ueno Tutorial sobre Hibernate

cado em 10/10/2009 - 204.428 visualizações

comentários: 39

Publicado por carlosbarretto

Em homenagem ao Luiz Carlos Metzger que me ajudou durante o dia todo (17/02/2004) com as minhas dúvidas sobre o Hibernate vou filtrar tudo o que foi feito no post em que discutimos:

Antes que você possa seguir os passos deste exemplo é necessário que tenha baixado o Hibernate-2.1 no próprio www.hibernate.org e importar os .jar da pasta hibernate\lib para a lib do seu projeto ou IDE ou JVM.

Segue os 5 passos necessários para este pequeno exemplo de persistência de dados:

1. criar a tabela no seu banco de dados onde os objetos vão persistir;
2. criar o objeto cujo estado vai ser persistido;
3. criar um XML, que relaciona as propriedades do objeto aos campos na tabela
4. criar um arquivo contendo as propriedades para que o Hibernate se conecte ao bando de dados
5. criar a classe DAO que vai persistir seu objeto;

ANTES DE COMEÇAR LEMBRE-SE: TODOS ARQUIVOS DEVEM SER SALVOS NA MESMA PASTA!

passo 1. criando uma tabela de amigos no MySQL, você pode usar qualquer banco (afinal este é o Hibernate) mas eu preferi usar MySQL

```
create table amigos(
nome varchar(40) NOT NULL default '',
endereco varchar(60) not null default '',
fone varchar(11) default null,
cel varchar(11) default null,
email varchar(70) default null,
nascimento datetime default null,
PRIMARY KEY (`nome`)
);
```

passo 2. Criar uma classe Amigo com seus atributos (campos do db) e métodos (set e get para cada atributo)

```
public class Amigo {
   private String nome;
   private String endereco;
   private String telefone;
   private String celular;
   private String email;
   private java.util.Date nascimento;

   public Amigo() {
   }

   public String getNome(){
      return nome;
   }

   public void setNome(String nome){
      this.nome = nome;
   }

   public String getEndereco(){
      return endereco;
   }

   public void setEndereco(String endereco){
      this.endereco = endereco;
   }

   public String getTelefone(){
      return telefone;
   }

   public void setTelefone(String telefone){
      this.telefone = telefone;
```

```
    }

    public String getCelular(){
        return celular;
    }

    public void setCelular(String celular){
        this.celular = celular;
    }

    public String getEmail(){
        return email;
    }

    public void setEmail(String email){
        this.email = email;
    }

    public java.util.Date getNascimento(){
        return nascimento;
    }

    public void setNascimento(java.util.Date nascimento){
        this.nascimento = nascimento;
    }

}
```

passo 3. Criar o Xml que irá relacionar os atributos com os campos da tabela

salvar este arquivo como Amigo.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN" "http://hibernate.sourceforge.net/hibernate-
mapping.dtd">
<hibernate-mapping>
    <class name="Amigo" table="amigos">
        <id name="nome" column="nome" type="string">
            <generator class="assigned"/>
        </id>
        <property name="endereco" type="string"/>
        <property name="telefone" column="fone" type="string"/>
        <property name="celular" column="cel" type="string"/>
        <property name="email" type="string"/>
        <property name="nascimento" type="date"/>
    </class>
</hibernate-mapping>
```

passo 4. Criar dois arquivos de propriedades para que o Hibernate se conecte ao Banco de Dados e para exibir os passos em um LOG

Salvar este arquivo como hibernate.properties

```
hibernate.dialect=net.sf.hibernate.dialect.MySQLDialect
hibernate.connection.driver_class = org.gjt.mm.mysql.Driver
hibernate.connection.url = jdbc:mysql://localhost:3306/test
hibernate.connection.username = root
hibernate.connection.password = root
```

e Salvar este arquivo como log4j.properties

```
log4j.rootLogger=DEBUG, dest1

log4j.appender.dest1=org.apache.log4j.ConsoleAppender
log4j.appender.dest1.layout=org.apache.log4j.PatternLayout
log4j.appender.dest1.layout.ConversionPattern=%d %-5p %-5c{3} %x -> %m%n

#log4j.appender.dest2=org.apache.log4j.RollingFileAppender
#log4j.appender.dest2.File=bridge.log

#log4j.appender.dest2.MaxFileSize=100KB
# Keep one backup file
#log4j.appender.dest2.MaxBackupIndex=3

