

**Master Ingénierie de Données et  
Développement Logiciel**  
*Architecture et Framework JEE*

**TP4 : Java Persistence API pour interagir  
avec les bases de données**

**REALISER PAR :**  
**YOUNES MAHMAR**

# Objectif de l'atelier

Dans cet atelier, on va aborder les sujets suivants:

- Créer une première entité JPA
- Interagir les entités JPA en utilisant EntityManager
- Génération de formulaires JSF à partir des entités JPA
- Générer des entités JPA à partir d'un schéma existant d'une base de données
- requêtes JPA nommées et JPQL
- relations entre les entités
- Générer des applications compètes JSF à partir des entités JPA

## Définition

Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.

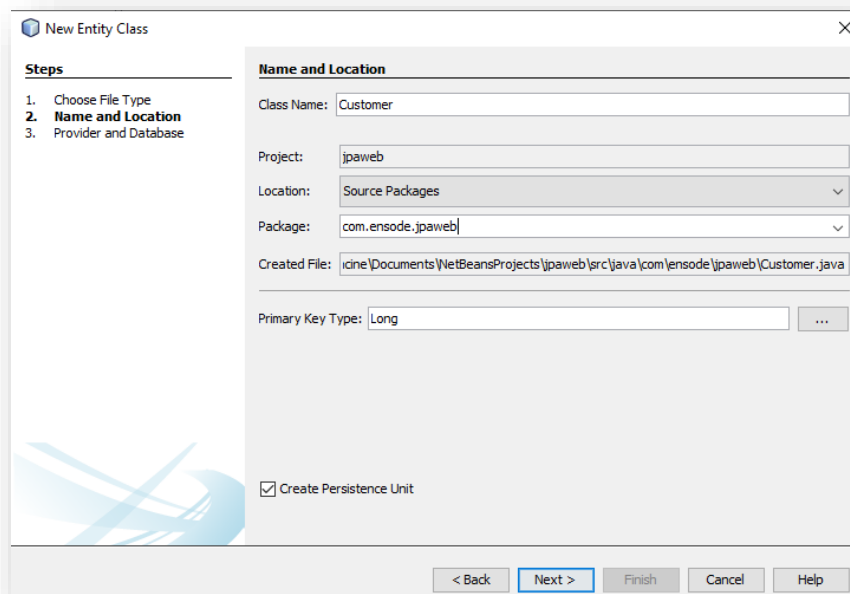
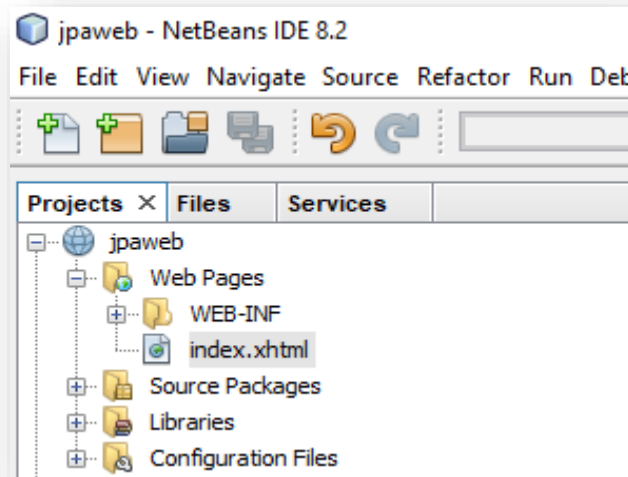
Java Persistence API repose essentiellement sur l'utilisation des annotations, introduites dans Java 5. Elles permettent de définir facilement des objets métier, qui pourront servir d'interface entre la base de données et l'application, dans le cadre d'un mapping objet-relationnel.

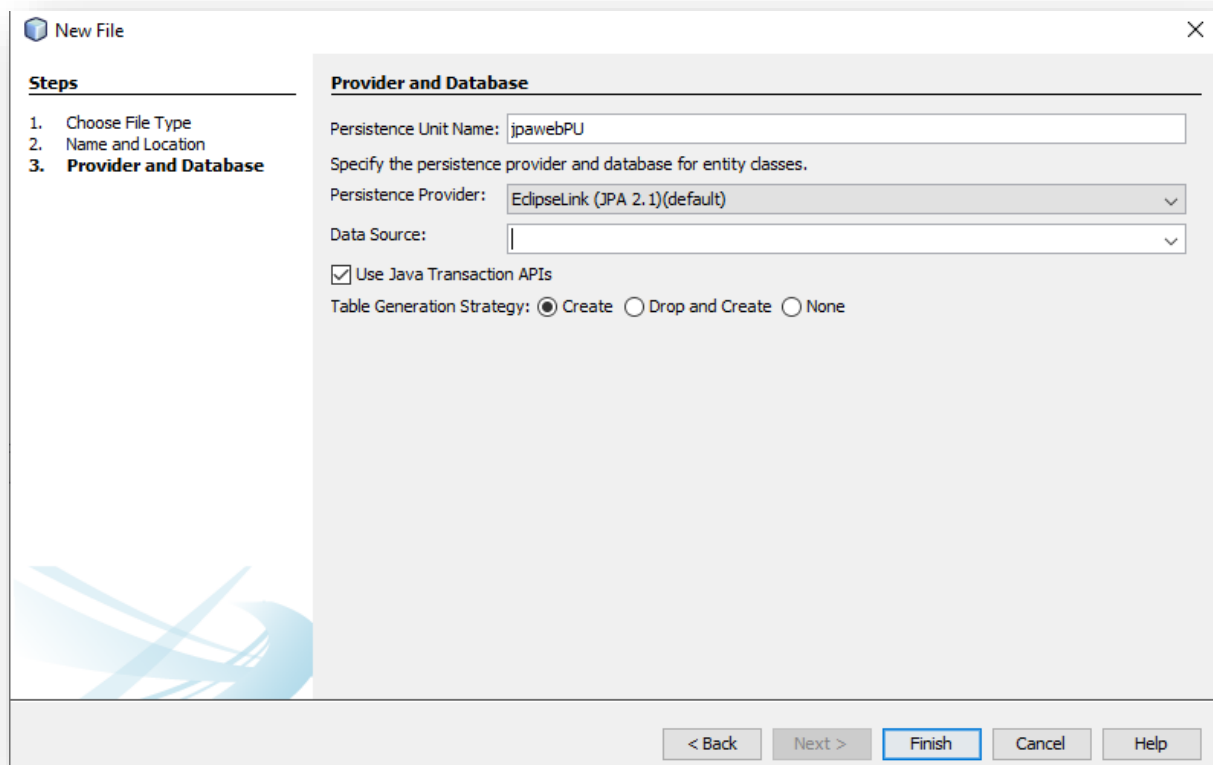
## 1 Créer notre première entité JPA

**Entité JPA** : Les entités JPA sont des classes Java dont les champs sont conservés dans une base de données par l'API JPA. Les entités JPA sont des POJOs (Plain Old Java Objects), en tant que tels, ils n'ont pas besoin d'hériter d'aucune classe spécifique ou d'implémenter aucune interface spécifique. Une classe Java est désignée comme étant une entité JPA avec l'annotation **@Entity**

