NewbieGuideToOJB2

Page title (cookbook approach, tutorial , ...)

- TARGET-AUDIENCE: ***beginner*** advanced expert
- COCOON-RELEASES: 2.0.3, 2.0.4
- DOCUMENT-STATUS: ***draft*** reviewed released

===Introduction===

This is copy&paste from http://joose.iki.fi/ojb/

This page is my setup and adventure to get cforms work with OJB and 1:n relationships. After all this is working, this could be used as a template to create tutorial / something to help others to do the same thing.

Status

OJB stuff seems to work ok, because the test-script executes ok.

UPDATE works when I added hidden field with id see template.

DELETE works.

Everything is working

TODO

Database

```
CREATE TABLE parent (
id SERIAL,
name VARCHAR(50), PRIMARY KEY (id));

CREATE TABLE childs (
id SERIAL,
parent_id INTEGER NOT NULL,
name VARCHAR(20), PRIMARY KEY(id), FOREIGN KEY(parent_id) REFERENCES parent (id));
```

So we have one-to-many (1:n) connection between parent and child.

So if we add new records, we first have to add record to table A and afterthat records to table childs.

Java classes

parent

```
package net.vettenranta;

import java.util.ArrayList; import java.util.Collection;

import java.io.Serializable;

public class Parent implements Serializable {
private int id;
private Collection childs = new ArrayList ();
private String name;

public int getId () { return id; }
public void setId (int id) { this.id = id; }

public Collection getChilds () { return childs; }
public void setChilds (Collection childs ) { this.childs = childs; }
public void addChild (Child child) {
child.setParent (this);
this.childs.add (child);
}
public String getName () { return name; }
public void setName (String name) { this.name = name; }
}
```

child

```
package net.vettenranta;

import java.io.Serializable;
```

```
public class Child implements Serializable {
private int id;
private Parent parent;
private String name;

public int getId () { return id; }
public void setId (int id) { this.id = id; }

public Parent getParent () { return parent; }
public void setParent (Parent parent) { this.parent = parent; }

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

DAO

```
package net.vettenranta;

import java.util.Iterator;

import javax.jdo.PersistenceManager; import javax.jdo.Transaction;

import org.apache.cocoon.ojb.jdo.components.JdoPMF; import org.apache.ojb.broker.Identity; import org.apache.ojb.broker.PersistenceBroker; import org.
apache.ojb.broker.PersistenceBrokerFactory; import org.apache.ojb.broker.util.ObjectModificationDefaultImpl;

public class DAO {
// requires lot's of comments
public Parent retrieve(Parent bean, JdoPMF pmf) {
PersistenceManager persistenceManager = pmf.getPersistenceManager();
PersistenceBroker broker = PersistenceBrokerFactory.defaultPersistenceBroker();
Query query = new QueryByIdentity(bean);
Parent result = (Parent) broker.getObjectByQuery(query);
broker.close();
return result;
}

public void insert(Parent bean, JdoPMF pmf) {
PersistenceManager persistenceManager = pmf.getPersistenceManager();
PersistenceBroker broker = PersistenceBrokerFactory.defaultPersistenceBroker();
broker.beginTransaction();
broker.store(bean);
broker.commitTransaction();
broker.close();
}

public void remove(Parent bean, JdoPMF pmf) {
PersistenceManager persistenceManager = pmf.getPersistenceManager();
PersistenceBroker broker = PersistenceBrokerFactory.defaultPersistenceBroker();
broker.beginTransaction();
broker.delete(bean);
broker.commitTransaction();
broker.close();
}

public void update(Parent bean, JdoPMF pmf) {
PersistenceManager persistenceManager = pmf.getPersistenceManager();
PersistenceBroker broker = PersistenceBrokerFactory.defaultPersistenceBroker();
Iterator it = bean.getChilds().iterator();

broker.beginTransaction();
broker.store(bean, ObjectModificationDefaultImpl.UPDATE);
broker.commitTransaction();
broker.close();
}

}
```

package.jdo

```
<?xml version="1.0"?>
<!DOCTYPE jdo SYSTEM "file:/javax/jdo/jdo.dtd">
```

```
<jdo>
<package name="net.vettenranta">
<class name="Parent" identity-type="datastore">
<field name="id" persistence-modifier="persistent">
<extension vendor-name="ojb" key="column" value="ID"/>
</field>
<field name="name" persistence-modifier="persistent">
<extension vendor-name="ojb" key="column" value="NAME"/>
</field>
<field name="childs" embedded="true">
<collection embedded-element="true" />
</field>
</class>
<class name="Child" identity-type="datastore">
<field name="id" persistence-modifier="persistent">
<extension vendor-name="ojb" key="column" value="ID"/>
</field>
<field name="parent" persistence-modifier="persistent">
<extension vendor-name="ojb" key="column" value="PARENT_ID"/>
</field>
<field name="name" persistence-modifier="persistent">
<extension vendor-name="ojb" key="column" value="NAME"/>
</field>
</class>
</package>
</jdo>
```

Enhancing the classes

I have untarred jdo to my homedirectory.

Manually

This will place enhanced binaries to output directory. It will create that directory.

