

# Configuration de la source de données JNDI dans WebSphere Liberty

Pour déterminer si un programme Java EAR utilise une source de données définie dans le fichier `server.xml` de WebSphere Liberty, vous devez vérifier si l'application accède à une source de données via un nom JNDI (Java Naming and Directory Interface) qui correspond à une source de données configurée dans la configuration du serveur, spécifiquement le fichier `server.xml` ou tout fichier qu'il inclut. Voici un guide étape par étape pour vous aider à confirmer cela :

## Étape 1 : Identifier le nom JNDI utilisé par l'application

La première tâche consiste à découvrir quel nom JNDI l'application EAR utilise pour rechercher sa source de données. Dans une application Java EE déployée sous forme de fichier EAR, cela peut être spécifié à plusieurs endroits :

- **Descripteurs de déploiement** : Vérifiez les descripteurs de déploiement de l'application, tels que `web.xml` (pour les modules web) ou `ejb-jar.xml` (pour les modules EJB), pour un élément `<resource-ref>`. Cet élément déclare une référence de ressource que l'application utilise pour accéder à une source de données. Par exemple :

```
<resource-ref>
  <res-ref-name>jdbc/myDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

Ici, l'application recherche la source de données en utilisant le nom JNDI `java:comp/env/jdbc/myDataSource`.

- **Fichiers de liaison** : Dans WebSphere Liberty, la référence de ressource du descripteur de déploiement peut être liée à un nom JNDI réel défini sur le serveur via des fichiers de liaison comme `ibm-web-bnd.xml` (pour les modules web) ou `ibm-ejb-jar-bnd.xml` (pour les EJB). Recherchez une liaison `<resource-ref>`, telle que :

```
<resource-ref name="jdbc/myDataSource" binding-name="jdbc/actualDataSource"/>
```

Dans ce cas, la référence de l'application `jdbc/myDataSource` est mappée au nom JNDI du serveur `jdbc/actualDataSource`.

- **Code de l'application** : Si vous avez accès au code source, recherchez les recherches JNDI ou les annotations :
  - **Recherche JNDI** : Recherchez du code comme :

```
Context ctx = new InitialContext();
DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/myDataSource");
```

Cela indique le nom JNDI `java:comp/env/jdbc/myDataSource`.

- **Annotations** : Dans les applications Java EE modernes, l'annotation `@Resource` peut être utilisée, comme :

```
@Resource(name = "jdbc/myDataSource")
private DataSource ds;
```

Cela pointe également vers `java:comp/env/jdbc/myDataSource`.

Si aucun fichier de liaison n'existe, le nom JNDI dans le code ou le descripteur de déploiement (par exemple, `jdbc/myDataSource`) peut correspondre directement au nom attendu dans la configuration du serveur.

## Étape 2 : Vérifier la configuration `server.xml`

Une fois que vous avez identifié le nom JNDI utilisé par l'application (directement ou via une liaison), vérifiez le fichier `server.xml` de WebSphere Liberty (et tout fichier de configuration inclus via un élément `<include>`) pour une définition de source de données correspondante. Une source de données dans `server.xml` est généralement définie avec un élément `<dataSource>`, comme ceci :

```
<dataSource id="myDataSource" jndiName="jdbc/myDataSource">
  <jdbcDriver libraryRef="myDBLib"/>
  <properties url="jdbc:mysql://localhost:3306/mydb" user="user" password="pass"/>
</dataSource>
```

- Recherchez l'attribut `jndiName` (par exemple, `jdbc/myDataSource`).
- Comparez-le au nom JNDI utilisé par l'application (par exemple, `jdbc/myDataSource` ou le nom lié comme `jdbc/actualDataSource`).

Si le nom JNDI correspond, l'application utilise la source de données définie dans `server.xml`.

## Étape 3 : Interpréter les résultats

- **Correspondance trouvée** : Si le nom JNDI utilisé par l'application correspond à un élément `<dataSource>` dans `server.xml` (ou un fichier inclus), alors le programme Java EAR utilise la source de données `server.xml`.
- **Pas de correspondance** : Si aucun nom JNDI ne correspond dans `server.xml`, l'application pourrait ne pas utiliser une source de données définie par le serveur. Elle pourrait créer sa propre source de données de manière programmatique (par exemple, en utilisant un pilote JDBC directement avec une URL, un nom d'utilisateur et un mot de passe), bien que cela soit moins courant et non recommandé dans un environnement géré comme WebSphere Liberty.

## Considérations supplémentaires

- **Fichiers de configuration inclus** : Si `server.xml` utilise des éléments `<include>` (par exemple, `<include location="datasources.xml"/>`), vérifiez ces fichiers pour les définitions de sources de données.
- **Comportement par défaut** : Si aucun fichier de liaison ne remplace le nom JNDI, l'application pourrait supposer que le nom de référence de ressource (par exemple, `jdbc/myDataSource`) correspond à une source de données dans `server.xml`.
- **Journaux du serveur ou outils** : Bien que cela ne fasse pas directement partie de la vérification de configuration, vous pourriez également examiner les journaux du serveur ou utiliser le WebSphere Liberty Admin Center pour voir quelles sources de données sont actives et associées à l'application, bien que cela soit plus une vérification en temps d'exécution.

## Conclusion

Pour confirmer si un programme Java EAR utilise la source de données `server.xml` dans WebSphere Liberty :

1. Identifiez le nom JNDI que l'application utilise (via les descripteurs de déploiement, les fichiers de liaison ou le code).
2. Vérifiez si ce nom JNDI correspond à une source de données définie dans `server.xml` ou ses fichiers inclus.

Si une correspondance est trouvée, l'application dépend de la source de données `server.xml` ; sinon, elle utilise probablement une approche alternative.