**La première étape** à faire est créé un nouveau projet base sure le Framework serverface.

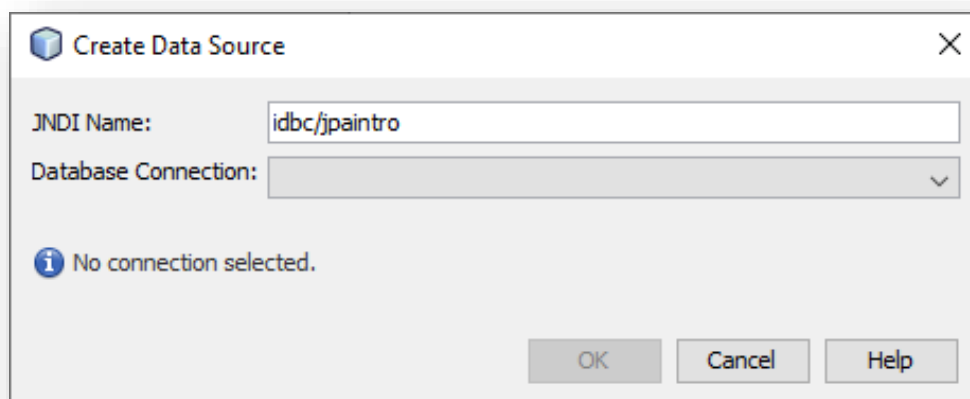
**Deuxième étape** crée un class java de type entity class, cette première class nous permet de créer une persistance pour notre application et le connecter avec une base de données de plus.

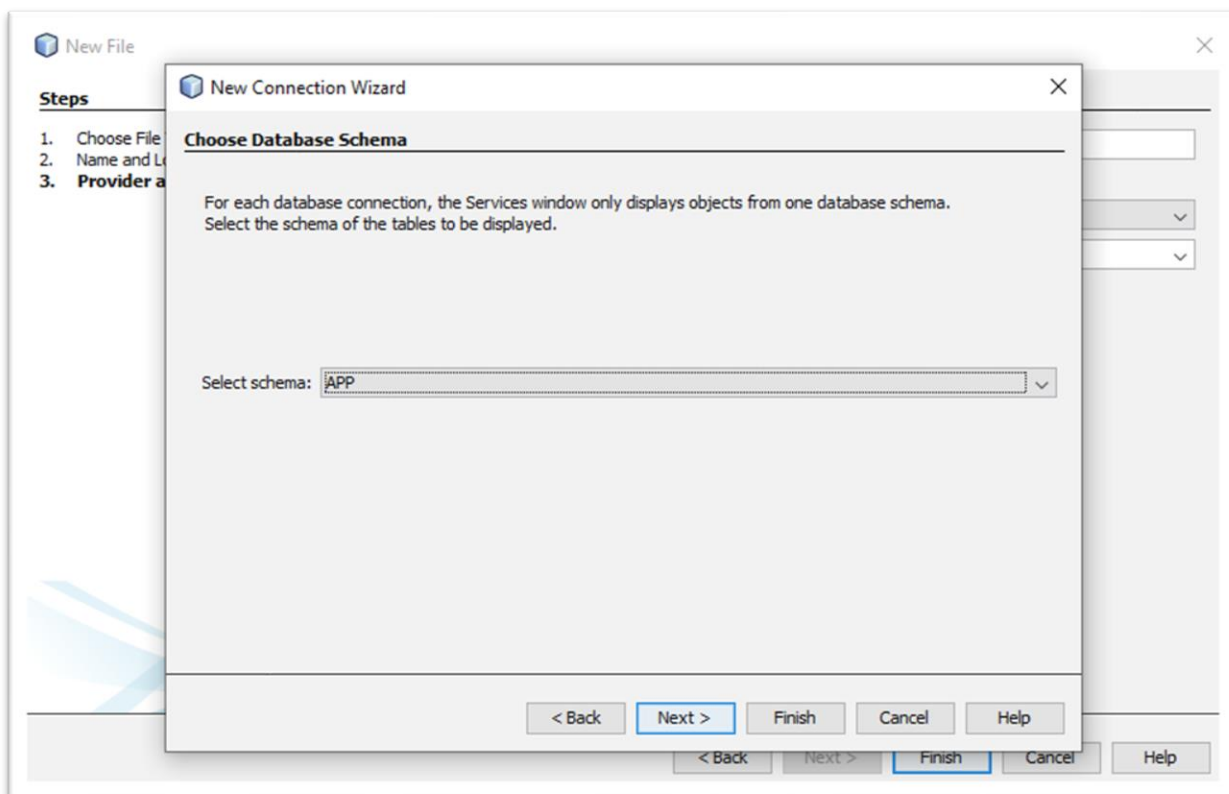
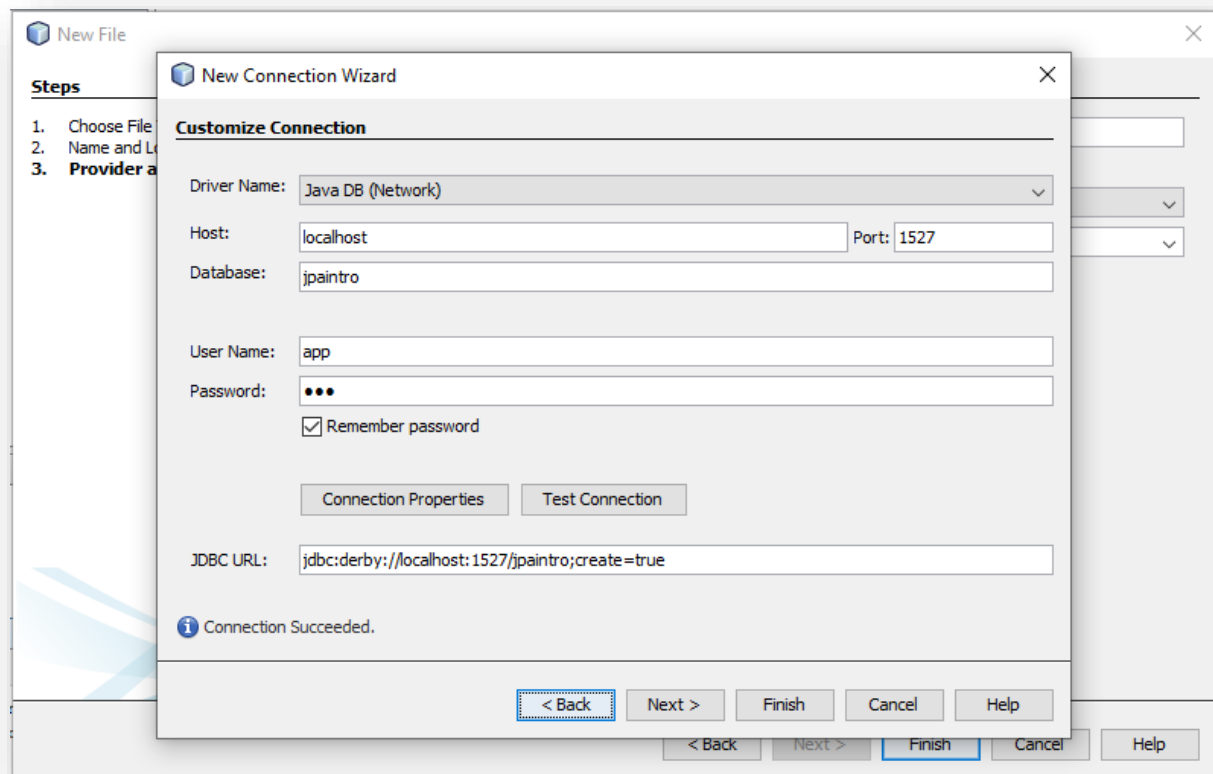