#log4j.appender.dest2.layout=org.apache.log4j.PatternLayout
#log4j.appender.dest2.layout.ConversionPattern=%d [%t] %-5p %-5c{3}(%L) %x -> %m%n
```

passo 5. Criar a classe DAO que vai persistir o objeto Amigo

```
import java.util.List;
```

```
import net.sf.hibernate.*;
import net.sf.hibernate.cfg.Configuration;

public class AmigoDAO{

   private SessionFactory factory;

   public AmigoDAO() throws Exception{
       factory = new Configuration().addClass(Amigo.class).buildSessionFactory();
   }

   public void insert(Amigo amigo) throws Exception{
      Session session = factory.openSession();
      session.save(amigo);
      session.flush();
      session.close();
   }

   public java.util.List getList(String condicao) throws Exception{
      Session session = factory.openSession();
      List amigos = session.find(condicao);
      session.flush();
      session.close();
      return amigos;
   }

   public Amigo retrieve(String pk) throws Exception{
      Session session = factory.openSession();
      Amigo amigo = (Amigo)session.load(Amigo.class, pk);
      session.flush();
      session.close();
      return amigo;
   }

   public void delete(Amigo amigo) throws Exception{
      Session session = factory.openSession();
      session.delete(amigo);
      session.flush();
      session.close();
   }
}
```

para testar todos os passos acima criaremos um arquivo que eu chamei de TesteAmigo.java onde vamos instanciar as duas classes criadas e coloca-las para funcionar, segue abaixo

```
public class TesteAmigo {

   public static void main(String[] args) throws Exception {

     try
     {
       Amigo amigo = new Amigo();
       amigo.setNome("seu nome");
       amigo.setEndereco("seu endereco");
       amigo.setTelefone("seu fone");
       amigo.setCelular("seu celular");
       amigo.setEmail("seu mail");
       //amigo.setNascimento("data do tipo Date");

       AmigoDAO dao = new AmigoDAO();
       dao.insert(amigo);


     }
     catch(Exception e)
     {
       e.printStackTrace();//aqui vc vai saber que xabu é esse.
     }
   }
}
```

Se seguirem os passos um a um poderão fazer a sua primeira persistência de dados utilizando o Hibernate

Leia também:
Acessando Banco de Dados em Java (PARTE 1)
Acessando Banco de Dados em Java (PARTE 2)
Acessando Banco de Dados em Java (PARTE 3)

Aplicativo Java acessando banco de dados:
Aplicativo Java com acesso a banco de dados: 1º parte - Dao
Acessando Dados com Java: Parte 2 - Prevendo problemas

Quer aprender mais sobre Java?

O que é Java?
Características Básicas
Orientação a Objetos

Tutoriais para Certificação Java
Fundamentos da Linguagem
Modificadores
Operadores e atribuições
Controle de Fluxo
Orientação a Objetos
Java Lang e Wrappers
Objetos e Conjuntos
Classes Internas
Threads (Segmentos)

Banco de Dados
Acessando Banco de Dados em Java (PARTE 1)
Acessando Banco de Dados em Java (PARTE 2)
Acessando Banco de Dados em Java (PARTE 3)

Aplicativo Java acessando banco de dados:
Aplicativo Java com acesso a banco de dados: 1º parte - Dao
Acessando Dados com Java: Parte 2 - Prevendo problemas

Quer aprender mais sobre Java?
O que é Java?
Características Básicas
Orientação a Objetos

Tutoriais para Certificação Java
Fundamentos da Linguagem
Modificadores
Operadores e atribuições
Controle de Fluxo
Orientação a Objetos
Java Lang e Wrappers
Objetos e Conjuntos
Classes Internas
Threads (Segmentos)

comentários: 39

Tópicos Relacionados

- Consulta Brasil Telecom
- Swing e database
- Tutorial de instalação Hibernate
- Formulários e Janelas MDI
- Dúvida em diagrama de classe + desempenho usando Hibernate
- Onde colocar o hibernate.properties ?
- Pontuação Tempo Real Fevereiro 2004
- Ajuda com o componente JTable
- sistema web struts + hibernate
- Compilação e Debugação no Eclipse
- hibernate e javabeans
- Andando em circulos

- Compilando no Linux - Ubuntu
- Segurança COM Applet
- sobre tutorial de Hibernate!
- Livro bom sobre J2EE / JSP
- Dúvida Hibernate..