

WebSphere Liberty - Application Hello World

Voici un exemple simple pour exécuter une application “Hello World” sur votre serveur WebSphere Liberty, en supposant que vous l’avez déjà configuré avec la fonctionnalité `javaee-8.0` activée selon les instructions précédentes. Cet exemple utilise un fichier JSP (JavaServer Pages) de base et le déploie en utilisant le répertoire `dropins` pour des raisons de simplicité.

Guide Étape par Étape

1. Créer le Répertoire et le Fichier de l’Application Vous allez créer une petite application web en plaçant un fichier JSP dans le répertoire `dropins` de votre serveur Liberty. Le répertoire `dropins` permet à Liberty de détecter et de déployer automatiquement les applications.

- **Localiser le répertoire `dropins`:** Accédez au répertoire `dropins` dans le dossier de votre serveur. Si votre installation Liberty est située à `/opt/ibm/wlp` et que votre serveur s’appelle `myServer`, le chemin est :

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

Remplacez `/opt/ibm/wlp` par votre répertoire d’installation Liberty réel et `myServer` par le nom de votre serveur.

- **Créer un répertoire WAR décompressé:** À l’intérieur du répertoire `dropins`, créez un répertoire nommé `myApp.war`. L’extension `.war` indique à Liberty de traiter ceci comme une application web. Utilisez cette commande :

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

- **Créer le fichier `index.jsp`:** À l’intérieur de `myApp.war`, créez un fichier nommé `index.jsp` avec le contenu suivant pour afficher “Hello World!” :

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

Vous pouvez le créer directement avec une commande comme :

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

Alternativement, utilisez un éditeur de texte pour créer `index.jsp` et enregistrez-le à cet emplacement.

2. Démarrer le Serveur (si ce n'est pas déjà fait) Si votre serveur n'est pas en cours d'exécution, vous devez le démarrer pour qu'il puisse déployer et servir l'application.

- **Accéder au répertoire bin:** Allez au répertoire `bin` dans votre installation Liberty :

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

- **Démarrer le serveur:** Exécutez le serveur en mode premier plan pour voir la sortie directement :

```
./server run myServer
```

Alternativement, démarrez-le en arrière-plan :

```
./server start myServer
```

Si le serveur est déjà en cours d'exécution, ignorez cette étape—Liberty détectera automatiquement la nouvelle application.

3. Vérifier le Déploiement de l'Application Liberty déploiera automatiquement l'application `myApp.war` lorsqu'il la détectera dans le répertoire `dropins`.

- **Vérifier la sortie de la console:** Si vous avez démarré le serveur en mode premier plan, surveillez un message comme :

```
[AUDIT    ] CWWKT0016I: Web application available (default_host): http://localhost:9080/myApp/
```

Cela confirme que l'application est déployée et disponible.

- **Vérifier les journaux (si en arrière-plan):** Si le serveur est en cours d'exécution en arrière-plan, consultez le fichier journal à :

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

pour le même message de déploiement.

4. Accéder à l'Application Maintenant que l'application est déployée, vous pouvez la consulter dans un navigateur web.

- **Ouvrir votre navigateur:** Accédez à :

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

La racine de contexte `/myApp` provient du nom du répertoire `myApp.war`, et `index.jsp` est servi par défaut comme fichier d'accueil.

- **Résultat attendu:** Vous devriez voir une page affichant :

Hello World!

en grand texte.

Notes

- **Numéro de port** : Cet exemple utilise le port 9080, le port HTTP par défaut pour Liberty. Si vous avez configuré un port différent, ajustez l'URL en conséquence (par exemple, `http://localhost:YOUR_PORT/myApp/`).
- **Permissions** : Assurez-vous d'avoir les permissions d'écriture sur le répertoire `dropins`.
- **Support JSP** : La fonctionnalité `javaee-8.0` inclut le support JSP, donc aucune configuration supplémentaire n'est nécessaire.
- **Dépannage** : Si la page ne se charge pas, vérifiez les journaux du serveur à `/opt/ibm/wlp/usr/servers/myServer` pour les erreurs, ou assurez-vous que le serveur est en cours d'exécution et que les fichiers sont à l'emplacement correct.

Cet exemple simple démontre comment déployer et exécuter une application "Hello World" sur votre serveur WebSphere Liberty en utilisant le répertoire `dropins` pour le déploiement automatique. Profitez de votre première application Liberty !

