JPA – Java Persistence API

JPA oferă facilități ORM (Object/Relational Mapping) pentru a gestiona date relaționate în aplicații Java, în principal pentru a memora datele pe un suport persistent (de obicei bază de date relațională) şi pentru a efectua interogări asupra datelor persistate. Java Persistence cuprinde 4 aspecte:

- Java Persistence API.
- Limbajul de interogare.
- Java Persistence Criteria API
- Metadate de mapare

Unitatea elementară care se persistă se numeşte *entitate*. De obicei o entitate este asociată cu un tabel din baza de date, o linie (înregistrare) din tabel fiind o instanță a entității. O entitate are câmpuri sau proprietăți persistente, care sunt asociate prin mapări cu elementele specifice suportului de stocare.

O clasă trebuie să îndeplinească anumite condiții pentru a fi entitate, ea va fi în plus marcată cu adnotarea *@Entity*.

Se pot impune anumite validări prin adnotări specifice. Astfel, un câmp obligatoriu se marchează cu *@NotNull* iar un format particular de introducere este precizat prin *@Pattern(regexp="...")*.

Entitățile posedă elemente unice de identificaer numite chei primare, care pot fi simple (un câmp) sau compuse (mai mutle câmpuri).

Relațiile dintre entități pot fi de următoarele multiplicități:

- One-to-one, în care o instanță a unei entități este legată (relaționată) cu o singură instanță a altei entități.
- One-to-many, în care o instanță a unei entități poate fi legată (relaționată) la mai multe instanțe ale celeilalte entități. Exemplul clasic este cel al unei comenzi (order) care poate avea mai multe linii.
- Many-to-one, este relația inversă față de one-to-many, când mai multe instanțe ale unei entități pot fi legate (relaționate) la o instanță a celeilalte entități.
- Many-to-many, când instanțele unei entități pot fi legate la mai multe instanțe ale celeilalte entități. De exemplu, un curs are mai mulți studenți şi fiecare student urmează mai multe cursuri.

Relațiile pot fi:

- Bidirecționale, în care fiecare entitate are un câmp prin care se referă la cealaltă entitate (fiecare entitate "ştie" de cealaltă).
- Unidirecționale, în care doar o entitate "ştie" de cealaltă.

Aceste relații crează anumite dependențe, care impun anumite reguli la realizarea unor operații (ştergere).

Entitățile sunt gestionate de un manager de entități, o instanță a javax.persistence.EntityManager, care este asociat cu un context de persistență (o mulțime de entități care există într-un anumit depozit de informații).

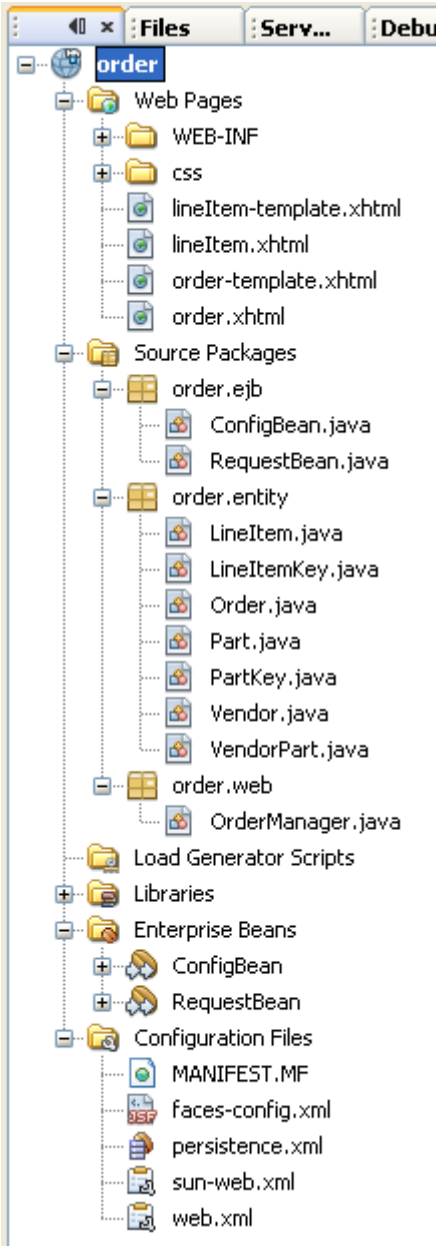
Instanțele de entități pot fi în dun din stările următoare:

- New
- Managed
- Detached
- Removed

1. Exemplul 1: "order"

Deschideți în NetBeans proiectul "order", care se găseşte în tutorialul JEE6 (examples/persistence) sau în arhiva cu surse de pe site. Aceasta este o aplicație simplă de gestiune a unor produse şi de realizare de comenzi pe aceste produse.