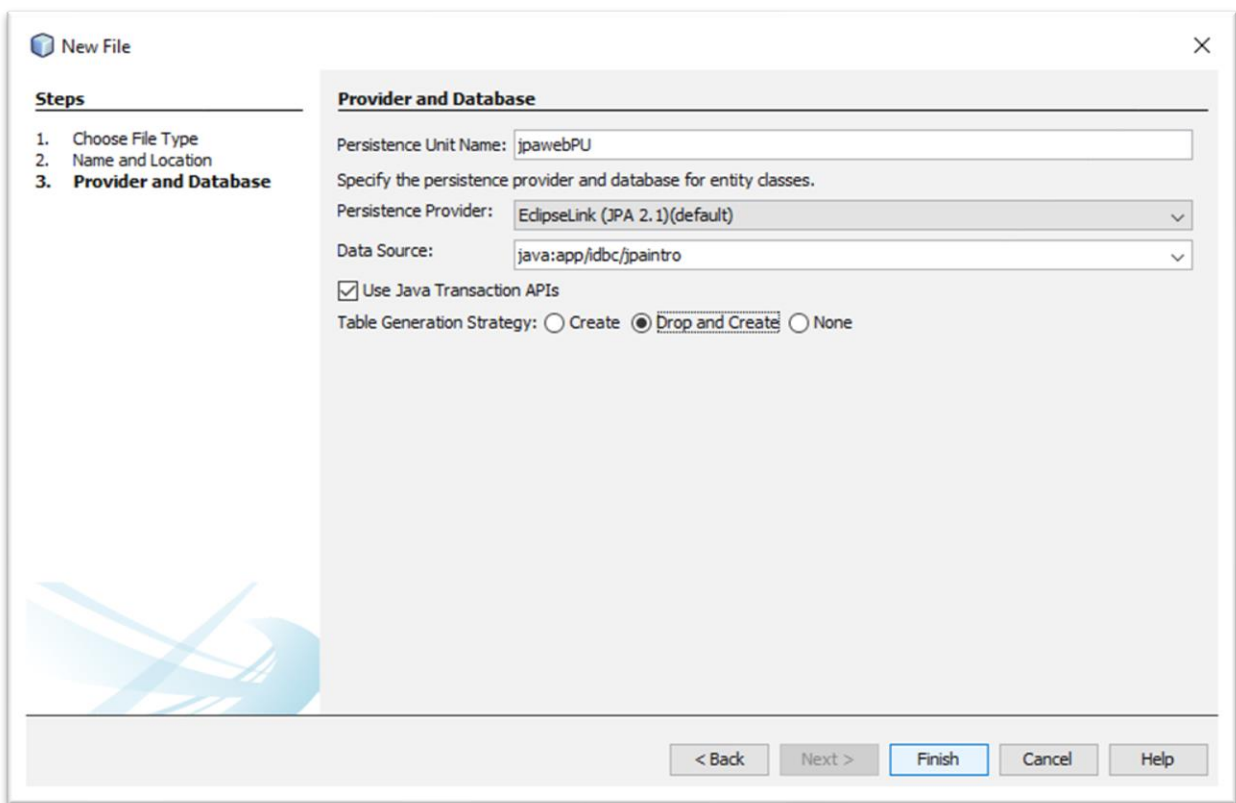


Dans cette étape, on doit sélection notre basse de donne, dans notre cas on va créer une nouvelle base de données vide.

