	Objectifs	Création d'un blog avec CakePHP – Partie 3
	Logiciels	XAMPP, phpmyadmin, netbeans
	Ressources	Documents fournis, Internet
	Mots clés	Framework, MVC

CONSIGNE : SUIVRE LES INSTRUCTIONS A LA LETTRE.

**L'objectif de ce TP est de créer un blog de A à Z en utilisant le Framework CakePHP.**

## 1\ Associer un post à un user

- Regardez l'utilisation de la balise SELECT à l'adresse suivante :

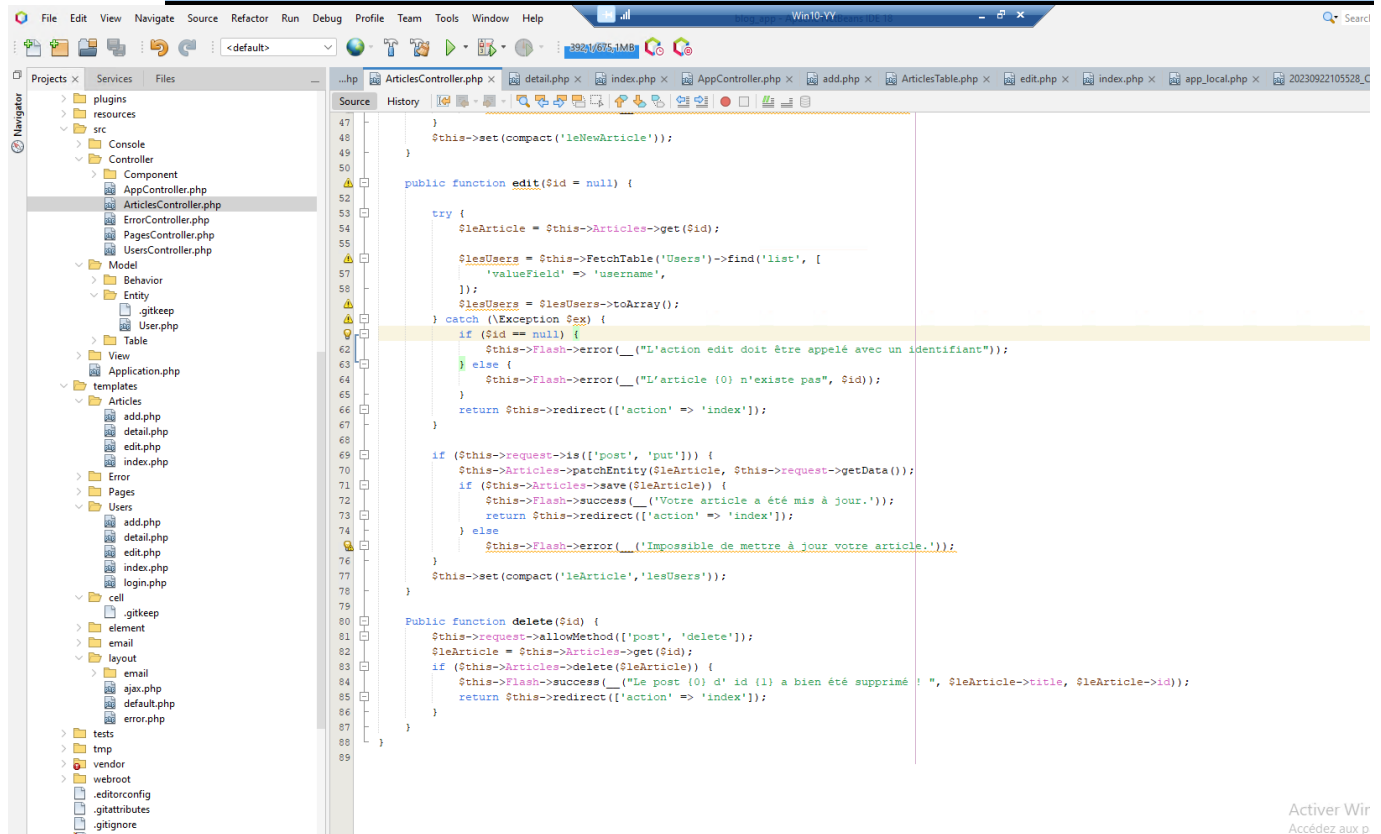
[https://www.w3schools.com/tags/tag\\_select.asp](https://www.w3schools.com/tags/tag_select.asp)

- De quoi a-t-on besoin pour faire un select ?
- A quoi devrait ressembler notre select ?