Analizați fişierele componente şi modul lor de organizare. Identificați entitățile aplicației (produs, furnizor, comandă, linie de comandă).



Identificați relațiile dintre entități și multiplicitățile acestora.

Rulați proiectul din NetBeans, identificând pagina de pornire (în fișierul de configurare).

Identificați și analizați cele 2 șabloane folosite. Aplicați modificările necesare pentru a schimba aspectul implicit al paginii de plecare:

Order Java Persistence Example DAI

View All Orders

Order ID	Shipment Info	Status	Last Updated	Discount	Actions
1111	333 New Court, New City, CA 90000	N	Fri Nov 12 02:39:19 EET 2010	10%	Delete
4312	333 New Court, New City, CA 90000	N	Fri Nov 12 02:39:19 EET 2010	0%	Delete

Create New Order

Order ID:
Shipment Info:
Status: Complete
Discount: 0 %
Submit

Find Vendors

Search for Vendors: Find Vendor

Vendor

Adăugați o comandă nouă, completând valori corecte în câmpurile comenzii:

Create New Order

Order ID: 1234
Shipment Info: abcd
Status: Complete
Discount: 0 %
Submit

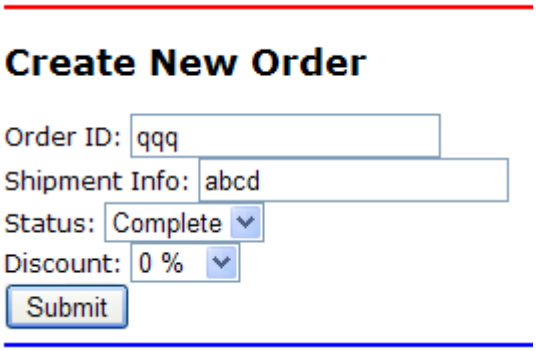
Observați efectul imediat:

Order Java Persistence Example DAI

View All Orders

Order ID	Shipment Info	Status	Last Updated	Discount	Actions
1111	333 New Court, New City, CA 90000	N	Fri Nov 12 02:39:19 EET 2010	10%	Delete
1234	abcd	Y	Fri Nov 12 03:26:23 EET 2010	0%	Delete
4312	333 New Court, New City, CA 90000	N	Fri Nov 12 02:39:19 EET 2010	0%	Delete

Adăugați o comandă nouă, completând valori incorecte în câmpurile comenzii:



Identificați de ce nu se adaugă înregistrarea dorită. Verificați unde se semnalează problemele.

Identificați cum se face "încărcarea" inițială a datelor.

Adăugați un buton de ştergere produs de pe comandă.

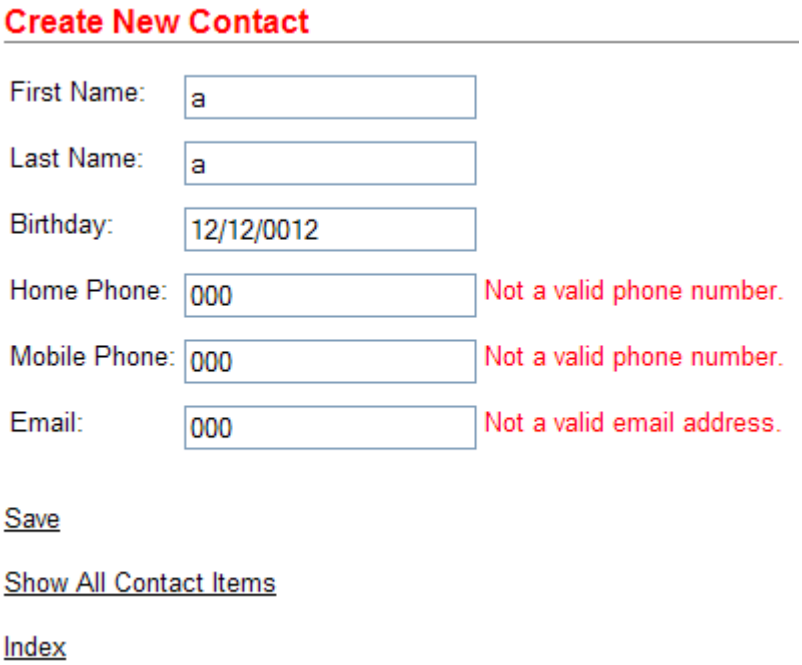
Adăugați un atribut nou pe comandă, numărul de produse conținute. Acesta va fi (re)calculat automat la adăugarea sau ştergerea unui produs.