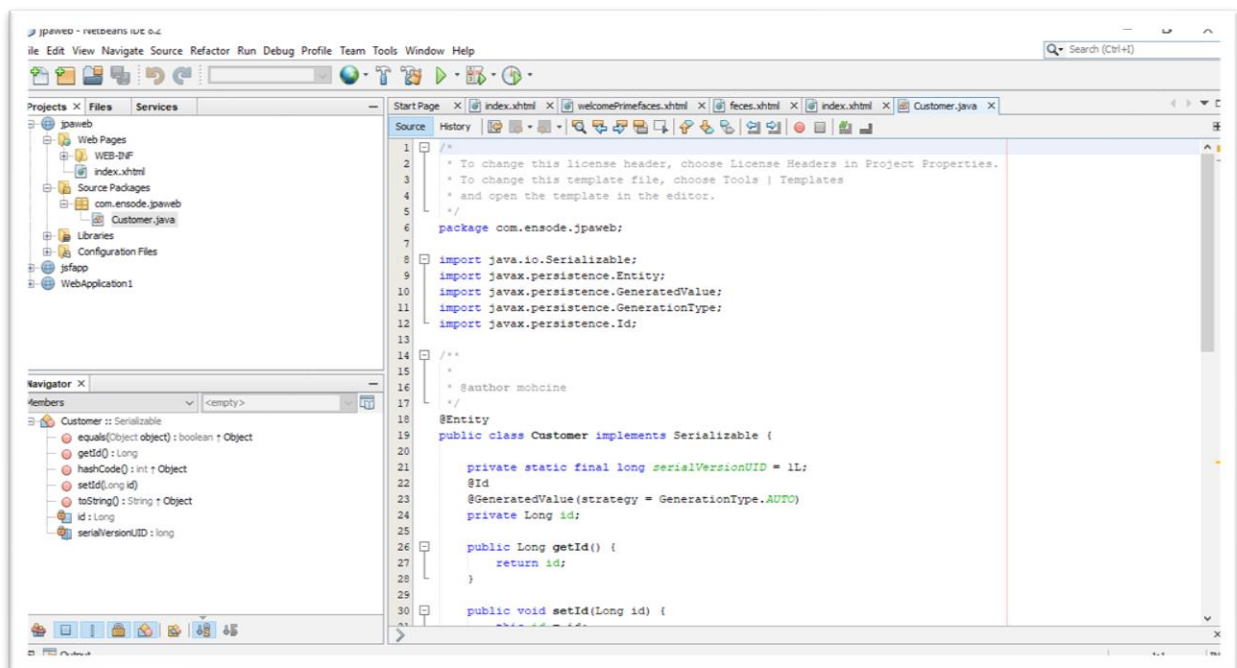
Dans les étapes suivantes on va créer notre base de donnes (choisir le driver, port, username ...).







Dans cette étape il est recommandé de sélectionner Drop and create dans les transactions Apis.



Et voilà enfin notre class.

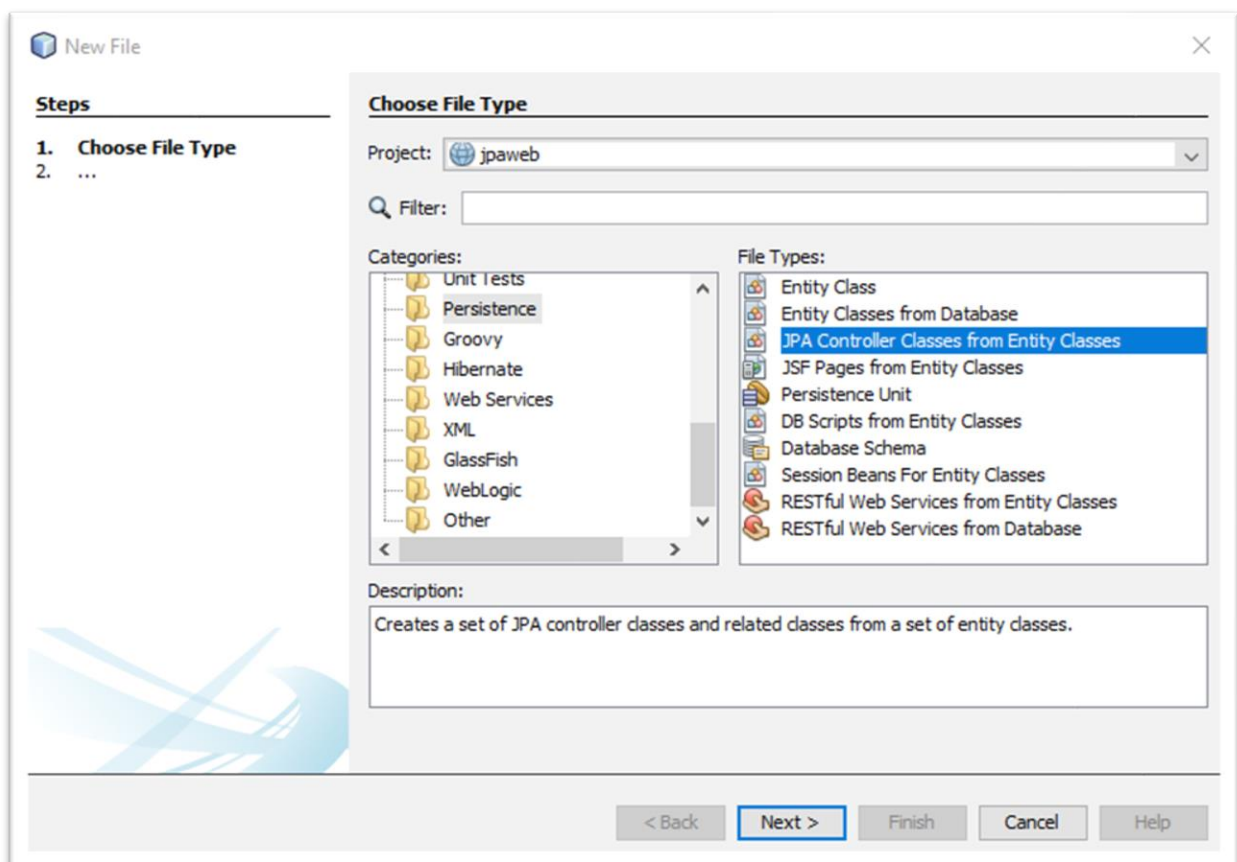
En remarque qu'il y a un ensemble d'annotation dans notre projet :

