

# WebSphere Liberty - Hallo Welt Anwendung

Hier ist ein einfaches Beispiel, um eine "Hello World"-Anwendung auf Ihrem WebSphere Liberty-Server auszuführen, vorausgesetzt, Sie haben ihn bereits mit dem javaee-8.0-Feature gemäß den vorherigen Anweisungen eingerichtet. Dieses Beispiel verwendet eine grundlegende JSP-Datei (JavaServer Pages) und stellt sie mithilfe des `dropins`-Verzeichnisses für die Einfachheit zur Verfügung.

## Schritt-für-Schritt-Anleitung

**1. Erstellen Sie das Anwendungsverzeichnis und die Datei** Erstellen Sie eine kleine Webanwendung, indem Sie eine JSP-Datei im `dropins`-Verzeichnis Ihres Liberty-Servers platzieren. Das `dropins`-Verzeichnis ermöglicht es Liberty, Anwendungen automatisch zu erkennen und zu bereitstellen.

- **Lokalisieren Sie das `dropins`-Verzeichnis:** Navigieren Sie zum `dropins`-Verzeichnis innerhalb des Ordners Ihres Servers. Wenn Ihre Liberty-Installation sich unter `/opt/ibm/wlp` befindet und Ihr Server `myServer` heißt, lautet der Pfad:

```
/opt/ibm/wlp/usr/servers/myServer/dropins
```

Ersetzen Sie `/opt/ibm/wlp` durch Ihr tatsächliches Liberty-Installationsverzeichnis und `myServer` durch den Namen Ihres Servers.

- **Erstellen Sie ein entpacktes WAR-Verzeichnis:** Erstellen Sie im `dropins`-Verzeichnis einen Ordner namens `myApp.war`. Die Erweiterung `.war` weist Liberty an, dies als Webanwendung zu behandeln. Verwenden Sie dazu diesen Befehl:

```
mkdir -p /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war
```

- **Erstellen Sie die `index.jsp`-Datei:** Erstellen Sie innerhalb von `myApp.war` eine Datei namens `index.jsp` mit folgendem Inhalt, um "Hello World!" anzuzeigen:

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

Sie können sie direkt mit einem Befehl wie diesem erstellen:

```
echo '<html><body><h2>Hello World!</h2></body></html>' > /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war
```

Alternativ können Sie einen Texteditor verwenden, um `index.jsp` zu erstellen und sie an diesem Ort zu speichern.

**2. Starten Sie den Server (falls er noch nicht läuft)** Wenn Ihr Server nicht läuft, müssen Sie ihn starten, damit er die Anwendung bereitstellen und bereitstellen kann.

- **Navigieren Sie zum bin-Verzeichnis:** Gehen Sie zum bin-Verzeichnis in Ihrer Liberty-Installation:

```
cd /opt/ibm/wlp/bin
```

- **Starten Sie den Server:** Führen Sie den Server im Vordergrundmodus aus, um die Ausgabe direkt zu sehen:

```
./server run myServer
```

Alternativ starten Sie ihn im Hintergrund:

```
./server start myServer
```

Wenn der Server bereits läuft, überspringen Sie diesen Schritt –Liberty wird die neue Anwendung automatisch erkennen.

**3. Überprüfen Sie die Anwendungsbereitstellung** Liberty stellt die `myApp.war`-Anwendung automatisch bereit, wenn es sie im `dropins`-Verzeichnis erkennt.

- **Überprüfen Sie die Konsolenausgabe:** Wenn Sie den Server im Vordergrundmodus gestartet haben, suchen Sie nach einer Nachricht wie:

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

Dies bestätigt, dass die Anwendung bereitgestellt und verfügbar ist.

- **Überprüfen Sie die Protokolle (wenn im Hintergrund ausgeführt):** Wenn der Server im Hintergrund läuft, suchen Sie in der Protokolldatei unter:

```
/opt/ibm/wlp/usr/servers/myServer/logs/console.log
```

nach der gleichen Bereitstellungsrichtung.

**4. Greifen Sie auf die Anwendung zu** Nun, da die Anwendung bereitgestellt ist, können Sie sie in einem Webbrowser anzeigen.

- **Öffnen Sie Ihren Browser:** Navigieren Sie zu:

```
http://localhost:9080/myApp/
```

Der Kontextstamm `/myApp` stammt aus dem Namen des `myApp.war`-Verzeichnisses, und `index.jsp` wird standardmäßig als Willkommensdatei bereitgestellt.

- **Erwartetes Ergebnis:** Sie sollten eine Seite sehen, die anzeigt:

Hello World!

in großer Schrift.

## Hinweise

- **Portnummer:** Dieses Beispiel verwendet den Port 9080, den Standard-HTTP-Port für Liberty. Wenn Sie einen anderen Port konfiguriert haben, passen Sie die URL entsprechend an (z. B. `http://localhost:YOUR_PORT/myApp/`).
- **Berechtigungen:** Stellen Sie sicher, dass Sie Schreibberechtigungen für das `dropins`-Verzeichnis haben.
- **JSP-Unterstützung:** Das `javaee-8.0`-Feature enthält JSP-Unterstützung, sodass keine zusätzliche Konfiguration erforderlich ist.
- **Fehlerbehebung:** Wenn die Seite nicht geladen wird, überprüfen Sie die Serverprotokolle unter `/opt/ibm/wlp/usr/servers/myServer/logs/messages.log` auf Fehler oder stellen Sie sicher, dass der Server läuft und die Dateien sich an der richtigen Stelle befinden.

