WildFly + Hibernate

Asked 2 years, 9 months ago Active 1 year, 6 months ago Viewed 10k times

3

3

I've been struggling for days with configuring Hibernate and run it on WildFly.

Here is my code:

META-INF/persistence.xml

```
<persistence version="1.0"
         xmlns="http://java.sun.com/xml/ns/persistence"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
   http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">

<persistence-unit name="blog" transaction-type="JTA">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

    <properties>
        <property name="hibernate.connection.url"
value="jdbc:mysql://localhost:3306/blog?createDatabaseIfNotExist=true"/>
        <property name="hibernate.connection.driver_class"
value="com.mysql.jdbc.Driver"/>
        <property name="hibernate.connection.username" value="abc"/>
        <property name="hibernate.connection.password" value="abc"/>
        <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>
        <property name="hibernate.hbm2ddl.auto" value="create"/>
        <property name="hibernate.show_sql" value="true"/>
        <property name="jboss.as.jpa.providerModule" value="org.hibernate:5.0"/>
    </properties>

</persistence-unit>

      .
    <dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-api</artifactId>
        <version>7.0</version>
    </dependency>

    <dependency>
        <groupId>org.hibernate.javax.persistence</groupId>
        <artifactId>hibernate-jpa-2.1-api</artifactId>
        <version>1.0.0.Final</version>
    </dependency>

    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>5.2.2.Final</version>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.1.38</version>
    </dependency>

</dependencies>

<properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

User:

```
@Entity
@Table(name = "users")
public class User {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private long id;
@Column(name = "password")
private String password;

@OneToMany(mappedBy = "user")
private Set<Post> posts;
}
```

When I load my main page no database is created.

Moreover I want to persist a User.

```
@Stateless
public class UserRepositoryImpl implements UserRepository {

@PersistenceContext(unitName = "blog")
private EntityManager entityManager;

public void create(User user) {
    this.entityManager.persist(user);
}
}
```

No database is created and the entity manager is null. What do I need to configure to make it run? I am using IntelliJ for testing.

hibernate jpa java-ee intellij-idea wildfly

asked Jan 4 '17 at 14:57

Teodor Dimitrov
510 1 5 14

2 First of all, when using `JTA` as transaction type, the database connection details shouldn't be in the `persistence.xml` file. They should be in the `standalone.xml` from your WilfFly. Then the `persistence.xml` just reference it. – cassiomolin Jan 4 '17 at 15:03

I guess something else is fundamentally broken. I can't find a good tutorial how to do it. I'm leaving it for now. Thanks – Teodor Dimitrov Jan 4 '17 at 16:06

The problem is likely in the way you use the UserRepositoryImpl. As in: the bit of code you did not present here. Apparently the JEE container is not triggered to actually inject the entityManager instance and thus start up the persistence context, which I can only assume means that the USerRepositoryImpl is not initialized as a proper EJB instance. – Gimby Jan 5 '17 at 12:39

8

A JTA datasource is managed by the jpa container (inside wildfly). You must define the url, username, password in the standalone.xml. Search for the datasources subsystem `<subsystem xmlns="urn:jboss:domain:datasources:4.0">` and add a datasource definition, for example:

```
<datasource jta="true" jndi-name="java:/jdbc/myDS" pool-name="MyDS"
enabled="true" use-java-context="true" use-ccm="true">
    <connection-url>jdbc:mysql://localhost:3306/blog</connection-url>
    <driver>mysqldriver.jar</driver>
    <security>
        <user-name>username</user-name>
        <password>password</password>
    </security>
</datasource>
```

Next you have to create a module for your database driver. For the details check the documentation: https://docs.jboss.org/author/display/WFLY8/DataSource+configuration

Then your persistence.xml will look like this:

```
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
         http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
    version="2.1">

    <persistence-unit name="blog">
        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
        <jta-data-source>jdbc/myDS</jta-data-source>
        <properties>
            <property name="hibernate.archive.autodetection" value="class" />
             <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>
         <property name="hibernate.hbm2ddl.auto" value="create"/>
        <property name="hibernate.show_sql" value="true"/>
         </properties>
     </persistence-unit>
 </persistence>
```

edited Mar 28 '18 at 0:24 answered Jan 9 '17 at 15:05

dwjohnston 3,601 14 54 106

patst 93 5

-2

In file persistence.xml you must include class to persist

```
<provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

<class>package.User<class>

<properties>
    <property name="hibernate.connection.url"
value="jdbc:mysql://localhost:3306/blog?createDatabaseIfNotExist=true"/>
    <property name="hibernate.connection.driver_class"
value="com.mysql.jdbc.Driver"/>
    <property name="hibernate.connection.username" value="abc"/>
    <property name="hibernate.connection.password" value="abc"/>
    <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>
    <property name="hibernate.hbm2ddl.auto" value="create"/>
    <property name="hibernate.show_sql" value="true"/>
    <property name="jboss.as.jpa.providerModule" value="org.hibernate:5.0"/>
</properties>
```

You can include as many classes as you like

answered Jan 5 '17 at 11:53

Borja 1 1

3 No you do not, in a JEE container like Wildfly entities are automatically discovered upon application deployment. – Gimby Jan 5 '17 at 12:40