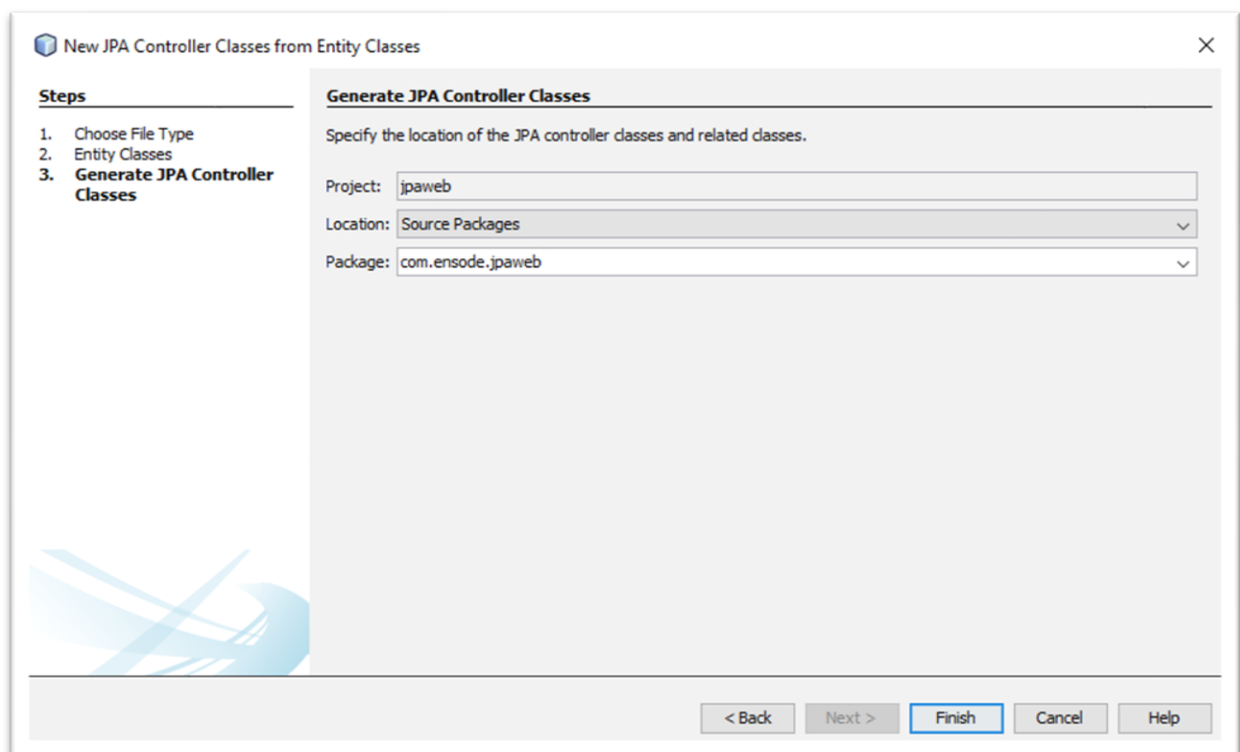
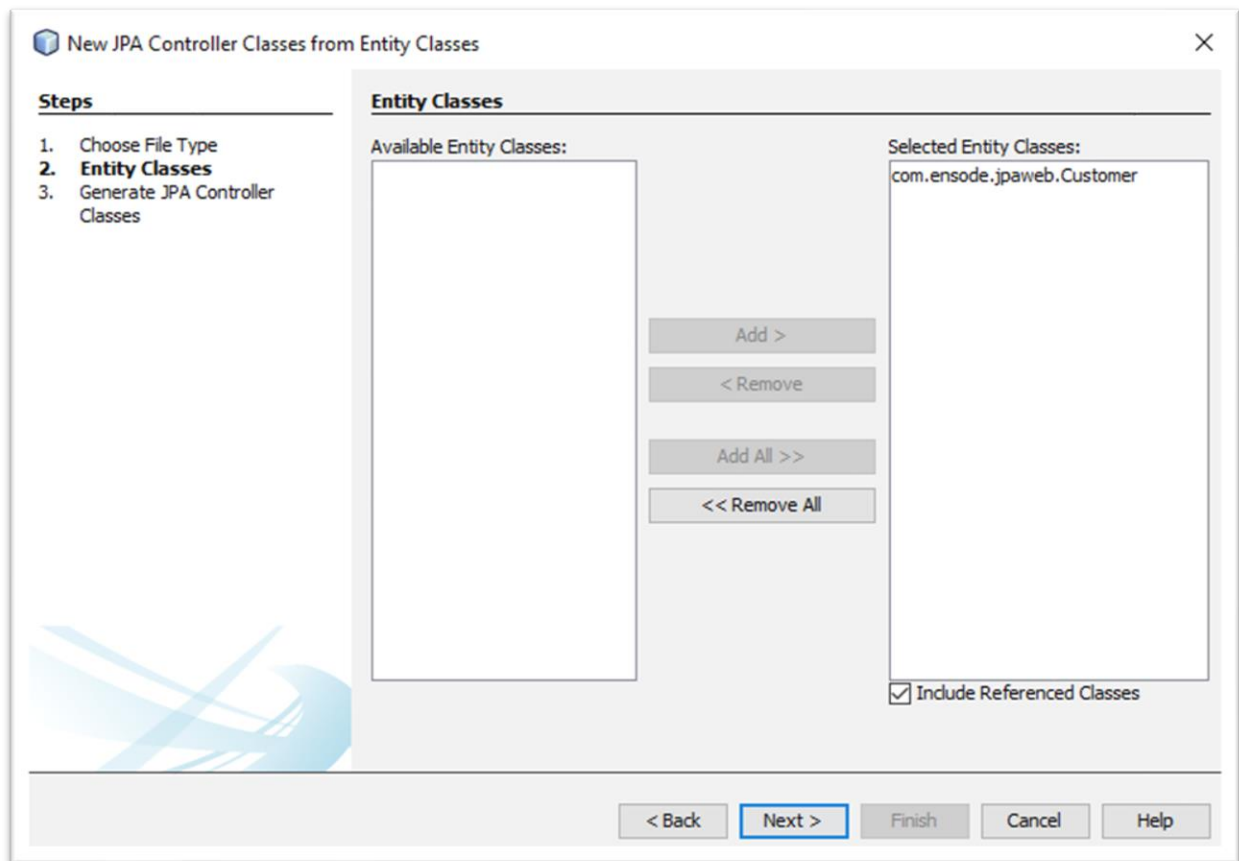
**@Entity** signifie que cette classe est une Entity classe.

**@id** et **@generatedValue** signifie que l'attribut qu'il va suivre est une clé primaire qui va être générée selon une stratégie, dans notre cas on a choisi de le générer automatiquement.

## 2 Création d'un DAO (Data Access Object)

Le design pattern Data Access Object (DAO) consiste à écrire du code qui interagit avec une base de données. Le DAO permet de garder toutes les fonctionnalités d'accès à la base de données dans les classes DAO. Ce qui permet de séparer les rôles, et libérer les autres couches (couche de présentation et couche métier) de toute logique de persistance.





### 3 Génération automatique des entités JPA



Create Java DB Database

Database Name: customerDb

User Name: app

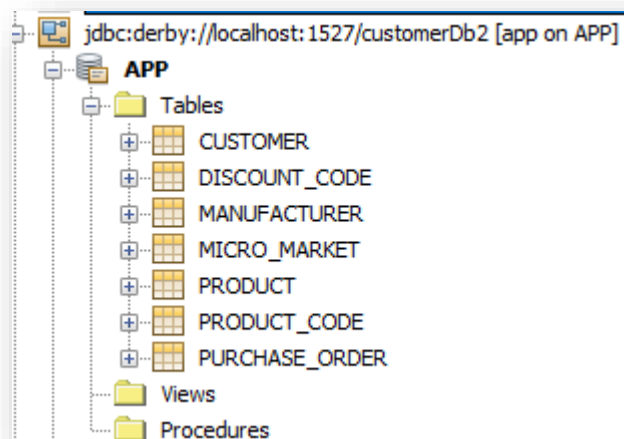
Password: ●●●

Confirm Password: ●●●

Database Location: C:\Users\mohcine\AppData\Roaming\NetBeans\Derby

Properties...

