Akses Database melalui Castor

Eko Budhi Suprasetiawan
ekobs@developerforce.net

(*) Kerjasama IlmuKomputer.Com dan DeveloperForce.Net

Motivasi

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

Persiapan

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

Dapatkan Castor library
Di sini dipakai : castor-0.9.5.2.jar

Deskripsi

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

Langkah

1. Tulis EmployeeBean.java
2. Tulis database.xml
3. Tulis mapping.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 /home/lab/myjava

database.xml

```
<?xml version="1.0"?>

<!DOCTYPE database PUBLIC "-//EXOLAB/Castor JDO Configuration DTD
Version 1.0//EN"
    "http://castor.exolab.org/jdo-conf.dtd">

<database name="mydatabase">
      <driver url="jdbc:mysql://localhost:3306/MYAPP_DB" class-
name="org.gjt.mm.mysql.Driver">
        <param name="user" value="ekobs"/>
        <param name="password" value="j2ee"/>
    </driver>
      <mapping  href="mapping.xml"/>
</database>
```

Mapping
Simpan di /home/lab/myjava

mapping.xml

```
<?xml version="1.0"?>

<!DOCTYPE  database PUBLIC "-//EXOLAB/Castor Object Mapping DTD
Version 1.0//EN"
   "http://castor.exolab.org/mapping.dtd">

<mapping>
      <class name="employee.EmployeeBean" identity="id">
            <map-to table="EMPLOYEE_TBL"/>
            <field name="id" type="string">
                  <sql name="ID" type="char"/>
            </field>
      <field name="name" type="string">
            <sql name="NAME" type="char"/>
      </field>
      <field name="department" type="string">
            <sql name="DEPARTMENT" type="char"/>
      </field>
      <field name="jobTitle" type="string">
            <sql name="JOB_TITLE" type="char"/>
      </field>
      <field name="hireDate" type="date">
            <sql name="HIRE_DATE" type="date"/>
      </field>
      <field name="permanentEmployee" type="boolean">
            <sql name="PERMANENT_EMPLOYEE" type="integer"/>
      </field>
      <field name="salary" type="double">
            <sql name="SALARY" type="double"/>
      </field>
      </class>
</mapping>
```

Client App

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

app/EmployeeApp.java

```
package app;

import java.text.SimpleDateFormat;
import java.util.Date;
import org.exolab.castor.jdo.*;
import org.exolab.castor.mapping.*;

import employee.*;

public class EmployeeApp
{
    public static void main(String[] args)
    {
            try
            {
                  JDO.loadConfiguration("database.xml");
                  JDO jdo = new JDO("mydatabase");


                  EmployeeBean bean = new EmployeeBean();
                  bean.setId("032");
                  bean.setName("Muhammad Sigit Arifianto");
                  bean.setDepartment("Business Development");
                  bean.setJobTitle("Account Manager");
                  bean.setHireDate(new 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();

                  Database db = jdo.getDatabase();
                  db.begin();
                  db.create(bean);
                  db.commit();
                  db.close();

                  db = jdo.getDatabase();
                  db.begin();
                  OQLQuery query = db.getOQLQuery("SELECT o from
employee.EmployeeBean o where id=$1");
                  query.bind("032");
                  QueryResults results = query.execute();

                  bean = (EmployeeBean) results.next();

                  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(MappingException e)
        {
            e.printStackTrace();
        }
        catch(PersistenceException e)
        {
            e.printStackTrace();
        }
    }
}
```

CLASSPATH

Set CLASSPATH …

```
$ export CLASSPATH=.:/home/lab/myjava/castor-0.9.5.2.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/jakarta-tomcat-
4.1.18/common/lib/jta.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/xerces-1_4_4/xerces.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
```