

# La spécification JPA 2.1 Entity Manager

30/09/2017

Walid YAICH

[walid.yaich@esprit.tn](mailto:walid.yaich@esprit.tn)

Bureau E204



# Plan

- Entity manager
- Persistence context
- Cycle de vie
- Manipulation des entités
- Comment obtenir un « EntityManager »

# Entity manager

Pour assurer la persistance d'une entité dans la base, il est nécessaire d'invoquer une interface JPA spécifique, appelée **EntityManager**.

les entités peuvent être utilisées comme des objets normaux et peuvent **devenir gérées par le gestionnaire d'entités** lorsqu'il faut charger ou insérer des données dans la base.

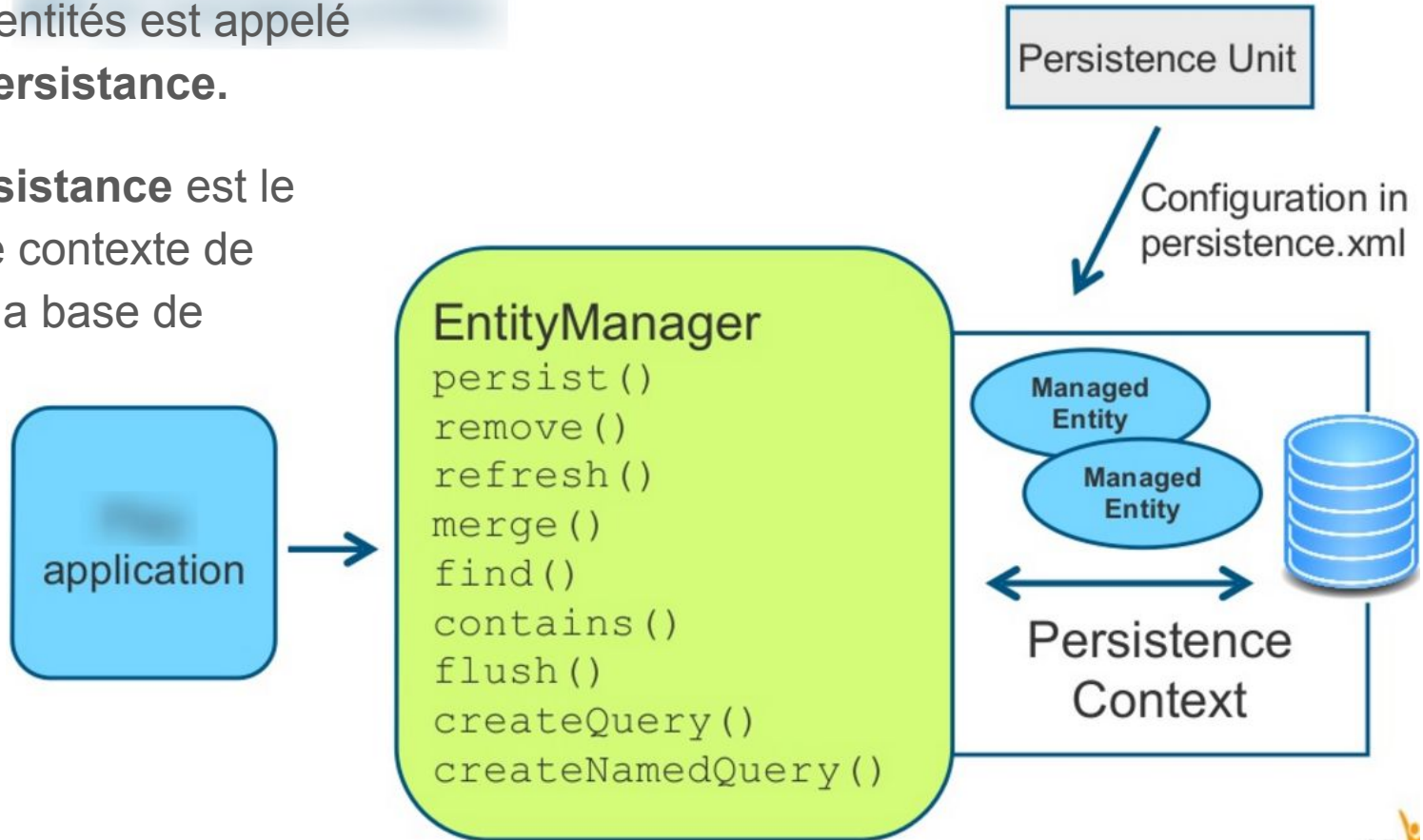
Lorsqu'une **entité est gérée**, il devient possible d'effectuer des opérations de persistance : le gestionnaire d'entités synchronisera **automatiquement** l'état de l'entité avec la base de données.