OK Cancel



Génération des entités à partir d'un schéma existant

New Entity Classes from Database

Steps

1. Choose File Type  
2. **Database Tables**  
3. Entity Classes  
4. Mapping Options

Database Tables

Data Source: jdbc/sample

Available Tables:

Any

Selected Tables:

CUSTOMER  
DISCOUNT\_CODE  
MANUFACTURER  
MICRO\_MARKET  
PRODUCT  
PRODUCT\_CODE  
PURCHASE\_ORDER

Add >

< Remove

Add All >>

<< Remove All

☒ Include Related Tables

< Back

Next >

Finish

Cancel

Help

New Entity Classes from Database

Steps

1. Choose File Type  
2. Database Tables  
3. **Entity Classes**  
4. Mapping Options

Entity Classes

Specify the names and the location of the entity classes.

Class Names:

Database Table	Class Name	Generation Type
CUSTOMER	Customer	New
DISCOUNT_CODE	DiscountCode	New
MANUFACTURER	Manufacturer	New
MICRO_MARKET	MicroMarket	New

Project:

jsfjpacreud

Location:

Source Packages

Package:

com.ensode.jpai

☒ Generate Named Query Annotations for Persistent Fields  
☒ Generate JAXB Annotations  
☐ Generate MappedSuperclasses instead of Entities  
☒ Create Persistence Unit

