

# Servlet-Version in WebSphere Liberty Profile

## Fehler verstehen

Die Fehlermeldung von `console.log` weist auf eine `DDParser ParseException` hin, die mit der Datei `web-fragment.xml` im `spring-web-6.2.3.jar` verbunden ist. Sie gibt an, dass der Bereitstellungsdeskriptor die Version "50" (die Servlet 5.0 entspricht) angibt, während der WebSphere Liberty Profile (WLP)-Server derzeit mit der Version "40" (Servlet 4.0) bereitgestellt ist. Dieser Unterschied tritt auf, weil:

- **Servlet 5.0** Teil von Jakarta EE 9 ist und von neueren Versionen von Spring Boot (z.B. Spring Boot 3.x, das `spring-web-6.2.3.jar` enthält) benötigt wird.
- **Servlet 4.0** Teil von Java EE 8 ist und WLP wahrscheinlich mit dem `javaee-8.0`-Feature konfiguriert ist, das Servlet 5.0 nicht unterstützt.

Um dies zu beheben, müssen Sie die von WLP unterstützte Servlet-Version mit der von Ihrer Anwendung benötigten Version abgleichen. Die empfohlene Lösung besteht darin, WLP zu aktualisieren, um Servlet 5.0 zu unterstützen, indem das `jakartaee-9.1`-Feature aktiviert wird.

---

## Lösung: WLP aktualisieren, um Servlet 5.0 zu unterstützen

Hier ist, wie Sie das Problem beheben können, indem Sie WLP so aktualisieren, dass es das `jakartaee-9.1`-Feature verwendet, das die Unterstützung für Servlet 5.0 enthält:

### 1. Datei `server.xml` finden

- Die `server.xml`-Konfigurationsdatei definiert die in WLP aktivierten Features.
- Sie befindet sich normalerweise im Serververzeichnis, z.B. `/opt/ibm/wlp/usr/servers/myServer/server.xml`, wobei `myServer` der Name Ihres Servers ist.

### 2. Datei `server.xml` bearbeiten

- Öffnen Sie die `server.xml`-Datei in einem Texteditor.
- Suchen Sie den Abschnitt `<featureManager>`, der die für den Server aktivierten Features auflistet. Er könnte derzeit so aussehen:

```
<featureManager>
  <feature>javaee-8.0</feature>
</featureManager>
```

- Ersetzen Sie das `javaee-8.0`-Feature durch `jakartaee-9.1`, um die Unterstützung für Servlet 5.0 zu aktivieren:

```
<featureManager>
  <feature>jakartaee-9.1</feature>
</featureManager>
```

- Speichern Sie die Datei.

### 3. Änderungen in WLP Entwicklungsmodus anwenden (falls zutreffend)

- Wenn Sie WLP im **Entwicklungsmodus** (z.B. mit dem Befehl `wlp/bin/server run` oder einer IDE-Integration) ausführen, können Sie die Änderungen wie folgt anwenden:

#### - Manuelles Neustarten:

- \* Stoppen Sie den Server:

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

- \* Starten Sie den Server erneut:

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

#### - Entwicklungsmodus Hot Reload:

- \* Wenn WLP im Entwicklungsmodus (z.B. über `server run` oder eine IDE) läuft, erkennt es möglicherweise Änderungen an `server.xml` automatisch. Um sicherzustellen, dass das neue Feature geladen wird, wird jedoch ein Neustart empfohlen.

### 4. Fix überprüfen

- Nach dem Neustarten des Servers setzen Sie Ihre Anwendung erneut ein (z.B. durch Kopieren der WAR-Datei in das `dropins`-Verzeichnis oder durch Verwendung Ihres Bereitstellungsverfahrens).
- Überprüfen Sie die Serverprotokolle auf eine Bestätigung der erfolgreichen Bereitstellung. Suchen Sie nach Nachrichten wie:

```
[AUDIT ] CWWKT0016I: Webanwendung verfügbar (default_host): http://localhost:9080/myapp/
```

```
CWWKZ0001I: Anwendung myapp gestartet in X.XXX Sekunden.
```

- Greifen Sie auf Ihre Anwendung zu (z.B. `http://localhost:9080/myapp/`), um zu bestätigen, dass sie ohne Fehler läuft.

## Alternative Lösung: Spring Boot herunterstufen

Wenn das Aktualisieren von WLP auf `jakartaee-9.1` keine Option ist (z.B. aufgrund von Kompatibilitätsbeschränkungen mit anderen Anwendungen), können Sie Ihre Spring Boot-Anwendung auf eine Version herunterstufen, die mit Servlet 4.0 kompatibel ist:

### 1. Auf Spring Boot 2.7.x umstellen

- Spring Boot 2.7.x unterstützt Servlet 4.0 (und frühere Versionen wie Servlet 3.1).
- Aktualisieren Sie Ihre `pom.xml`, um Spring Boot 2.7.x zu verwenden:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.7.18</version> <!-- Verwenden Sie die neueste 2.7.x-Version -->
  <relativePath/>
</parent>
```

### 2. Neu erstellen und neu bereitstellen

- Erstellen Sie Ihre Anwendung neu:

```
mvn clean package
```

- Stellen Sie die generierte WAR-Datei in WLP neu bereit und überprüfen Sie, ob sie ohne Fehler startet.

---

## Zusammenfassung

- **Ursache des Fehlers:** Ihre Anwendung (`spring-web-6.2.3.jar`) benötigt Servlet 5.0, aber WLP ist für Servlet 4.0 konfiguriert.
- **Bevorzugte Lösung:** Aktualisieren Sie WLP, um das `jakartaee-9.1`-Feature zu verwenden, indem Sie `server.xml` bearbeiten und den Server neu starten.
- **Alternative Lösung:** Herunterstufen auf Spring Boot 2.7.x, um die Unterstützung für Servlet 4.0 von WLP zu entsprechen.

Die bevorzugte Lösung (Aktualisieren von WLP) stellt sicher, dass Ihre Anwendung die neuesten Funktionen von Spring Boot 3.x und Jakarta EE 9 nutzen kann. Folgen Sie den obigen Schritten, um das Problem zu beheben und Ihre Anwendung erfolgreich bereitzustellen.