2. Exemplul 2: "address-book"

Deschideți în NetBeans proiectul "address-book", care se găseşte în tutorialul JEE6 (examples/persistence) sau în arhiva cu surse de pe site. Aceasta este o aplicație simplă de gestiune a unei liste de contacte.

Analizați fişierele componente şi modul lor de organizare. Identificați entitățile aplicației (contact).

Modificați mesajele afişate la introducerea incorectă a datelor.



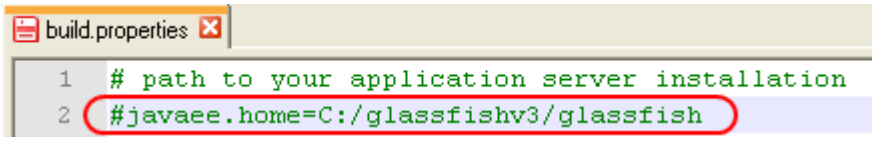
Adăugați un câmp nou pe entitate, care să memoreze ocupația persoanei şi propagați acest câmp în toate paginile necesare.

3. Exemplul 3: "advancedMapping"

Deschideți în NetBeans proiectul "advancedMapping", care se găseşte în exemplele JPA (https://glassfish-samples.dev.java.net/servlets/ProjectDocumentList?folderID=5214&expandFolder=5214&folderID=0) sau în arhiva cu surse de pe site. Descrierea acestei aplicații se găseşte la adresa https://glassfish-samples.dev.java.net/source/browse/*checkout*/glassfish-samples/tags/JAVAEE6_SAMPLES-0_9-b16/ws/javaee6/jpa/advancedMapping/docs/index.html.

Identificați entitățile acestei aplicații.

Comentați linia marcată mai jos din fișierul bp-project\build.properties:



Rulați aplicația, mai întâi direct, apoi pas cu pas (depanare).

4. Exemplul 4: "criteriaQuery"

Deschideți în NetBeans proiectul "criteriaQuery", care se găsește în exemplele JPA (https://glassfish-samples.dev.java.net/servlets/ProjectDocumentList?folderID=5214&expandFolder=5214&folderID=0) sau în arhiva cu surse de pe site. Descrierea acestei aplicații se găsește la adresa https://glassfish-samples.dev.java.net/source/browse/*checkout*/glassfish-samples/tags/JAVAEE6_SAMPLES-0_9-b16/ws/javaee6/jpa/criteriaQuery/docs/index.html.

Identificați entitățile acestei aplicații.

Rulați aplicația, mai întâi direct, apoi pas cu pas (depanare).

Identificați diferențele față de cele din Exemplul 4: "criteriaQuery".

5. Resurse

Resurse utile:

- http://download.oracle.com/javaee/6/tutorial/doc/
- http://download.oracle.com/javaee/6/tutorial/doc/bnbpy.html
-
- http://download.oracle.com/javaee/6/javaserverfaces/2.0/docs/pdldocs/facelets/h/tld-summary.html
- http://www.coreservlets.com/JSF-Tutorial/jsf1/
- http://www.coreservlets.com/JSF-Tutorial/jsf2/
- https://javaeetutorial.dev.java.net/servlets/ProjectDocumentList
- http://download.oracle.com/javaee/6/javaserverfaces/2.0/docs/pdldocs/facelets/
- http://netbeans.org/kb/docs/web/jsf20-intro.html