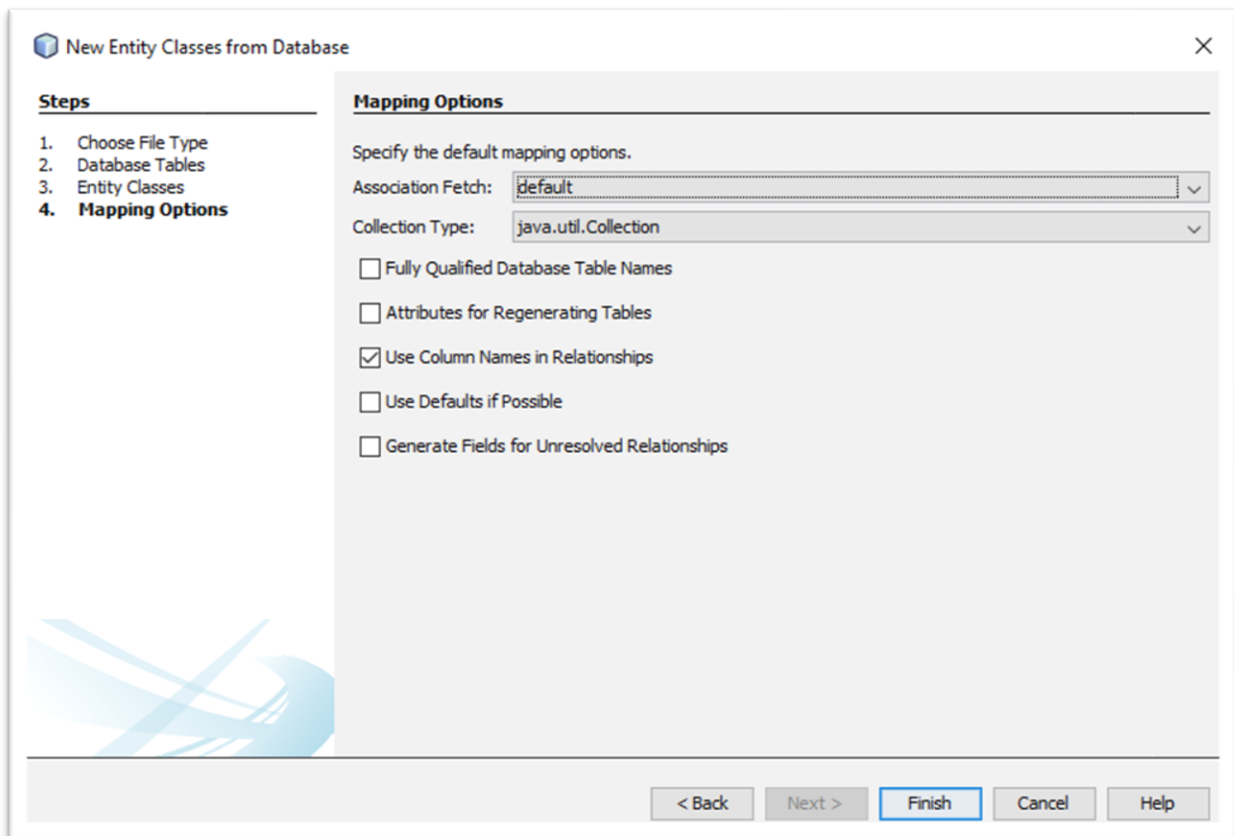
< Back

Next >

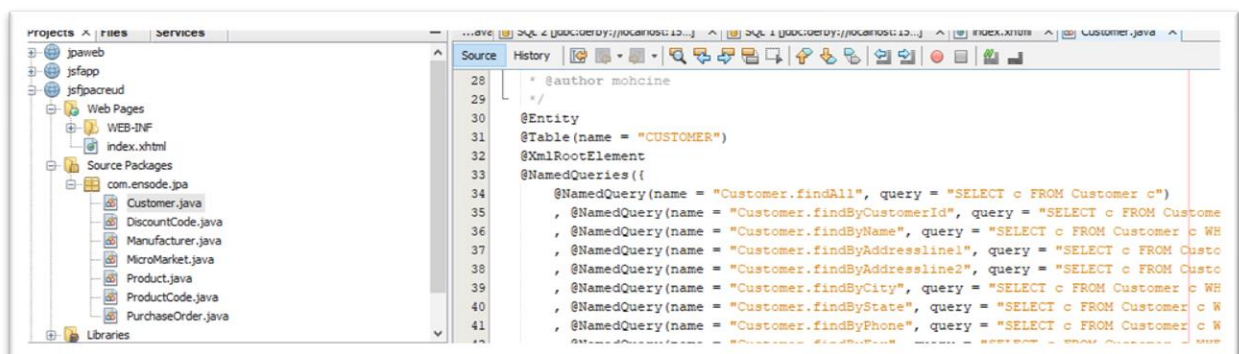
Finish

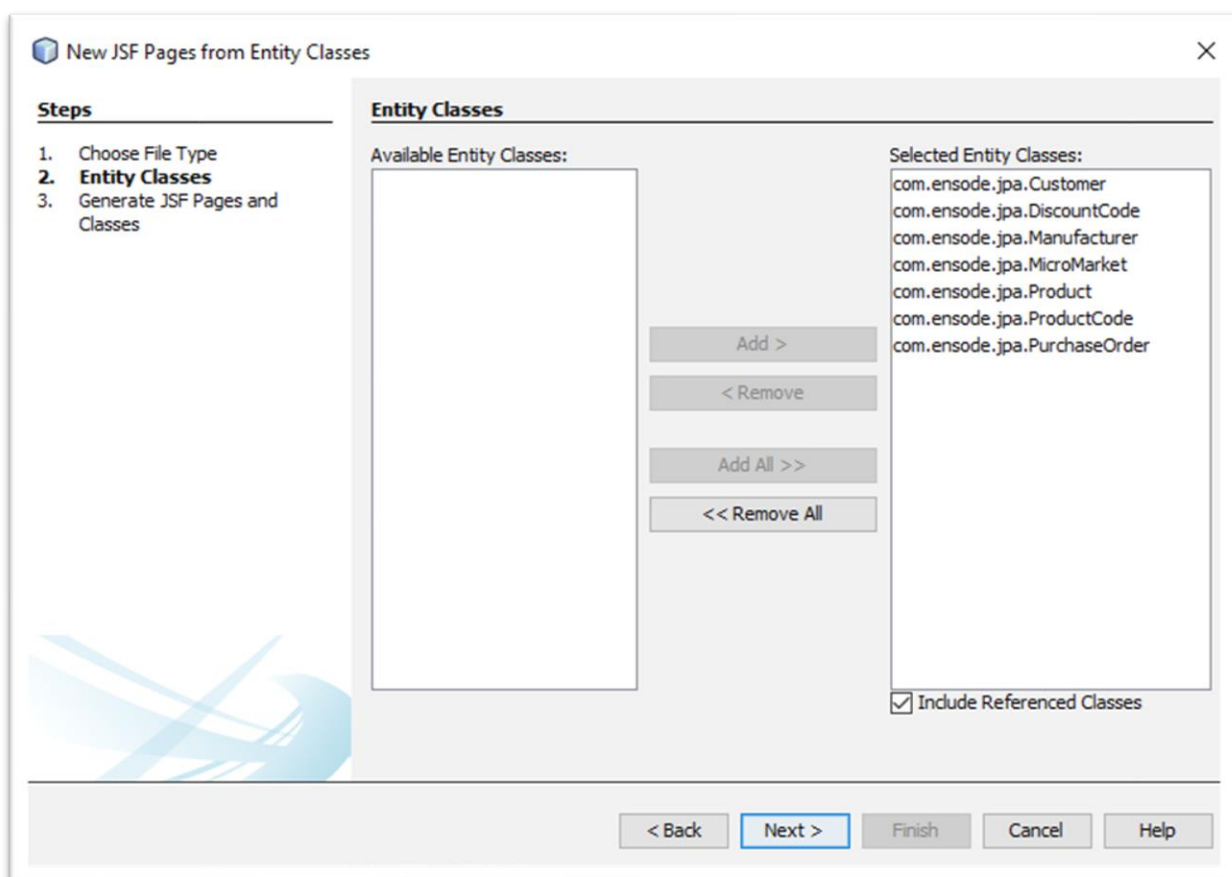
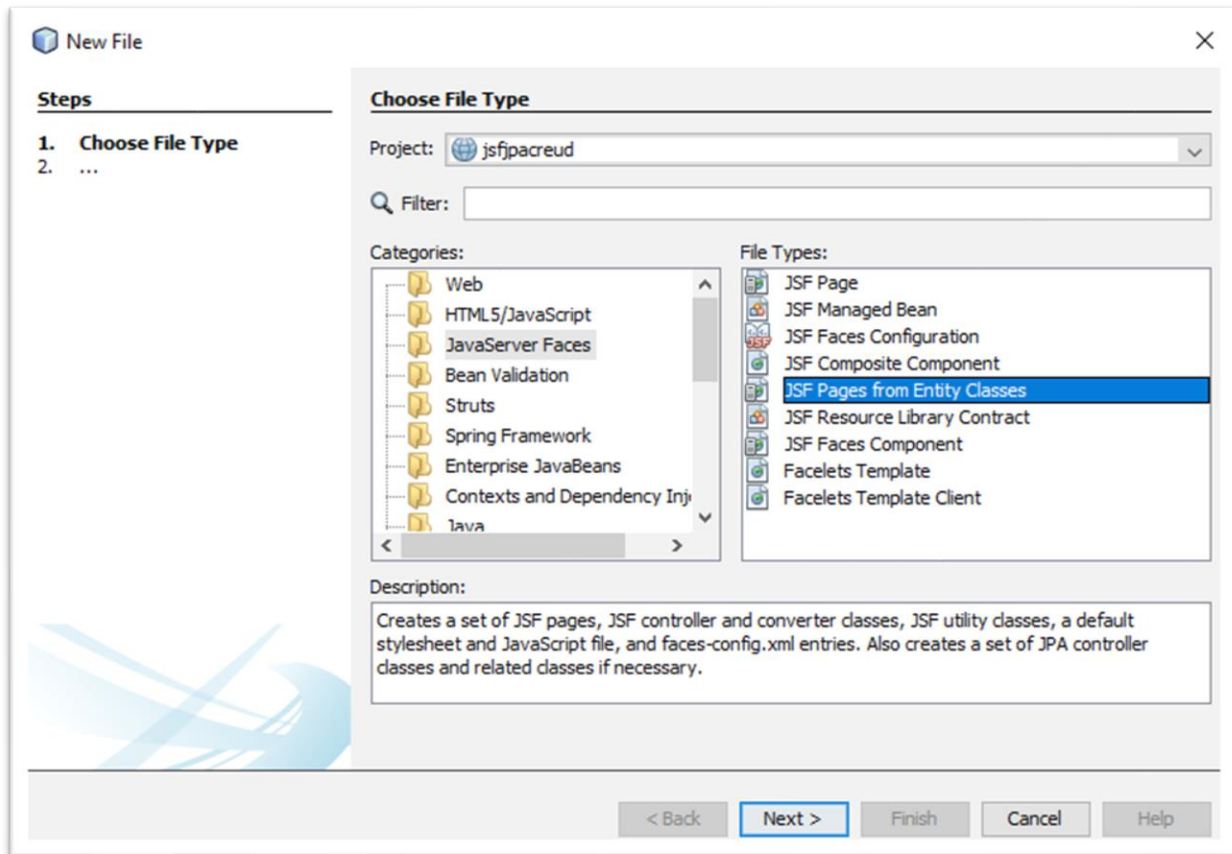
Cancel

Help



## Génération d'applications JSF à partir des entités





New JSF Pages from Entity Classes

Steps

1. Choose File Type

2. Entity Classes

3. Generate JSF Pages and Classes

Generate JSF Pages and Classes

Specify the package of existing or new EJBs and the package of JSF classes.