**Mise en œuvre :**

Dans la fonction Edit du controller Articles

- Est-ce qu'on a accès à la table Users depuis le controller Articles ?  
Non car dans le controller Articles on a accès qu'à la table Articles
- Charger le modèle Users
- `$lesUsers = $this->FetchTable('Users')->find('list', [`



Activer Wir  
Accédez aux p

- Quel find dans CakePHP doit-on utiliser ?

<https://book.cakephp.org/4/fr/orm/retrieving-data-and-resultsets.html>

- Générer un tableau qui contient les id et les username ?
- Modifiez la vue Edit pour mettre le select

```

echo $this->Form->control('user_id', ['options'=> $lesUsers, 'label' =>
'Selectionnez un user']);

```

- Rendez ce champ obligatoire comme le titre et le contenu  
->notEmptyString('user\_id');
- Faire la même chose pour l'ajout d'un article



- Page 3 sur 6

## 2\ Nouvelle fonctionnalité : les commentaires

- Par quoi est caractérisé un commentaire ?  
Un commentaire est caractérisé par son id et son contenu
- Avec qui sont rattachés les commentaires ?  
Les commentaires sont rattachés par Articles has\_many comments comments belongsTo Articles
- Implémentez la table des commentaires  
cake bake migration CreateComments content:text created
- Créez le modèle et le contrôleur  
Copier coller : ArticlesController et ArticlesTable
- Mettez la relation dans cakephp Sur la liste des articles, affichez le nombre de commentaires associés dans une nouvelle colonne (fonction count() dans la vue...)s
- Créez une méthode et une vue add pour les commentaires
- Insérez des commentaires pour vérifier que le comptage fonctionne bien en changeant la valeur du article\_id dans le contrôleur.
- Quel est le problème de ce système ?

Le problème c'est que chaque commentaire qui sera créer sera rattacher à l'article si on veut modifier l'article il faudra modifier dans le CommentsController

- Sur la page de l'article en question (detail.php), rajoutez ce qui suit :

```
<?php
echo $this->Html->script('jquery360min');
?>

<script>
    $(document).ready(function () {
        $("#showcom").click(function () {
            if ($("#display").is(":visible") == false)
            {
                $("#display").show();
            }
            else {
                $("#display").hide();
            }
        });
    });
</script>

<?=
$this->Html->link(
    'Ajoutez un commentaire', '#', ['class' => 'button', 'id' =>
'showcom']
);
?>

<div id="display" style="display: none">
```

```
<?= $this->element('comments'); ?>
</div>
```

- Télécharger le fichier js de jquery et mettez-le dans le dossier webroot/js et renommez le en jquery360min.js
- Créez dans le dossier src/Template/Element le fichier comments.php et copiez collez le contenu de la vue add.php de comments
- Lancez la page index des articles et corrigez les erreurs.
- Sur la page detail d'un article, affichez tous ses commentaires, du plus ancien au plus récent par date de modification (juste avant le bouton rajouter un commentaire)

```
$leArticle = $this->Articles->get($id, [
    'contain' => ['Comments' => function ($q) {
        return $q
    }
]);

$order(['Comments.created asc']);

<h3>Les commentaires</h3>
<?php foreach ($leArticle->comments as $comm): ?>
    <table border="1">
        <tr>
            <td><?= $comm->title ?></td>
        </tr>
        <tr>
            <td><?= nl2br(h($comm->content)) ?></td>
        </tr>
        <tr>
            <td>id : <?= $comm->id ?>
                Créé le : <?= $comm->created-
                >format(DATE_RFC850) ?>
                Modifié le : <?= $comm->created-
                >format(DATE_RFC850) ?>
            </td>
        </tr>
    </table>
<?php endforeach; ?>
```

- A côté de chaque commentaire, un bouton pour supprimer le commentaire
- Rendre les champs obligatoires pour créer un commentaire
- Essayer de supprimer un article sans commentaire puis un article avec commentaires. Que se passe-t-il ?
- Suppression en Cascade du post et donc des commentaires associés :

Quand les entités sont supprimées, les données associées peuvent aussi être supprimées. Si vos associations HasOne et HasMany sont configurées avec dependent, les opérations de suppression se feront aussi en 'cascade' sur leurs entités.

```
$this->hasMany('Comments', [
```



```
| 'dependent' => true,  
| );
```

- Testez
- Faites un bouton pour afficher/cacher les users sur la liste des users