

Examen Appli_Web 21

Question de cours

1. `<a href>` 使用 `get` 方法提交请求, 即 ` Ajout`
2. `` 这是 HTTP 的 `get` 方法
3. 在表单中, `<form method="POST" action="GET">` 说明该表单使用 `POST` 方法, 而请求的 URL 为 `GET`;
4. 在 `POST` 方法的 HTTP 请求报文中, 请求的内容出现在请求体 `Request Body` 中, 它的格式可以由请求头 `Request Head` 指定, 常见的有最普通的格式 `param1=xxx&value1=xxx`, 还可以使用 `XML` 和 `JSON` 格式
- 5.
6. `EntityManager` 可以根据主键找到对应的 `EntityBeans`, 可以使用 `@PersistenceContext` 来注释
7. 在 `EntityManager em` 中, 我们可以使用 `em.persist()`; 来保存 (`INSERT`) 数据
8. `@Singleton` 意味着该 `EJB Container` 仅创建一个实例来处理请求
- 9.
0. 在 `Servlet` 中, 如果我们想引用 `Facade`, 我们需要使用 `@EJB` 并实例化一个 `Facade`
1. 在请求转发 `request.getRequestDispatcher(URL).forward(request,response)` 中, `URL` 是转发目的地的 `url`, 可以是 `html`, 也可以是 `jsp`
2. `servlet` 使用 `request.getParameter` 来得到请求体中的参数
3. 如果传入一个临时对象, 我们对其 `merge()` 操作, 我们可以将其永久化操作 `INSERT` or `UPDATE`
4. 在 `MVC` 框架中, `servlet` 作为 `Controller` 可以完成对 `JSP` 的调用
5. 在 `MVC` 框架中, `JSP` 作为 `View`, 可以向浏览器返回页面、接收从 `servlet` 传来的数据并显示

Problem

1. One entreprise → Many chantiers ; One chantier → One entreprise : bi-direction @OneToMany
2. One chantiers → Many ouvriers ; One ouvrier → One chantier : bi-direction @OneToMany

Question 1

```
@Entity
public class Ouvrier {
    @Id
    @GeneratedValue
    int id;
    String nom;

    @JoinColumn(name="chantier_id")
    @ManyToOne
    Chantier chantier;
}
```

```
@Entity
public class Chantier {
    @Id
```

```

@GeneratedValue
int id;
String adresse;

@OneToMany(mappedBy=chantier)
List<Ouvrier> ouvriers;

@JoinColumn(name="entreprise_id")
@ManyToOne
Entreprise entreprise;
}

```

```

@Entity
public class Entreprise{
    @Id
    @GeneratedValue
    int id;
    String numero;

    @OneToMany(mappedBy=entreprise)
    List<Chantier> chantiers;
}

```

Question 2

BD:

Ouvrier table:

id (PrimaryKey)	nom	chantier_id(ForeignKey)
-----------------	-----	-------------------------

Chantier table:

id (PrimaryKey)	adresse	entreprise_id (ForeignKey)
-----------------	---------	----------------------------

Entreprise table:

id (PrimaryKey)	numero
-----------------	--------

Question 3

Facade.java

```

@Singleton
public class Facade {
    @PersistenceContext
    EntityManager em;

    public void ajoute_ouvrier (String nom, int id_chantier){
        Ouvrier ouvrier = new Ouvrier();
    }
}

```

```

    Chantier chantier = em.find(Chantier.class, id_chantier);
    ouvrier.setNom(nom);
    ouvrier.setChantier(chantier);
    chantier.ouvriers.put(ouvrier.id, ouvrier);
    em.persist(ouvrier);
}

public void changer_entreprise(int id_chantier, int id_entreprise){
    Chantier chantier = em.find(Chantier.class, id_chantier);
    Entreprise entreprise = em.find(Entreprise.class, id_entreprise);
    chantier.setEntreprise(entreprise);
    em.merge(chantier);
}
}

```

Question 4

```

@Entity
public class Ouvrier {
    @Id
    @GeneratedValue
    int id;
    String nom;

    @JoinTable(
        name="t_Ouvrier_Chantier",
        joinColumns={
            @JoinColumn(name="ouvrier_id",
                referenceColumnName="id")},
        inverseJoinColumns={
            @JoinColumn(name="chantier_id",
                referenceColumnName="id")
        })
    @ManyToMany
    Collection<Chantier> chantiers;
}

public Collection<chantier> getChantier(){
    return this.chantiers;
}

public void setChantier (Chantier chantier){
    this.getChantier().put(chantier.id, chantier);
}
}

```

```

@Entity
public class Entreprise{
    @Id
    @GeneratedValue
    int id;
    String numero;
}

```

```

@JoinTable(
    name="t_Entreprise_Chantier",
    joinColumns={
        @JoinColumn(name="entreprise_id",
            referenceColumnName="id")},
    inverseJoinColumns={
        @JoinColumn(name="chantier_id",
            referenceColumnName="id")
    })
@ManyToMany
Collection<Chantier> chantiers;
}

```

```

public void ajoute_ouvrier(String nom, int[] id_chantier){
    Ouvrier ouvrier = new Ouvrier();
    ouvrier.setNom(nom);
    for(i=0; i<id_chantier.size; i++){
        Chantier chantier = em.find(Chantier.class, id_chantier[i]);
        ouvrier.setChantier(chantier);
        chantier.setOuvrier(ouvrier);
        em.persist(ouvrier);
        // ouvrier.setChantier(null);
    }
}

```

Question 5

```

Collection<Entreprise> getEntreprises(){
    String jpql = "SELECT * FROM Entreprise e";
    Query query = em.createQuery(jpql);
    Collection<Entreprise> result = (Collection<Entreprise>) query.getResultList();
    return result;
}

```

Question 6