

# EAR Deployment im WebSphere Liberty Profil

Es scheint, dass Sie erfolgreich eine EAR-Datei (Enterprise Archive) mit `mvn package` und Maven erstellt haben und der Build-Prozess ohne Fehler abgeschlossen wurde. Wenn Sie diese EAR jedoch auf WebSphere Liberty Profile (WLP) bereitstellen und den Server starten, erhalten Sie die Fehlermeldung "kein Modul gefunden." Dies ist verwirrend, da ein reibungsloser Build normalerweise darauf hinweist, dass alles in Ordnung ist, der Server Ihre Anwendung jedoch nicht wie erwartet erkennt. Lassen Sie uns dies analysieren und herausfinden, was passiert sein könnte, sowie Schritte zur Behebung des Problems.

## Verständnis des Problems

Eine EAR-Datei ist ein Standard-Java-EE-Archiv, das Ihre Anwendung enthält und normalerweise ein oder mehrere Module wie WAR-Dateien (für Webanwendungen), JAR-Dateien (für EJBs oder Bibliotheken) und einen Bereitstellungsdeskriptor (`application.xml`) enthält. Wenn Sie eine EAR auf WLP bereitstellen, sollte der Server diese Module erkennen und die Anwendung starten. Die Meldung "kein Modul gefunden" deutet darauf hin, dass WLP entweder keine Module innerhalb Ihrer EAR findet oder sie nicht erkennt, was verhindert, dass die Anwendung gestartet wird.

Da Ihr Maven-Build erfolgreich war ("alles läuft reibungslos"), liegt das Problem wahrscheinlich in einem der drei Bereiche: 1. **Der Inhalt der EAR-Datei:** Die EAR könnte die erwarteten Module nicht enthalten, oder der Bereitstellungsdeskriptor könnte fehlen oder falsch sein. 2. **Der Bereitstellungsprozess:** Die Art und Weise, wie Sie die EAR auf WLP bereitstellen, könnte nicht mit der erwarteten Methode des Servers übereinstimmen, um sie zu finden und zu verarbeiten. 3. **Die Serverkonfiguration:** WLP könnte möglicherweise nicht so konfiguriert sein, dass er die Module in Ihrer EAR erkennt, aufgrund fehlender Funktionen oder falscher Konfiguration.

Lassen Sie uns diese Möglichkeiten untersuchen und handlungsfähige Schritte zur Diagnose und Behebung des Problems anbieten.

---

## Mögliche Ursachen und Lösungen

**1. Die EAR-Datei könnte leer oder falsch verpackt sein** Auch wenn der Build erfolgreich war, ist es möglich, dass Ihre EAR keine Module (z.B. WAR- oder JAR-Dateien) enthält oder dass die `application.xml`-Datei, die dem Server mitteilt, welche Module enthalten sind, fehlt oder falsch konfiguriert ist.

- **Warum dies passiert:** In einem Maven-EAR-Projekt ist das `maven-ear-plugin` dafür verantwortlich, die EAR zusammenzustellen. Es fügt Module basierend auf Ihrer `pom.xml`-Konfiguration oder Abhängigkeiten hinzu. Wenn keine Module angegeben sind oder Abhängigkeiten (wie eine WAR) nicht korrekt definiert oder aufgelöst sind, könnte die EAR leer sein oder eine korrekte `application.xml` fehlen.

- **Wie man überprüft:**

- Öffnen Sie Ihre EAR-Datei (es handelt sich um ein ZIP-Archiv) mit einem Tool wie `unzip` oder führen Sie `jar tf myApp.ear` im Terminal aus, um deren Inhalt aufzulisten.
- Achten Sie auf:
  - \* Moduldateien (z.B. `my-web.war`, `my-ejb.jar`) an der Wurzel der EAR.
  - \* Eine `META-INF/application.xml`-Datei.
- In `application.xml` sollten Einträge zu sehen sein, die Ihre Module definieren, wie:

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <module>
    <web>
      <web-uri>my-web.war</web-uri>
      <context-root>/myapp</context-root>
    </web>
  </module>
</application>
```

- **Wie man es behebt:**

- Überprüfen Sie Ihr `pom.xml` für das EAR-Modul. Stellen Sie sicher, dass Sie Abhängigkeiten für die Module, die Sie enthalten möchten, angegeben haben, z.B.:

```
<dependencies>
  <dependency>
    <groupId>com.example</groupId>
    <artifactId>my-web</artifactId>
    <type>war</type>
  </dependency>
</dependencies>
```

- Konfigurieren Sie das `maven-ear-plugin` bei Bedarf:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-ear-plugin</artifactId>
  <version>3.3.0</version>
  <configuration>
    <modules>
      <webModule>
        <groupId>com.example</groupId>
        <artifactId>my-web</artifactId>
        <contextRoot>/myapp</contextRoot>
      </webModule>
```

```
        </modules>
    </configuration>
</plugin>
```

- Neu bauen mit `mvn clean package` und erneut die EAR-Inhalte überprüfen.

Wenn die EAR leer ist oder `application.xml` fehlt/fehlerhaft ist, ist dies wahrscheinlich die Ursache. Die Behebung der Maven-Konfiguration sollte dies beheben.

---

**2. Problem mit dem Bereitstellungsverfahren** Wie Sie die EAR auf WLP bereitstellen, könnte ebenfalls das Problem sein. WLP unterstützt zwei Hauptbereitstellungsmethoden: das `dropins`-Verzeichnis und die explizite Konfiguration in `server.xml`.

- **Verwendung des `dropins`-Verzeichnisses:**

- Wenn Sie die EAR im Verzeichnis `wlp/usr/servers/<serverName>/dropins/` platziert haben, sollte WLP sie automatisch erkennen und bereitstellen.
- Für EAR-Dateien funktioniert die automatische Bereitstellung jedoch möglicherweise nicht immer wie erwartet, insbesondere wenn zusätzliche Konfigurationen (wie Kontextwurzeln) erforderlich sind.

- **Verwendung von `server.xml`:**

- Für EAR-Dateien ist es oft besser, die Anwendung explizit in `wlp/usr/servers/<serverName>/server.xml` zu konfigurieren:

```
<server>
    <featureManager>
        <feature>servlet-3.1</feature> <!-- Stellen Sie sicher, dass die erforderlichen Funktionen al
    </featureManager>
    <application id="myApp" name="myApp" type="ear" location="${server.config.dir}/apps/myApp.ear"/>
</server>
```

- Platzieren Sie die EAR in `wlp/usr/servers/<serverName>/apps/` (oder passen Sie den `location`-Pfad entsprechend an).

- **Wie man überprüft:**

- Bestätigen Sie, wo Sie die EAR platziert haben und wie Sie den Server starten (z.B. `./bin/server run <serverName>`).
- Überprüfen Sie die Serverprotokolle (z.B. `wlp/usr/servers/<serverName>/logs/console.log` oder `messages.log`) auf Bereitstellungsmeldungen.

- **Wie man es behebt:**

- Versuchen Sie, die EAR in `server.xml` wie oben gezeigt zu konfigurieren, anstatt auf `dropins` zu setzen.
- Starten Sie den Server nach dem Vornehmen von Änderungen neu: `./bin/server stop <serverName>` gefolgt von `./bin/server start <serverName>`.

Wenn die EAR nicht ordnungsgemäß beim Server registriert wurde, könnte dies die Fehlermeldung erklären.

---

**3. Fehlende Server-Funktionen** WLP ist ein leichtgewichtiger Server, der nur die Funktionen lädt, die Sie in `server.xml` aktivieren. Wenn Ihre EAR Module enthält, die spezifische Funktionen (z.B. Servlets, EJBs) erfordern, und diese Funktionen nicht aktiviert sind, könnte WLP die Module möglicherweise nicht erkennen oder laden.