```
$ export CLASSPATH=/jdo-1_0_1-ri/jdo.jar:/jdo-1_0_1-ri/jdori-enhancer.jar:~/jdo-1_0_1-ri/jdori.jar:.
$ java com.sun.jdori.enhancer.Main -d output package.jdo net/vettenranta/Parent.class net/vettenranta/Child.class
```

repository.xml

```
<descriptor-repository version="1.0" isolation-level="read-uncommitted">
<jdbc-connection-descriptor jcd-alias="test" default-connection="true" platform="PostgreSQL" subprotocol="postgresql">
<sequence-manager className="org.apache.ojb.broker.util.sequence.SequenceManagerNextValImpl"/>
</jdbc-connection-descriptor>
<class-descriptor class="net.vettenranta.Parent" table="PARENT">
<field-descriptor name="id" primarykey="true" nullable="false" default-fetch="true" autoincrement="true" column="ID" sequence-name="parent_id_seq"
jdbc-type="INTEGER"/>
<field-descriptor name="name" default-fetch="true" column="NAME" jdbc-type="VARCHAR"/>
<collection-descriptor name="childs" element-class-ref="net.vettenranta.Child" auto-retrieve="true" auto-delete="true" auto-update="true">
<inverse-foreignkey field-ref="parent_id"/>
</collection-descriptor>
</class-descriptor>
<class-descriptor class="net.vettenranta.Child" table="CHILDS">
<field-descriptor name="id" primarykey="true" nullable="false" default-fetch="true" column="ID" jdbc-type="INTEGER" autoincrement="true" sequence-
name="childs_id_seq" />
<field-descriptor name="parent_id" nullable="false" default-fetch="true" column="PARENT_ID" jdbc-type="INTEGER" access="anonymous"/>
<field-descriptor name="name" default-fetch="true" column="NAME" jdbc-type="VARCHAR"/>
<reference-descriptor name="parent" class-ref="net.vettenranta.Parent">
<foreignkey field-ref="parent_id" />
</reference-descriptor>
</class-descriptor>
</descriptor-repository>
```

Test-script (in cocoon flowscript)

This script works. So It's pretty safe to say that everything should be ok.

```
function test () {
var factory = cocoon.getComponent(Packages.org.apache.cocoon.ojb.jdo.components.JdoPMF.ROLE);
var bean = new Packages.net.vettenranta.Parent ();
var child;
var dao = new Packages.net.vettenranta.DAO ();
var id;
var iterator;
// 1.
// let's make a new bean with 1 child
bean.setName ('Dad 1');
child = new Packages.net.vettenranta.Child ();
child.setName ('Child 1');
bean.addChild (child);

// 2.
// let's save it
dao.insert (bean, factory);

// 3.
// let's retrieve it from database
id = bean.getId();
bean = new Packages.net.vettenranta.Parent ();
bean.setId(id);
bean = dao.retrieve(bean, factory);

iterator=bean.getChilds().iterator();

while (iterator.hasNext()) {
child = iterator.next();
child.setName ('Child XXX');
}

// 4.
// let's modify the bean

bean.setName ('Dad 2');
child = new Packages.net.vettenranta.Child ();
child.setName ('Child 2');
bean.addChild (child);

// let's save it to the database
dao.update (bean, factory);

// and remove it from database
dao.remove (bean, factory);

cocoon.releaseComponent(factory);

cocoon.redirectTo ("add.html");
}
```

CForms flowscript

```
bean.setId (id);
bean = dao.retrieve (bean, factory);
form.load(bean);
form.showForm("forms/edit.html"); // EVERYTHING OK SO FAR
form.save(bean);
dao.update (bean, factory);
```

CForms binding

```
<fb:context xmlns:fb="http://apache.org/cocoon/forms/1.0#binding" path="/" >

<fb:value id="id" path="id" direction="load"/>
<fb:value id="name" path="name" />

<fb:repeater id="childs"
parent-path="."
row-path="childs">

<fb:identity>
<fb:value id="id" path="@id"/>
</fb:identity>

<fb:on-bind>
<fb:value id="id" path="id"/>
<fb:value id="name" path="name"/>
</fb:on-bind>
```

```
<fb:unique-row>
<fb:unique-field id="id" path="id"/>
</fb:unique-row>

<fb:on-delete-row>
<fb:delete-node />
</fb:on-delete-row>

<fb:on-insert-row>
<fb:insert-bean
classname="net.vettenranta.Child"
addmethod="addChild"/>
</fb:on-insert-row>
</fb:repeater>
</fb:context>
```

CForms forms

```
<fd:form
xmlns:fd="http://apache.org/cocoon/forms/1.0#definition"
xmlns:i18n="http://apache.org/cocoon/i18n/2.1">

<fd:widgets>
<fd:field id="id">
<fd:datatype base="integer" />
</fd:field>
<fd:field id="name">
<fd:datatype base="string" />
<fd:label>Name</fd:label />
</fd:field>

<fd:repeater id="childs">
<fd:widgets>

<fd:field id="id">
<fd:datatype base="integer" />
</fd:field>

<fd:field id="name" required="true">
<fd:datatype base="string">
<fd:validation>
<fd:length min="3"/>
</fd:validation>
</fd:datatype>
<fd:label>Name</fd:label>
</fd:field>

<fd:booleanfield id="select">
<fd:label>Select</fd:label>
</fd:booleanfield>

</fd:widgets>
</fd:repeater>
<fd:repeater-action id="addchild" action-command="add-row" repeater="childs">
<fd:label>New child</fd:label>
</fd:repeater-action>

<fd:repeater-action id="removechild" action-command="delete-rows" repeater="childs" select="select">
<fd:label>Remove child</fd:label>
</fd:repeater-action>
</fd:widgets>
</fd:form>
```

CForms template

CForms template file has to have (in row-repeater)

```
<fi:widget id="id"><fi:styling type="hidden"><fi:widget>
```

Software

ojb 1.0.rc6 jdo 1.0.1-ri (sun) IBM Java2 1.4.1 Cocoon 2.1.5.1 Postgresql 7.3.4-3.rhl9 postgresql-jdbc-7.3.4-3.rhl9 redhat 9

---

page metadata

- AUTHOR: JooseVettenranta
- AUTHOR-CONTACT: joose@iki.fiBR

- REVIEWED-BY:BR
- REVIEWER-CONTACT:BR