

OFFRES D'EMPLOIS MVC

1. Modifier les fichiers fournis afin de pouvoir **créer une offre** en base de données à partir des données saisies dans un formulaire.

Pour cela, vous aurez à modifier :

Le contrôleur : controller/créer_offre.php

L'insertion de l'offre en base de données ne pourra se faire qu'après avoir reçu les variables \$_POST['title'] et \$_POST['description'] venant du formulaire (view).

```
if (isset($_POST['title']) && isset($_POST['description'])) {  
    insérer l'offre en base de données;  
    afficher la vue en envoyant la variable status et l'offre créée ;  
}else {  
    afficher la vue ;  
}
```

La vue : view/creer_offre.html.twig

Si aucune variable n'est reçue, on peut afficher le formulaire, sinon, on va afficher le résultat de l'insertion dans la base de données.

```
{% if status is not defined %}  
    Afficher Le formulaire  
{% else %}  
    Afficher Le résultat de L'insertion  
{% endif %}
```

REMARQUE :

Attention à l'action dans le formulaire. Grâce au htaccess, le fichier exécuté est toujours index.php, ce qui nous intéresse c'est le sous contrôleur à appeler grâce à la variable action dans l'url. Vous allez spécifier ce contrôleur uniquement par son nom dans votre formulaire :

```
<form action="creer_offre" method="post">
```

2. Supprimer une offre.

Pour cela, vous aurez à :

Modifier la vue « `offres.html.twig` » : rajouter un bouton sur chaque offre qui permet de supprimer cette offre en base de données.

Ceci est notre header	
Delete	php Développeur : Vous travaillerez comme développeur après votre super formation AFPA
Delete	Java développeur : Pro des lignes de code
Delete	Python Développeur : Special offer for JJ
Ceci est notre footer	

Le plus simple est de créer un formulaire par offre. Récupérer l'id de l'offre et l'insérer de manière cachée dans votre formulaire.

```
<form action="delete_offre" method="post">
  <div class="container">
    <button type="submit">Delete</button>
    <label for="title"><b> {{offer.get_title}} : </b></label>
    <label for="description"><b> {{offer.get_description}}</b></label>
    <input value="{{offer.get_id}}" name="id" type="hidden">
  </div>
</form>
```

Créer le contrôleur « `delete_offre.php` »

Ce contrôleur va recevoir l'id de l'offre à supprimer via la variable `$_POST['id']`. Il va demander au modèle de supprimer l'offre correspondante. Il va également afficher la vue « `delete_offre.html.twig` » en lui envoyant le résultat de la suppression (suppression OK ou Erreur suppression).

Modifier le Modèle : Ecrire la méthode `deleteOne($id)` dans le fichier `OffresDAO.php`. Voici la requête qui va permettre de supprimer une offre dans la base de données :

```
public function deleteOne($idOffer): int
{
    $query = $this->_bdd->prepare('DELETE FROM offers WHERE offers.id = :idOffer');
    $query->execute(array(':idOffer' => $idOffer));
    return ($query->rowCount());
}
```

La méthode va retourner le nombre d'éléments supprimés.

Créer la vue « `delete_offre.html.twig` » : afficher le résultat de la suppression.