

JNDI-Datenquellenkonfiguration in WebSphere Liberty

Um zu bestimmen, ob ein EAR Java-Programm eine in der `server.xml`-Datei definierte Datenquelle in WebSphere Liberty verwendet, müssen Sie überprüfen, ob die Anwendung auf eine Datenquelle über einen JNDI (Java Naming and Directory Interface) Namen zugreift, der einer in der Serverkonfiguration, insbesondere der `server.xml`-Datei oder deren enthaltenen Dateien, konfigurierten Datenquelle entspricht. Hier ist eine Schritt-für-Schritt-Anleitung, um dies zu überprüfen:

Schritt 1: Identifizieren Sie den von der Anwendung verwendeten JNDI-Namen

Die erste Aufgabe besteht darin herauszufinden, welchen JNDI-Namen die EAR-Anwendung verwendet, um ihre Datenquelle zu suchen. In einer als EAR-Datei bereitgestellten Java EE-Anwendung kann dies an mehreren Stellen angegeben sein:

- **Bereitstellungsdeskriptoren:** Überprüfen Sie die Bereitstellungsdeskriptoren der Anwendung, wie `web.xml` (für Web-Module) oder `ejb-jar.xml` (für EJB-Module), auf ein `<resource-ref>`-Element. Dieses Element deklariert eine Ressourcenreferenz, die die Anwendung verwendet, um auf eine Datenquelle zuzugreifen. Zum Beispiel:

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

Hier sucht die Anwendung die Datenquelle mit dem JNDI-Namen `java:comp/env/jdbc/myDataSource`.

- **Bindungsdateien:** In WebSphere Liberty kann die Ressourcenreferenz aus dem Bereitstellungsdeskriptor über Bindungsdateien wie `ibm-web-bnd.xml` (für Web-Module) oder `ibm-ejb-jar-bnd.xml` (für EJBs) an einen tatsächlichen JNDI-Namen gebunden werden, der auf dem Server definiert ist. Suchen Sie nach einer `<resource-ref>`-Bindung, wie:

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

In diesem Fall wird die Anwendungsreferenz `jdbc/myDataSource` auf den JNDI-Namen des Servers `jdbc/actualDataSource` abgebildet.

- **Anwendungscode:** Wenn Sie Zugriff auf den Quellcode haben, suchen Sie nach JNDI-Suchen oder Annotationen:
 - **JNDI-Suche:** Suchen Sie nach Code wie:

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

Dies zeigt den JNDI-Namen `java:comp/env/jdbc/myDataSource`.

- **Annotationen:** In modernen Java EE-Anwendungen kann die `@Resource`-Annotation verwendet werden, wie:

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

Dies verweist ebenfalls auf `java:comp/env/jdbc/myDataSource`.

Wenn keine Bindungsdatei vorhanden ist, kann der JNDI-Name im Code oder im Bereitstellungsdeskriptor (z. B. `jdbc/myDataSource`) direkt dem Namen entsprechen, der in der Serverkonfiguration erwartet wird.

Schritt 2: Überprüfen Sie die `server.xml`-Konfiguration

Sobald Sie den JNDI-Namen identifiziert haben, den die Anwendung verwendet (entweder direkt oder über eine Bindung), überprüfen Sie die WebSphere Liberty `server.xml`-Datei (und alle Konfigurationsdateien, die über ein `<include>`-Element enthalten sind) auf eine passende Datenquellendefinition. Eine Datenquelle in `server.xml` wird in der Regel mit einem `<dataSource>`-Element definiert, wie folgt:

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

- Suchen Sie nach dem Attribut `jndiName` (z. B. `jdbc/myDataSource`).
- Vergleichen Sie es mit dem von der Anwendung verwendeten JNDI-Namen (z. B. `jdbc/myDataSource` oder dem gebundenen Namen wie `jdbc/actualDataSource`).

Wenn der JNDI-Name übereinstimmt, verwendet die Anwendung die in `server.xml` definierte Datenquelle.

Schritt 3: Interpretieren Sie die Ergebnisse

- **Übereinstimmung gefunden:** Wenn der von der Anwendung verwendete JNDI-Name einem `<dataSource>`-Element in `server.xml` (oder einer enthaltenen Datei) entspricht, verwendet das EAR Java-Programm die `server.xml`-Datenquelle.
- **Keine Übereinstimmung:** Wenn es keinen übereinstimmenden JNDI-Namen in `server.xml` gibt, verwendet die Anwendung möglicherweise keine serverdefinierte Datenquelle. Sie könnte ihre eigene Datenquelle programmgesteuert erstellen (z. B. unter Verwendung eines JDBC-Treibers direkt mit einer URL, einem Benutzernamen und einem Passwort), dies ist jedoch weniger üblich und in einer verwalteten Umgebung wie WebSphere Liberty nicht empfohlen.

Zusätzliche Überlegungen

- **Enthaltene Konfigurationsdateien:** Wenn `server.xml` `<include>`-Elemente verwendet (z. B. `<include location="datasources.xml"/>`), überprüfen Sie diese Dateien auf Datenquellendefinitionen.
- **Standardverhalten:** Wenn keine Bindungsdatei den JNDI-Namen überschreibt, könnte die Anwendung annehmen, dass der Ressourcenreferenzname (z. B. `jdbc/myDataSource`) mit einer Datenquelle in `server.xml` übereinstimmt.
- **Serverprotokolle oder Tools:** Obwohl dies nicht direkt Teil der Konfigurationsprüfung ist, können Sie auch Serverprotokolle überprüfen oder das WebSphere Liberty Admin Center verwenden, um zu sehen, welche Datenquellen aktiv sind und mit der Anwendung verknüpft sind, dies ist jedoch eher eine Laufzeitverifikation.

Fazit

Um zu bestätigen, ob ein EAR Java-Programm die `server.xml`-Datenquelle in WebSphere Liberty verwendet:

1. Identifizieren Sie den JNDI-Namen, den die Anwendung verwendet (über Bereitstellungsdeskriptoren, Bindungsdateien oder Code). 2. Überprüfen Sie, ob dieser JNDI-Name mit einer in `server.xml` oder deren enthaltenen Dateien definierten Datenquelle übereinstimmt.

Wenn es eine Übereinstimmung gibt, verwendet die Anwendung die `server.xml`-Datenquelle; andernfalls verwendet sie wahrscheinlich einen alternativen Ansatz.