Project:

jsfjpacreud

Location:

Source Packages

Session Bean Package:

com.ensode.jpaController

JSF Classes Package:

com.ensode.jsf

Specify the location of new JSF pages.

JSF Pages Folder:

Browse...

Localization Bundle Name:

/Bundle

☐ Override existing files

Choose Templates:

Standard JavaServer Faces

[Customize Template](#)

< Back

Next >

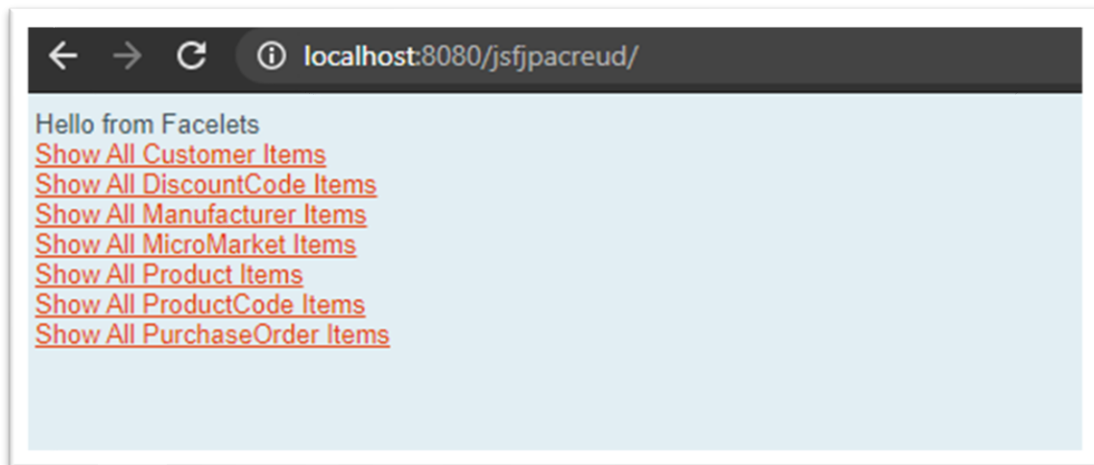
Finish

Cancel

Help

The Project Explorer view displays the following structure for the **jsfjpacreud** project:

- Web Pages
  - WEB-INF
  - customer
    - Create.xhtml
    - Edit.xhtml
    - List.xhtml
    - View.xhtml
  - discountCode
  - manufacturer
  - microMarket
  - product
  - productCode
  - purchaseOrder



A screenshot of a web browser window displaying a table of customer data. The browser's address bar shows 'localhost:8080/jsfjpacreud/faces/customer/List.xhtml?\_afPfm=2670b43d373613c3afa001e949bb'. The page title is 'List'. Below the title, there is a pagination control showing '1. 10/13' and a 'Next 10' link. The table has 12 columns: 'CustomerId', 'Name', 'Addressline1', 'Addressline2', 'City', 'State', 'Phone', 'Fax', 'Email', 'CreditLimit', 'DiscountCode', and 'Zip'. Each row represents a customer and includes a 'View Edit Destroy' link in the final column. At the bottom of the table, there is a 'Create New Customer' link.

CustomerId	Name	Addressline1	Addressline2	City	State	Phone	Fax	Email	CreditLimit	DiscountCode	Zip	
1	Jumbo Eagle Corp	111 E. Las Olivas Blvd	Suite 51	Fort Lauderdale	FL	305-555-0188	305-555-0189	jumboeagle@example.com	100000	N	95117	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
2	New Enterprises	9754 Main Street	P.O. Box 567	Miami	FL	305-555-0148	305-555-0149	www.new.example.com	50000	M	95035	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
25	Wren Computers	8989 Red Albatross Drive	Suite 9897	Houston	TX	214-555-0133	214-555-0134	www.wrencomp.example.com	25000	M	85638	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
3	Small Bill Company	8585 South Upper Murray Drive	P.O. Box 456	Alanta	GA	555-555-0175	555-555-0176	www.smalbill.example.com	90000	L	12347	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
36	Bob Hosting Corp.	65653 Lake Road	Suite 2323	San Mateo	CA	650-555-0160	650-555-0161	www.bobhostcorp.example.com	65000	H	94401	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
106	Early CentralComp	829 E Flex Drive	Suite 853	San Jose	CA	408-555-0157	408-555-0150	www.centralcomp.example.com	26500	L	95035	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
149	John Valley Computers	4381 Kelly Valley Ave	Suite 77	Santa Clara	CA	408-555-0169	408-555-0167	www.johnvalley.example.com	70000	L	95117	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
863	Big Network Systems	456 444th Street	Suite 45	Redwood City	CA	650-555-0181	650-555-0180	www.bignet.example.com	25000	N	94401	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
777	West Valley Inc.	88 Northsouth Drive	Building C	Dearborn	MI	313-555-0172	313-555-0171	www.westv.example.com	100000	L	48128	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
753	Zed Motor Co	2267 NE Michigan Ave	Building 21	Dearborn	MI	313-555-0151	313-555-0152	www.parts@ford.example.com	5000000	H	48128	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
722	Big Car Parts	52963 Notouter Dr	Suite 35	Detroit	MI	313-555-0144	313-555-0145	www.bparts.example.com	50000	N	48124	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[Create New Customer](#)

(1) Facebook

List

localhost:8080/jsfjpacreud/faces/productCode/List.xhtml;jsess

## List

1..6/6

ProdCode	DiscountCode	Description	
SW	M	Software	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
HW	H	Hardware	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
FW	L	Firmware	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
BK	L	Books	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
CB	N	Cables	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
MS	N	Misc	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[Create New ProductCode](#)

[Index](#)

localhost:8080/jsfjpacreud/faces/productCode/List.xhtml

## Create New ProductCode

ProdCode:

DiscountCode:

Description:

[Save](#)

[Show All ProductCode Items](#)

[Index](#)