Dieses einfache Beispiel zeigt, wie Sie eine “Hello World”-Anwendung auf Ihrem WebSphere Liberty-Server mit dem `dropins`-Verzeichnis für die automatische Bereitstellung bereitstellen und ausführen. Viel Spaß bei Ihrer ersten Liberty-Anwendung!

---

Hier ist eine Erklärung, wie eine “Hello World”-Anwendung funktioniert, wenn sie auf WebSphere Liberty bereitgestellt wird:

### 1. Das Dropins-Verzeichnis

WebSphere Liberty bietet eine bequeme Funktion zum Bereitstellen von Anwendungen über sein `dropins`-Verzeichnis. Dies ist ein spezieller Ordner, in dem Sie Ihre Anwendung –entweder als WAR-Datei (Web Application Archive) oder als entpacktes WAR-Verzeichnis –einfach platzieren können, und Liberty wird sie automatisch erkennen und bereitstellen. Im Gegensatz zu traditionellen Bereitstellungsmethoden müssen Sie die Anwendung nicht manuell in der `server.xml`-Datei konfigurieren. Wenn Liberty startet oder eine Änderung im `dropins`-Ordner (wie das Hinzufügen einer neuen Anwendung) bemerkt, startet es den Bereitstellungsprozess automatisch.

### 2. Verwendung eines entpackten WAR-Verzeichnisses

In diesem Beispiel wird die Anwendung als ein entpacktes WAR-Verzeichnis namens `myApp.war` bereitgestellt, anstatt einer einzigen gepackten WAR-Datei. Ein entpacktes WAR ist einfach ein Ordner, der alle Inhalte

einer Webanwendung (wie HTML, JSP-Dateien und andere Ressourcen) in entpackter Form enthält. Liberty behandelt diesen Ordner genau wie eine WAR-Datei und stellt ihn als voll funktionsfähige Webanwendung bereit. Diese Methode ist besonders während der Entwicklung nützlich, da Sie die Dateien direkt bearbeiten können (z. B. HTML oder JSP anpassen), ohne alles wieder in eine WAR-Datei packen zu müssen.

### 3. Die JSP-Datei

Das Herzstück dieser “Hello World”-Anwendung ist eine Datei namens `index.jsp`, eine JavaServer Page (JSP). Diese Datei enthält grundlegendes HTML, um “Hello World!” auf dem Bildschirm anzuzeigen, und könnte Java-Code enthalten, wenn dies erforderlich ist (in diesem Fall wird sie jedoch einfach gehalten). Wenn Sie auf die Anwendung zugreifen, kompiliert Liberty die JSP dynamisch in einen Servlet – ein kleines Java-Programm, das die Webseite erzeugt – und stellt sie Ihrem Browser zur Verfügung.

### 4. Aktivieren von Java EE-Funktionen

Damit all dies funktioniert, verlässt sich Liberty auf bestimmte Funktionen, die in seiner Konfigurationsdatei `server.xml` aktiviert sind. Hier wird das `javaee-8.0`-Feature aktiviert, das Unterstützung für Technologien wie JSPs, Servlets und andere Komponenten der Java Enterprise Edition (EE) 8-Plattform bietet. Diese Funktion stellt sicher, dass Liberty die notwendigen Bibliotheken und Einstellungen geladen hat, um die Anwendung reibungslos auszuführen.

### 5. Automatischer Bereitstellungsprozess

Sobald Sie das `myApp.war`-Verzeichnis in den `dropins`-Ordner legen und Liberty starten (oder wenn es bereits läuft), erkennt der Server die Anwendung automatisch und stellt sie bereit. Sie sehen Protokollnachrichten in der Liberty-Ausgabe, die anzeigen, dass die Anwendung gestartet wurde und unter einer bestimmten URL verfügbar ist. Dieser hands-off-Bereitstellungsprozess macht es schnell und einfach, eine Anwendung zum Laufen zu bringen.

### 6. Zugreifen auf die Anwendung: Kontextstamm

Die URL, über die Sie auf die Anwendung zugreifen können, wird durch ihren **Kontextstamm** bestimmt, der einfach der Name der WAR-Datei oder des Verzeichnisses ist. In diesem Fall lautet der Kontextstamm `/myApp`, da der Ordner `myApp.war` heißt. Standardmäßig läuft Liberty auf Port 9080, sodass die vollständige URL zum Zugriff auf die Anwendung `http://localhost:9080/myApp/` lautet.

### 7. Bereitstellen der Willkommensdatei

Wenn Sie `http://localhost:9080/myApp/` in Ihrem Browser aufrufen, muss Liberty herausfinden, was angezeigt werden soll. Es sucht nach einer **Willkommensdatei** – einer Standardseite, die bereitgestellt

wird, wenn keine spezifische Datei angefordert wird. In diesem Fall befindet sich `index.jsp` im Stammverzeichnis von `myApp.war` und wird standardmäßig als Willkommensdatei erkannt. Liberty kompiliert und stellt `index.jsp` bereit, die dann "Hello World!" auf Ihrem Bildschirm anzeigt.

## Zusammenfassung

Diese "Hello World"-Anwendung zeigt, wie WebSphere Liberty die Bereitstellung von Webanwendungen vereinfacht. Durch die Nutzung des `dropins`-Verzeichnisses für die automatische Bereitstellung, die Unterstützung entpackter WARs für einfache Bearbeitungen und die Verwendung des `javaee-8.0`-Features für die Java EE-Kompatibilität ermöglicht Liberty Ihnen, schnell eine einfache Webanwendung bereitzustellen und zu testen. Das Ergebnis ist eine leichte, entwicklerfreundliche Möglichkeit, "Hello World!" unter `http://localhost:9080/myApp/` zu sehen, perfekt zum Lernen oder Prototypen.