

## Uso do JPA com EJB

Este exercício trata de um componente EJB que realiza persistência usando JPA. Para definir a camada de persistência, apresentaremos as etapas para sua realização. Como exercício o aluno definirá a camada de apresentação usando Servlets e JSP.

1. Definição do datasource (para várias aplicações, definir um datasource no servidor JBoss): na pasta **[JBoss]\modules\com\mysql\main** (criar a pasta):

- a. copiar o driver do mysql
- b. criar o arquivo module.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.13-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

2. Localizar o arquivo **standalone.xml** na pasta **[JBoss]\standalone\configuration**. Dentro da tag **<datasources>** incluir o elemento:

```
<datasource jndi-name="java:/jdbc/fiap" pool-name="fiap" enabled="true" use-java-context="true">
  <connection-url>jdbc:mysql://localhost:3306/livrosejb</connection-url>
  <driver>com.mysql</driver>
  <security>
    <user-name>root</user-name>
    <password>fiap</password>
  </security>
</datasource>
```

e dentro da tag **<drivers>** a seguinte configuração:

```
<driver name="com.mysql" module="com.mysql">
  <xa-datasource-class>com.mysql.jdbc.Driver</xa-datasource-class>
</driver>
```

3. Criar um novo projeto **EjbProject**, chamado **03\_LivrosEJB**. Selecionar Wildfly 9.x. **Não Incluir o projeto em um repositório EAR.**
4. Na pasta **ejbModule**, criar a classe para a entidade Livros:

```
package br.com.fiap.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="livros", schema="livrosejb")
public class Livros implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="ID")
    private Integer id;

    @Column(name="TITULO")
    private String titulo;

    @Column(name="AUTOR")
    private String autor;

    @Column(name="PRECO")
    private Double preco;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getTitulo() {
        return titulo;
    }
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }
    public String getAutor() {
        return autor;
    }
    public void setAutor(String autor) {
        this.autor = autor;
    }
    public Double getPreco() {
        return preco;
    }
    public void setPreco(Double preco) {
        this.preco = preco;
    }
}
```

5. Criar, na pasta \META-INF, o arquivo **persistence.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0"
    xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

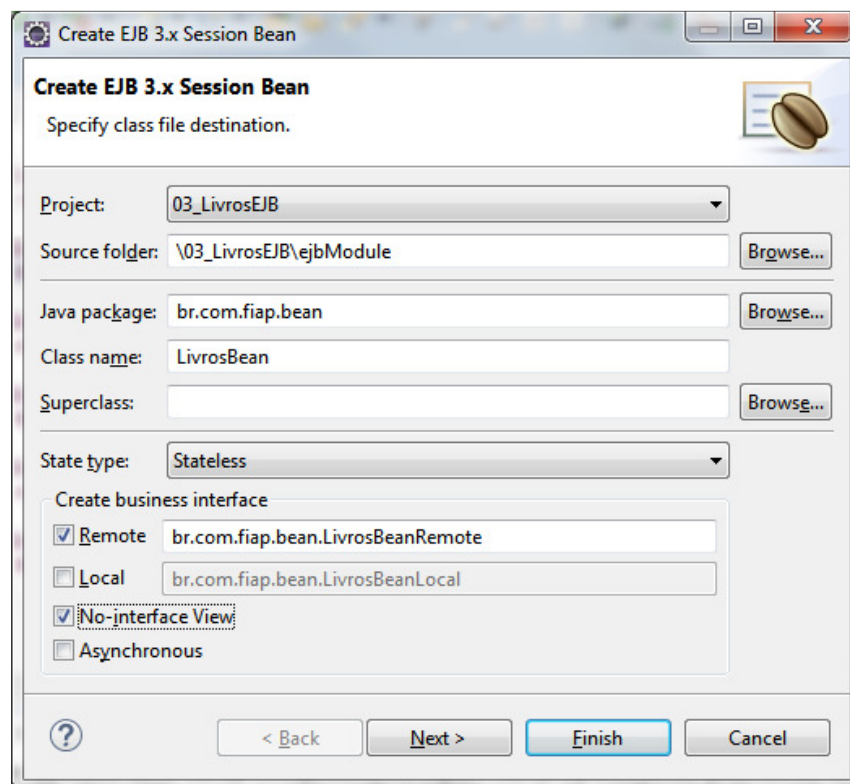
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">

<persistence-unit name="fiapPU" transaction-type="JTA">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <jta-data-source>jdbc/fiap</jta-data-source>
  <properties>
    <property name="hibernate.show_sql" value="true" />
    <property name="hibernate.format_sql" value="true" />
    <property name="hibernate.hbm2ddl.auto" value="update" />
    <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5InnoDBDialect" />
  </properties>
</persistence-unit>
</persistence>

```

6. Definir o banco de dados no MySQL.

7. Criar o SessionBean e a interface **remota**. Chamar a classe de **LivrosBean**:



8. Codificar a classe e a interface:

```

package br.com.fiap.bean;

import java.util.List;

import javax.ejb.Remote;
import br.com.fiap.entity.Livros;

```

```
@Remote
public interface LivrosBeanRemote {
    void add(Livros livro);
    List<Livros> getAll();
}

package br.com.fiap.bean;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.TypedQuery;

import br.com.fiap.entity.Livros;

@Stateless
public class LivrosBean implements LivrosBeanRemote {

    @PersistenceContext(unitName="fiapPU")
    private EntityManager em;

    @Override
    public void add(Livros livro){
        em.persist(livro);
    }

    @Override
    public List<Livros> getAll(){
        TypedQuery<Livros> query = em.createQuery("select u from Livros u",
Livros.class);
        return query.getResultList();
    }
}
```

9. Iniciar o servidor e verificar se o datasource foi definido:

**Bound data source [java:/jdbc/fiap]**

10. Criar um projeto **Dynamic Web Project** chamado **03\_LivrosWeb**.

11. A partir do projeto **03\_Livros**, criar um arquivo jar incluindo o pacote com a interface e a entidade. Adicionar este arquivo ao projeto web

12. Usando Servlet e JSP, criar a aplicação cliente, tanto para inserir como para listar os livros. Como acesso ao EJB, basear-se no código a seguir:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    PrintWriter out = response.getWriter();
}
```

```

try {
    InitialContext ctx = new InitialContext();
    LivrosBeanRemote service =
        (LivrosBeanRemote)ctx.
lookup("ejb:/03_LivrosEJB/LivrosBean!br.com.fiap.bean.LivrosBeanRemote");
    List<Livros> lista = service.getAll();
    request.setAttribute("lista", lista);
    request.getRequestDispatcher("lista.jsp").forward(request, response);

} catch (NamingException e) {
    out.print(e.getMessage());
    //e.printStackTrace();
}
}

```

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <h1>Lista de Livros</h1>
    <ul>
    <c:forEach var="liv" items="${lista}">
        <li>${liv.titulo}</li>
    </c:forEach>
    </ul>
</body>
</html>

```

Bom trabalho!

Prof. Emilio