L'interface **javax.persistence.EntityManager** est implémentée par un fournisseur de persistance qui **produira et exécutera des instructions SQL**.

# Entity manager

L'ensemble des instances d'entités qui sont gérées par le gestionnaire d'entités est appelé **contexte de persistance**.

L'**unité de persistance** est le pont qui relie le contexte de persistance et la base de données.



# Persistence context

Le gestionnaire d'entités modifie ou consulte le **contexte de persistance** à chaque appel d'une méthode de l'interface `javax.persistence.EntityManager`.

Lorsque la méthode `persist()` est appelée, par exemple, l'entité passée en paramètre **sera ajoutée au contexte de persistance** si elle ne s'y trouve pas déjà.

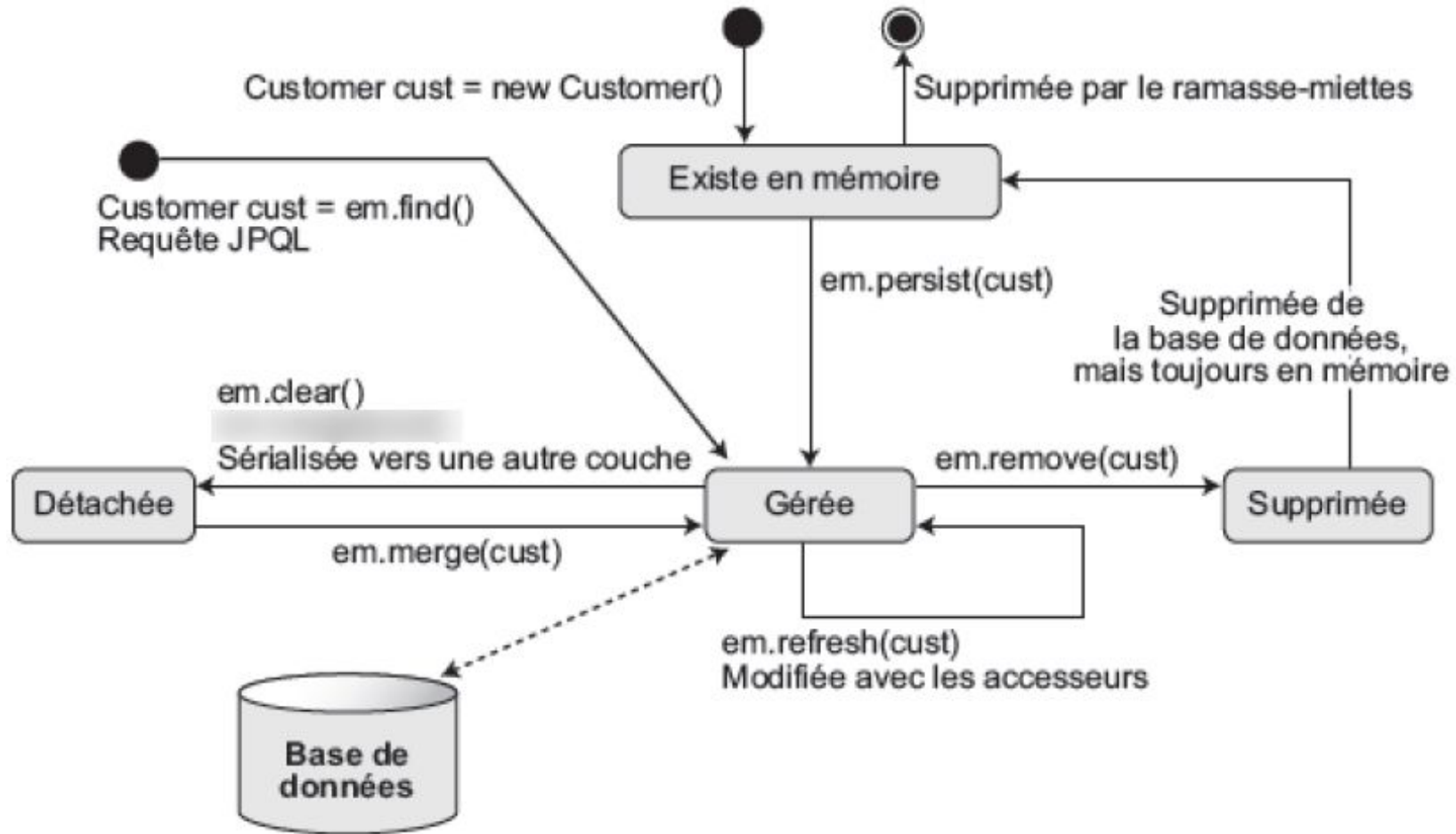
De même, lorsque l'on recherche une entité à partir de son identifiant, le gestionnaire d'entités **vérifie d'abord si elle existe déjà** dans le contexte de persistance.