Voici une explication de la manière dont une application "Hello World" fonctionne lorsqu'elle est déployée sur WebSphere Liberty :

1. Le Répertoire Dropins

WebSphere Liberty propose une fonctionnalité pratique pour déployer des applications via son répertoire `dropins`. Il s'agit d'un dossier spécial où vous pouvez simplement placer votre application—soit sous forme de fichier WAR (Web Application Archive) ou de répertoire WAR décompressé—and Liberty la détectera et la déployera automatiquement. Contrairement aux méthodes de déploiement traditionnelles, vous n'avez pas besoin de configurer manuellement l'application dans le fichier `server.xml`. Lorsque Liberty démarre ou remarque un changement dans le dossier `dropins` (comme l'ajout d'une nouvelle application), il lance automatiquement le processus de déploiement.

2. Utilisation d'un Répertoire WAR Décompressé

Dans cet exemple, l'application est déployée sous forme de répertoire WAR décompressé nommé `myApp.war` au lieu d'un fichier WAR unique. Un WAR décompressé est simplement un dossier contenant tous les contenus d'une application web (comme HTML, fichiers JSP et autres ressources) sous forme décompressée. Liberty traite ce répertoire exactement comme il le ferait avec un fichier WAR, le déployant comme une application web entièrement fonctionnelle. Cette méthode est particulièrement utile pendant le développement car vous pouvez éditer les fichiers directement (par exemple, ajuster le HTML ou le JSP) sans avoir besoin de tout repacker dans un fichier WAR.

3. Le Fichier JSP

Le cœur de cette application “Hello World” est un fichier appelé `index.jsp`, une page JavaServer (JSP). Ce fichier contient du HTML de base pour afficher “Hello World!” à l’écran et pourrait inclure du code Java si nécessaire (bien que dans ce cas, il soit gardé simple). Lorsque vous accédez à l’application, Liberty compile dynamiquement le JSP en un servlet—a small Java program that generates the webpage—and serves it to your browser.

4. Activation des Fonctionnalités Java EE

Pour que tout cela fonctionne, Liberty s’appuie sur des fonctionnalités spécifiques activées dans son fichier de configuration, `server.xml`. Ici, la fonctionnalité `javaee-8.0` est activée, ce qui fournit un support pour des technologies comme les JSP, les servlets et d’autres composants de la plateforme Java Enterprise Edition (EE) 8. Cette fonctionnalité garantit que Liberty dispose des bibliothèques et des paramètres nécessaires pour exécuter l’application en douceur.

5. Processus de Déploiement Automatique

Une fois que vous avez placé le répertoire `myApp.war` dans le dossier `dropins` et démarré Liberty (ou s’il est déjà en cours d’exécution), le serveur détecte et déploie automatiquement l’application. Vous verrez des messages de journal dans la sortie de Liberty indiquant que l’application a démarré et est disponible à une URL spécifique. Ce processus de déploiement sans intervention rend rapide et facile la mise en place d’une application.

6. Accéder à l’Application : Racine de Contexte

L’URL où vous pouvez accéder à l’application est déterminée par sa **racine de contexte**, qui est simplement le nom du fichier WAR ou du répertoire. Dans ce cas, puisque le répertoire s’appelle `myApp.war`, la racine de contexte devient `/myApp`. Par défaut, Liberty s’exécute sur le port 9080, donc l’URL complète pour accéder à l’application est `http://localhost:9080/myApp/`.

7. Servir le Fichier d’Accueil

Lorsque vous visitez `http://localhost:9080/myApp/` dans votre navigateur, Liberty doit déterminer ce qu’il doit afficher. Il recherche un **fichier d’accueil**—une page par défaut à servir lorsqu’aucun fichier spécifique n’est demandé. Dans ce cas, `index.jsp` est situé à la racine du répertoire `myApp.war` et est reconnu comme un fichier d’accueil par défaut. Liberty compile et sert `index.jsp`, qui affiche ensuite “Hello World!” à l’écran.

Résumé

Cette application “Hello World” montre comment WebSphere Liberty simplifie le déploiement des applications web. En utilisant le répertoire `dropins` pour le déploiement automatique, en prenant en charge les WAR décompressés pour une édition facile et en utilisant la fonctionnalité `javaee-8.0` pour la compatibilité Java EE, Liberty vous permet de déployer et de tester rapidement une application web simple sans configuration complexe. Le résultat est une méthode légère et conviviale pour les développeurs pour voir “Hello World!” à `http://localhost:9080/myApp/`, parfaite pour l’apprentissage ou le prototypage.