


06 Arrumando a casa




35%


ATIVIDADES
6 de 10

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD



MODO
NOTURNO



ABRIR
CADERNO

Vamos integrar o JPA com os EJBs? Primeiro arrumaremos a casa!

Downloads:

- Arquivos de configuração (<https://s3.amazonaws.com/caelum-online-public/ejb/resources.zip> (<https://s3.amazonaws.com/caelum-online-public/ejb/resources.zip>))

1) Primeiro precisamos preparar nosso servidor para trabalhar com o banco de dados *MySQL*. Faremos isso registrando um módulo no JBoss AS com o driver do *MySQL*.

a) Procure a pasta onde o JBoss AS foi instalado (descompactado) e crie, dentro da subpasta `modules > com`, uma pasta chamada `mysql`, e dentro desta nova pasta, uma outra pasta chamada `main`. Por fim teremos a estrutura `$JBOSS_HOME/modules/com/mysql/main`:

```
$JBOSS_HOME
  \- modules
      \- com
          \- mysql
              \- main
```

COPIAR CÓDIGO

b) Depois, abra a pasta `resources`, que foi fornecida no início do treinamento, e procure os arquivos `module.xml` e `mysql-connector-java-5.1.XX-bin.jar` dentro da subpasta `module-mysql`. Copie-os para a pasta `main` que você acaba de criar.

```
$JBOSS_HOME
  \- modules
      \- com
          \- mysql
              \- main
                  \- module.xml
                  \- mysql-connector-
java-5.1.XX-bin.jar
```

COPIAR CÓDIGO





35%

ATIVIDADES

6 de 10

FÓRUM DO

CURSO

VOLTAR

PARA

DASHBOARD



MODO NOTURNO



ABRIR CADERNO

c) Agora vamos editar o arquivo `standalone.xml` , que está na pasta `standalone > configuration` . Procure o elemento `<drivers>` e dentro dessa seção cole as linhas de configuração que estão preparadas no arquivo `datasource-jboss7.txt` da pasta `resources` que você baixou.

```
<drivers>
  <!-- Demais drivers declarados aqui
-->

  <driver name="com.mysql"
module="com.mysql">
    <xa-datasource-class>

com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
    </xa-datasource-class>
  </driver>
</drivers>
```

COPIAR CÓDIGO

d) Por último, vamos configurar o componente que no JEE chamamos de *DataSource*, nome simplificado que damos a interface do *pool* de conexões disponibilizado pelo JEE à aplicação. Copie as linhas correspondentes a configuração do `DataSource` , preparadas no arquivo `datasource-jboss7.txt` da pasta `resources` , e cole-as na seção `<datasources>` ainda do arquivo `standalone.xml` onde já deve haver por padrão um `DataSource` exemplo configurado:

```
<datasources>
  <!-- DataSource exemplo declarado aqui
-->

  <datasource jndi-
name="java:/livrariaDS" pool-
name="livrariaDS"
    enabled="true" use-java-
context="true">

    <connection-
url>jdbc:mysql://localhost:3306
/livraria</connection-url>
    <driver>com.mysql</driver>
    <pool>
      <min-pool-size>10</min-pool-
size>
      <max-pool-size>100</max-pool-
size>
```



35%

ATIVIDADES
6 de 10

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD



MODOS
NOTURNOS



ABRIR
CADERNOS

```
<prefill>true</prefill>
</pool>
<security>
  <user-name>root</user-name>
  <password></password>
</security>
</datasource>
<drivers>
  <!-- Drivers declarados aqui -->
</drivers>
</datasources>
```

COPIAR CÓDIGO

2) Após salvar todos os arquivos, volte ao eclipse e reinicie o servidor JBoss AS ficando atento as informações do console. Tivemos um erro?

3) Pois é, o JBoss AS lançou uma *exception* dando como causa: `Unknown database 'livraria'` . Faltou então criar o banco de dados *livraria* no MySQL Server. Abra o console de comandos do MySQL e crie então o banco de dados. No nosso caso, digitamos no terminal do linux:

```
$ mysql -u root
```

COPIAR CÓDIGO

Já no console de comandos do MySQL digitamos:

```
mysql> CREATE DATABASE livraria;
```

COPIAR CÓDIGO

4) Reinicie novamente o servidor JBoss AS e veja pelo console que a *exception* não existe mais, pois o banco de dado já foi criado no MySQL. Não feche o console de comandos do MySQL ainda pois vamos utilizá-lo no próximo exercício.