Ce contexte peut donc être considéré comme un cache de premier niveau : c'est un espace réduit **où le gestionnaire stocke les entités avant d'écrire son contenu dans la base de données.**

# Cycle de vie

- **nouvelle (new)** : elle est créée mais pas associée à un contexte de persistance
- **gérée (managed)** par un gestionnaire de persistance elle a une identité dans la base de données (un objet peut devenir géré par la méthode persist, ou merge d'une entité détachée ; un objet géré peut aussi provenir d'une requête faite par un gestionnaire d'entités ou d'une navigation à partir d'un objet géré)
- **détachée (detached)** : elle a une identité dans la base mais elle n'est plus associée à un contexte de persistance (une entité peut, par exemple, devenir détachée si le contexte de persistance est vidé ou si elle est envoyée dans une autre JVM par RMI)
- **supprimée (removed)**: elle a une identité dans la base ; elle est associée à un contexte de persistance et ce contexte doit la supprimer de la base de données (passe dans cet état par la méthode remove)

# Cycle de vie



# Manipulation des entités

<i>Méthode</i>	<i>Description</i>
void persist(Object entity)	Crée une instance gérée et persistante
<T> T find(Class<T> entityClass, Object primaryKey)	Recherche une entité de la classe et de la clé indiquées.
void remove(Object entity)	Supprime l'instance d'entité du contexte de persistance et de la base de données.
<T> T merge(T entity)	Fusionne l'état de l'entité indiquée dans le contexte de persistance courant
void refresh(Object entity)	Rafraîchit l'état de l'instance à partir de la base de données en écrasant les éventuelles modifications apportées à l'entité
void flush()	Synchronise le contexte de persistance avec la base de données.
void clear()	Vide le contexte de persistance. Toutes les entités gérées deviennent détachées
void clear(Object entity)	Supprime l'entité indiquée du contexte de persistance
boolean contains(Object entity)	Teste si l'instance est une entité gérée appartenant au contexte de persistance courant



# Comment obtenir un « EntityManager »

On obtient un entitymanager par **injection de dépendances**.

```
@Stateless
public class UtilisateurService implements UtilisateurService {

    @PersistenceContext(unitName="imputation-ejb")
    EntityManager em;
}
```

```
<persistence-unit name="imputation-ejb">
  <jta-data-source>java:/IMPUTATION_DS</jta-data-source>
  <properties>
    <property name="hibernate.hbm2ddl.auto" value="create" />
    <property name="hibernate.show_sql" value="true" />
    <property name="hibernate.format_sql" value="true" />
  </properties>
</persistence-unit>
```

```
@Stateless
public class UtilisateurService implements UtilisateurService {

    @PersistenceContext(unitName="imputation-ejb")
    EntityManager em;

    @Override
    public void ajouterUtilisateur(Utilisateur user) {
        em.persist(user);
    }
}
```