- **Warum dies passiert:** Zum Beispiel benötigt eine WAR-Datei die Funktion `servlet-3.1` (oder höher), während ein EJB-Modul `ejbLite-3.2` benötigt. Ohne diese Funktionen könnte der Server möglicherweise die Module nicht verarbeiten.
- **Wie man überprüft:**
  - Sehen Sie sich Ihre `server.xml` an und überprüfen Sie den Abschnitt `<featureManager>`.
  - Gängige Funktionen umfassen:
    - \* `<feature>servlet-3.1</feature>` für Webmodule.
    - \* `<feature>ejbLite-3.2</feature>` für EJB-Module.
  - Überprüfen Sie die Serverprotokolle auf Meldungen zu fehlenden Funktionen (z.B. "erforderliche Funktion ist nicht installiert").
- **Wie man es behebt:**
  - Fügen Sie die erforderlichen Funktionen in `server.xml` basierend auf den Anforderungen Ihrer Anwendung hinzu:

```
<featureManager>
  <feature>servlet-3.1</feature>
  <!-- Fügen Sie andere Funktionen nach Bedarf hinzu -->
</featureManager>
```
  - Starten Sie den Server neu, um die Änderungen zu übernehmen.

Wenn die Funktionen fehlen, sollten diese aktiviert werden, damit WLP die Module erkennen kann.

---

## Diagnose-Schritte

Um das Problem zu lokalisieren, befolgen Sie diese Schritte:

### 1. Überprüfen Sie die EAR-Datei:

- Führen Sie `jar tf myApp.ear` aus oder entpacken Sie sie.
- Stellen Sie sicher, dass sie Ihre Module (z.B. `my-web.war`) und eine gültige `META-INF/application.xml` enthält.

### 2. Überprüfen Sie den Maven-Build:

- Überprüfen Sie das `pom.xml` Ihres EAR-Moduls, um Abhängigkeiten und die Konfiguration des `maven-ear-plugin` zu bestätigen.
- Führen Sie `mvn clean package` erneut aus und überprüfen Sie die Build-Ausgabe auf Meldungen zum Hinzufügen von Modulen (z.B. "Modul `my-web.war` hinzufügen").

### 3. Überprüfen Sie die Bereitstellung:

- Bestätigen Sie den Speicherort der EAR (z.B. `dropins` oder `apps`).
- Wenn Sie `dropins` verwenden, versuchen Sie, es in `apps` zu verschieben und es zu `server.xml` hinzuzufügen.

### 4. Überprüfen Sie die Serverprotokolle:

- Starten Sie den Server und überprüfen Sie `console.log` oder `messages.log` auf detaillierte Fehlermeldungen über "kein Modul gefunden."
- Achten Sie auf Hinweise wie "Anwendung enthält keine Module" oder fehlerbezogene Meldungen.

### 5. Testen Sie mit einer einfachen EAR:

- Erstellen Sie eine grundlegende EAR mit einer einzelnen WAR-Datei unter Verwendung eines bekannten funktionierenden `pom.xml` und stellen Sie sie auf WLP bereit. Wenn es funktioniert, ist das Problem spezifisch für die Verpackung Ihrer ursprünglichen EAR.

---

## Wahrscheinlichste Ursache

Da Ihr Build reibungslos verlief, WLP jedoch "kein Modul gefunden" meldet, ist die wahrscheinlichste Ursache, dass **die EAR-Datei keine Module enthält** oder **die `application.xml` fehlt/fehlerhaft ist**. Dies könnte passieren, wenn Ihr `pom.xml` die zu enthaltenden Module nicht korrekt angibt, auch wenn Maven keinen Fehler meldet. Alternativ könnte, wenn die EAR korrekt ist, das Problem mit der Bereitstellung (z.B. nicht in `server.xml` konfiguriert) oder fehlenden Serverfunktionen liegen.

## Endgültige Empfehlung

- Beginnen Sie mit der Überprüfung der EAR-Inhalte (`jar tf myApp.ear`) und Ihrer `pom.xml`-Konfiguration.
- Wenn die EAR leer ist oder `application.xml` fehlt, beheben Sie die Maven-Einrichtung und bauen Sie neu.
- Wenn die EAR korrekt aussieht, konfigurieren Sie sie in `server.xml`, stellen Sie sicher, dass die erforderlichen Funktionen aktiviert sind, und stellen Sie erneut bereit.
- Überprüfen Sie die Serverprotokolle auf zusätzlichen Kontext.

Durch die Befolgung dieser Schritte sollten Sie in der Lage sein, zu ermitteln, ob das Problem im Build, in der Bereitstellung oder in der Serverkonfiguration liegt und Ihre Anwendung erfolgreich auf WLP auszuführen. Lassen Sie mich wissen, wenn Sie Hilfe bei spezifischen Konfigurationen benötigen!