

Akses Database melalui Hibernate

Eko Budhi Suprasetiawan

ekobs@developerforce.net

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

(*) *Kerjasama IlmuKomputer.Com dan DeveloperForce.Net*

Motivasi

Hibernate membebaskan developer untuk menulis SQL melalui JDBC dalam mengakses database.

Persiapan

Persiapkan direktori kerja, dalam contoh ini /home/lab/myjava

Dapatkan Hibernate library

di-extract ke /home/lab/hibernate-1.2

Deskripsi

Sebuah Java Bean EmployeeBean akan disimpan/dibaca dari table EMPLOYEE_TBL di database. EmployeeBean.hbm.xml digunakan untuk memetakan antara property dari EmployeeBean dan column di EMPLOYEE_TBL Tulis hibernate.properties digunakan sebagai database configuration.

Langkah

1. Tulis EmployeeBean.java
2. Tulis hibernate.properties
3. Tulis EmployeeBean.hbm.xml
4. Tulis EmployeeApp.java

5. Set CLASSPATH
6. Compile
7. Launch

EmployeeBean

Simpan di sub directory employee di bawah /home/lab/myjava

employee/EmployeeBean.java

```
package employee;

import java.util.Date;

public class EmployeeBean
{
    /* DATA */

    private String id;
    private String name;
    private String department;
    private String jobTitle;
    private Date hireDate;
    private boolean permanentEmployee;
    private double salary;

    public void setId(String id)
    {
        this.id = id;
    }

    public String getId()
    {
        return this.id;
    }

    public void setName(String name)
    {
        this.name = name;
    }
    public String getName()
    {
        return this.name;
    }

    public void setDepartment(String department)
    {
        this.department = department;
    }
    public String getDepartment()
```

```
{
    return this.department;
}

public void setJobTitle(String jobTitle)
{
    this.jobTitle = jobTitle;
}
public String getJobTitle()
{
    return this.jobTitle;
}

public void setHireDate(Date hireDate)
{
    this.hireDate = hireDate;
}
public Date getHireDate()
{
    return this.hireDate;
}
public void setPermanentEmployee(boolean permanentEmployee )
{
    this.permanentEmployee = permanentEmployee;
}
public boolean getPermanentEmployee()
{
    return this.permanentEmployee;
}
public void setSalary(double salary)
{
    this.salary = salary;
}
public double getSalary()
{
    return this.salary;
}
}
```

Configuration

Simpan di sub directory employee di bawah /home/lab/myjava

hibernate.properties

```
hibernate.connection.driver_class = org.gjt.mm.mysql.Driver
hibernate.connection.url = jdbc:mysql://localhost:3306/MYAPP_DB
hibernate.connection.username = ekobs
hibernate.connection.password = j2ee
hibernate.connection.dialect = cirrus.hibernate.sql.MySQLDialect
```

Mapping

Simpan di sub directory employee di bawah /home/lab/myjava

Employee/EmployeeBean.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
Version 1.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping.dtd">

<hibernate-mapping>
  <class name="employee.EmployeeBean" table="EMPLOYEE_TBL">
    <id name="id" column="ID">
      <generator class="assigned"/>
    </id>
    <property name="name" column="NAME"/>
    <property name="department" column="DEPARTMENT"/>
    <property name="jobTitle" column="JOB_TITLE"/>
    <property name="hireDate" column="HIRE_DATE" type="date"/>
    <property name="permanentEmployee" column="PERMANENT_EMPLOYEE"/>
    <property name="salary" column="SALARY" type="double"/>
  </class>
</hibernate-mapping>
```

Client App

Simpan di sub directory app di bawah /home/lab/myjava

app/EmployeeApp.java

```
package app;

import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.*;

import cirrus.hibernate.*;

import employee.*;

public class EmployeeApp
{
    public static void main(String[] args)
    {
        try
        {
            Datastore ds = Hibernate.createDatastore();
            ds.storeClass(EmployeeBean.class);
            SessionFactory sessionFactory =
ds.buildSessionFactory();

            EmployeeBean bean = new EmployeeBean();
            bean.setId("032");
            bean.setName("Muhammad Sigit Arifianto");
            bean.setDepartment("Business Development");
            bean.setJobTitle("Account Manager");
            bean.setHireDate(new java.util.Date());
            bean.setPermanentEmployee(true);
            bean.setSalary(5000);

            System.out.println("E M P L O Y E E --- BEFORE STORE");
            System.out.println("Id : " + bean.getId());
            System.out.println("Name : " + bean.getName());
            System.out.println
                ("Department : " + bean.getDepartment());
            SimpleDateFormat format
                = new SimpleDateFormat("dd/MMMM/yyyy");
            System.out.println
                ("Hire Date : "
                    + format.format(bean.getHireDate()));
            System.out.println
                ("Permanent Emp.: "
                    + bean.getPermanentEmployee());
            System.out.println("Salary : " + bean.getSalary());
            System.out.println();
        }
    }
}
```

```
        Session session = sessionFactory.openSession();
        session.save(bean);
        session.flush();
        session.connection().commit();
        session.close();

        session = sessionFactory.openSession();

        List beans = session.find("from o in class
employee.EmployeeBean where id='032'");
        bean = (EmployeeBean) beans.get(0);

        System.out.println("E M P L O Y E E --- AFTER LOAD");
        System.out.println("Id : " + bean.getId());
        System.out.println("Name : " + bean.getName());
        System.out.println
            ("Department : " + bean.getDepartment());

        System.out.println
            ("Hire Date : "
            + format.format(bean.getHireDate()));

        System.out.println
            ("Permanent Emp.: "
            + bean.getPermanentEmployee());
        System.out.println("Salary : " + bean.getSalary());
        System.out.println();
    }
    catch(HibernateException e)
    {
        e.printStackTrace();
    }
    catch(SQLException e)
    {
        e.printStackTrace();
    }
}

}
```

CLASSPATH

Set CLASSPATH ...

```
$ export CLASSPATH=./home/lab/hibernate-1.2/hibernate.jar  
$ export CLASSPATH=$CLASSPATH:/home/lab/hibernate-1.2/lib/commons-  
lang.jar
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/hibernate-1.2/lib/commons-  
logging.jar
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/hibernate-1.2/lib/commons-  
collections.jar
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-  
bin.jar
```

Compile

Compile dengan javac ...

```
$ javac app/EmployeeApp.java
```

Launch

Launch dengan java ...

```
$ java app.EmployeeApp
```