Akses Database Melalui Jaxor

Eko Budhi Suprasetiawan
ekobs@developerforce.net

(*) Kerjasama IlmuKomputer.Com dan DeveloperForce.Net

Motivasi

Jaxor bisa meng-generate Java Bean sesuai definisi dalam file XML. Dan juga meng-generate JDBC code.

Persiapan

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

Dapatkan Jaxor library
Extract ke : /home/lab/jaxor-3.3

Deskripsi

Sebuah Java Bean EmployeeBean akan disimpan/dibaca dari table EMPLOYEE_TBL di database. employee.jaxor digunakan untuk memetakan antara property dari EmployeeBean dan column di EMPLOYEE_TBL.

Langkah

1. Tulis build.xml
2. Tulis mapper.registry
3. Tulis employee.jaxor
4. Jalankan Ant
5. Tulis EmployeeApp.java
6. Set CLASSPATH
7. Compile

8. Launch

build.xml
Simpan di sub directory employee di bawah /home/lab/myjava

build.xml

```
<project name="myproject" default="jaxor">
      <property name="debug" value="true"/>

      <target name="setup">
            <mkdir dir="java"/>
      </target>

  <path id="jaxor.class.path">
    <fileset dir="/home/lab/jaxor-3.3/lib">
      <include name="**/jaxor.jar"/>
        <include name="**/qdox-1.2.jar"/>
      <include name="**/velocity-1.3.1.jar"/>
      <include name="**/velocity-dep-1.3.1.jar"/>
    </fileset>
  </path>

      <target name="jaxor" depends="setup">
            <taskdef name="jaxor"
                  classpathref="jaxor.class.path"
                  classname="net.sourceforge.jaxor.parser.AntTask"/>
            <jaxor  destdir="java" mappers="mapper.registry">
                  <fileset dir=".">
                  <include name="*.jaxor"/>
                  </fileset>
            </jaxor>
      </target>
</project>
```

Mapper
Simpan di /home/lab/myjava

mapper.registry

```
String=net.sourceforge.jaxor.mappers.StringMapper
Long=net.sourceforge.jaxor.mappers.LongMapper
Boolean=net.sourceforge.jaxor.mappers.BooleanMapper
java.sql.Timestamp=net.sourceforge.jaxor.mappers.TimestampMapper
java.util.Date=net.sourceforge.jaxor.mappers.DateMapper
```

Mapping
Simpan di /home/lab/myjava

employee.jaxor

```
<jaxor package="employee">
      <entity name="EMPLOYEE_TBL">
        <primary-key>
            <attribute name="ID" type="String" />
        </primary-key>
      <attribute name="NAME" type="String"/>
      <attribute name="DEPARTMENT" type="String"/>
      <attribute name="JOB_TITLE" type="String"/>
      <attribute name="HIRE_DATE" type="java.util.Date"/>
      <attribute name="PERMANENT_EMPLOYEE" type="Boolean"/>
      <attribute name="SALARY" type="Long"/>
      </entity>
</jaxor>
```

Ant

Jalankan Ant dari /home/lab/myjava

```
$ ant
Buildfile: build.xml

setup:
    [mkdir] Created dir: /home/lab/myjava/java

jaxor:
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblFinder.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblImpl.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblEntity.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblList.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblIterator.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblFinderImpl.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblMetaRow.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblEntityRow.java
    [jaxor] Generating
/home/lab/myjava/java/employee/EmployeeTblResultSet.java

BUILD SUCCESSFUL
Total time: 4 seconds
```

Client App

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

app/EmployeeApp.java

```
package app;

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

import net.sourceforge.jaxor.*;
import net.sourceforge.jaxor.api.*;
import net.sourceforge.jaxor.db.*;

import employee.*;

public class EmployeeApp
{
    public static void main(String[] args)
    {
            try
            {
                  String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd  = "j2ee";
            Connection conn = DriverManager.getConnection(
                    url, user, pwd);

                  ConnectionFactory factory = new
SingleConnectionFactory(conn);

                  JaxorContextImpl context = new
JaxorContextImpl(factory);
                  EmployeeTblFinderImpl finder = new
EmployeeTblFinderImpl(context);

                  EmployeeTblEntity bean = finder.newInstance("032");

                  bean.setName("Muhammad Sigit Arifianto");
                  bean.setDepartment("Business Development");
                  bean.setJobTitle("Account Manager");
                  bean.setHireDate(new java.sql.Date(new
java.util.Date().getTime()));
                  bean.setPermanentEmployee(new Boolean(false));
                  bean.setSalary(new Long(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().booleanValue());
            System.out.println("Salary : " +
bean.getSalary().longValue());
            System.out.println();

                  context.commit();
                  context.end();


            conn = DriverManager.getConnection(
                    url, user, pwd);
                  factory = new SingleConnectionFactory(conn);

                  context = new JaxorContextImpl(factory);
                  finder = new EmployeeTblFinderImpl(context);

                  bean = finder.selectByPrimaryKey("032", false);

                  context.commit();
                  context.end();

                  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(ClassNotFoundException e)
        {
            e.printStackTrace();
        }
            catch(SQLException e)
        {
            e.printStackTrace();
        }
    }
}
```

CLASSPATH

Set CLASSPATH …

```
$ export CLASSPATH=java/
$ export CLASSPATH=$CLASSPATH:/home/lab/jaxor-3.3/lib/jaxor.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/jaxor-3.3/lib/qdox-1.2.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/jaxor-3.3/lib/velocity-1.3.1.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/jaxor-3.3/lib/velocity-dep-1.3.1.jar

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

Compile

Compile dengan javac …

```
$ javac -sourcepath java/ -d java/ java/app/EmployeeApp.java
```

Launch

Launch dengan java …

```
$ java app.EmployeeApp
```