dropins` de WLP pour un déploiement automatique, et la fonctionnalité `javaee-8.0` de WLP fournit une prise en charge de Servlet 4.0, qui est compatible avec les exigences de Spring Boot.

---

## Pourquoi Aucun Changement N'est Nécessaire avec Spring Boot 2.7.18

Spring Boot 2.7.18 fait partie de la série 2.7.x, et des changements significatifs aux mécanismes de déploiement ou aux API se produisent généralement entre les versions majeures (par exemple, 2.x à 3.x), et non dans les versions mineures ou correctives. Voici une analyse détaillée :

### 1. Compatibilité avec `SpringBootServletInitializer`

- **But :** `SpringBootServletInitializer` reste la méthode standard pour configurer une application Spring Boot pour un déploiement WAR dans la série 2.x. Il s'intègre avec le conteneur de servlets externe en fournissant un point d'accroche pour configurer le contexte de l'application.
- **Stabilité dans 2.7.18 :** Il n'y a pas de dépréciations ou de remplacements pour `SpringBootServletInitializer` dans Spring Boot 2.7.18. Les changements majeurs, tels que le passage à Jakarta EE (remplaçant les API Java EE), se produisent dans Spring Boot 3.x, qui nécessite également Java 17. Puisque 2.7.18 reste dans la série 2.x et utilise Java EE, l'implémentation existante dans `DemoApplication` reste valide et inchangée.

### 2. Compatibilité du Conteneur de Servlets

- **Exigences de Spring Boot :** Spring Boot 2.7.x nécessite Servlet 3.1 ou supérieur. Le guide utilise WLP avec la fonctionnalité `javaee-8.0`, qui inclut Servlet 4.0—une spécification encore plus récente. Cela garantit une compatibilité totale.
- **Pas de Changements dans 2.7.18 :** Les versions correctives comme 2.7.18 n'altèrent pas la compatibilité des servlets ou n'introduisent pas de nouvelles exigences qui affecteraient la manière dont `SpringBootServletInitializer` interagit avec WLP.

### 3. Configuration des Dépendances et de l'Emballage

- **Tomcat en tant que provided** : Le guide définit correctement `spring-boot-starter-tomcat` à `<scope>provided</scope>` dans `pom.xml`, garantissant que le Tomcat intégré est exclu du fichier WAR. Il s'agit d'une pratique standard pour les déploiements WAR vers des conteneurs externes et reste inchangée dans 2.7.18.
- **Configuration Maven** : Le type d'emballage (`war`) et la configuration des dépendances sont cohérents avec les conventions de Spring Boot 2.7.x, et aucune mise à jour spécifique à 2.7.18 n'est nécessaire.

### 4. Spécificités du Déploiement sur WLP

- **Répertoire Dropins** : Le mécanisme de déploiement `dropins` de WLP n'est pas affecté par la version de Spring Boot, car il repose sur la structure du fichier WAR et les spécifications des servlets, toutes deux restant compatibles.
- **Racine du Contexte et Port** : La racine du contexte (par exemple, `/myapp`) et le port par défaut (par exemple, `9080`) sont déterminés par WLP et le nom du fichier WAR, et non par la version de Spring Boot, donc ils restent les mêmes.

### 5. Fonctionnalités ou Dépréciations Potentielles

- **Considération des Notes de Version** : Les notes de version de Spring Boot 2.7.x (y compris 2.7.18) couvrent généralement les mises à jour des dépendances, les correctifs de sécurité et les améliorations mineures, mais aucune n'impacte le déploiement WAR ou `SpringBootServletInitializer`. Par exemple, aucune nouvelle annotation ou méthode de configuration n'a été introduite dans 2.7.18 pour remplacer l'approche existante.
- **Contraste avec 3.x** : En contraste, Spring Boot 3.x adopte Jakarta EE et pourrait nécessiter des ajustements (par exemple, différentes classes de base ou API de servlets mises à jour), mais 2.7.18 ne fait pas la transition vers ces changements.

---

### Conclusion

Pour Spring Boot 2.7.18, la classe `DemoApplication` et l'utilisation de `SpringBootServletInitializer` telles que montrées dans le guide ne nécessitent aucune modification. La configuration est entièrement compatible avec :

- L'emballage et le déploiement WAR vers WLP.
- La fonctionnalité `javaee-8.0` dans WLP (Servlet 4.0).
- Les exigences de servlet et de déploiement de Spring Boot 2.7.x.

Vous pouvez procéder avec les étapes et le code exacts du guide—en utilisant Spring Boot 2.7.18 au lieu d’une version générique 2.7.x—et vous attendre à ce que l’application se déploie et s’exécute avec succès sur WLP, retournant “Hello World!” à <http://localhost:9080/myapp/>. Aucune configuration supplémentaire, modification de propriété ou ajustement de code n’est nécessaire.