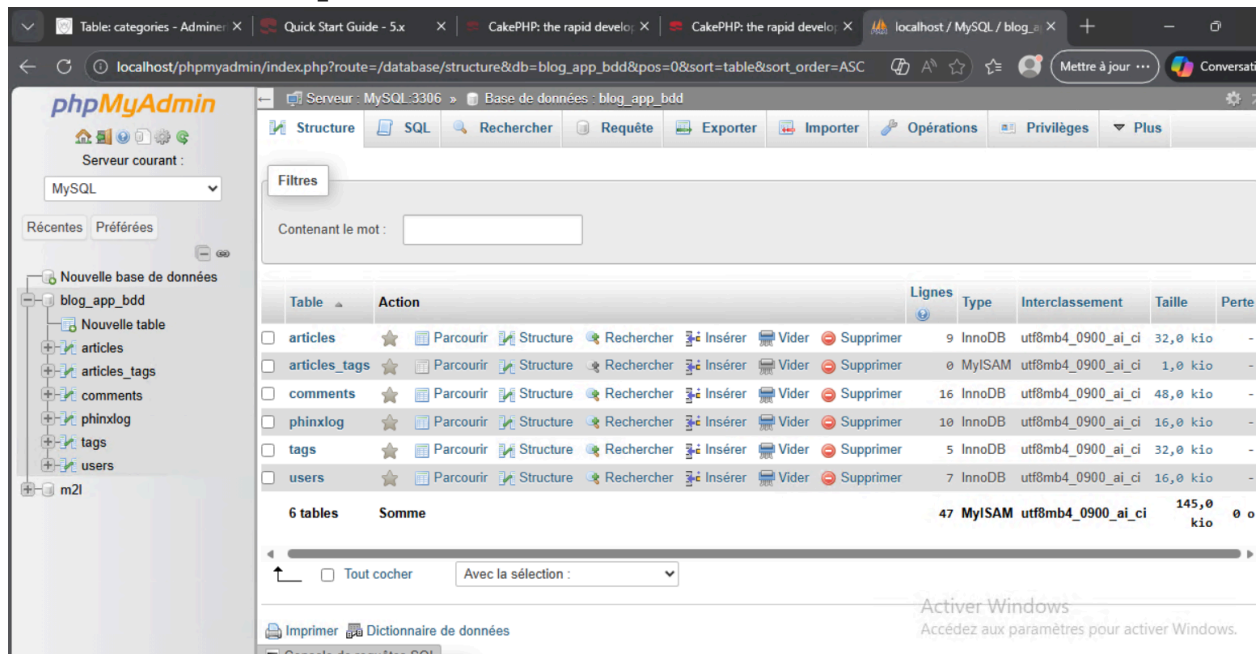




Table de matière

Les base de donner Blog_app.....	2
1 ère partie création de table avec cake php.....	2
Importation dans netbeans.....	3
Creation utilisateur.....	12
Modifier de migration et l'insertion d'un champs.....	19
La suite TP3 :.....	20
Faire un commentaire.....	21
Création de bouton pour passer d'utilisateur à articles.....	28
Creation des icones et réglage de la taille de l'icone.....	30
3 eme création deAuthentification.....	31
Modification de l'articles controller :.....	40
Jointure de modele.....	40
4 eme création des commentaire.....	43
5eme partie Création des tags.....	44
Diagramme de classe final du TP blog_app:.....	50
Recap de Base :.....	50
1.Structure du projet.....	50
2. MVC en action.....	50

Les base de donner Blog_app



The screenshot shows the phpMyAdmin interface for a MySQL database named 'blog_app_bdd'. The left sidebar displays the database structure with tables: articles, articles_tags, comments, phinxlog, tags, and users. The main area shows a table structure overview with columns: Table, Action, Lignes, Type, Interclassement, Taille, and Perte.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
articles	Parcourir Structure Rechercher Insérer Vider Supprimer	9	InnoDB	utf8mb4_0900_ai_ci	32,0 kio	-
articles_tags	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8mb4_0900_ai_ci	1,0 kio	-
comments	Parcourir Structure Rechercher Insérer Vider Supprimer	16	InnoDB	utf8mb4_0900_ai_ci	48,0 kio	-
phinxlog	Parcourir Structure Rechercher Insérer Vider Supprimer	10	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
tags	Parcourir Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8mb4_0900_ai_ci	32,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
6 tables	Somme	47	MyISAM	utf8mb4_0900_ai_ci	145,0 kio	0 0

1ère partie création de table avec cake php

- Création du projet avec composer

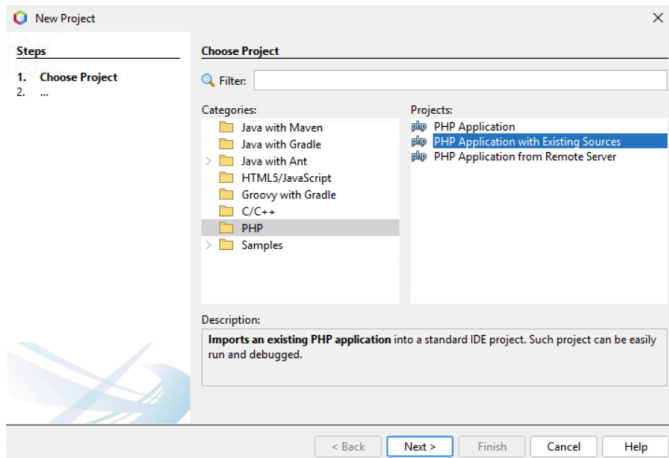
Pour créer projet aller sur explorateur de fichier ,puis cliquer sur ce pc , disque local c: aller dans wamp 64 , cliquer sur le dossier www et lancer un terminal:

```
C:\wamp64\www>composer create-project --prefer-dist cakephp/app:5.*
```

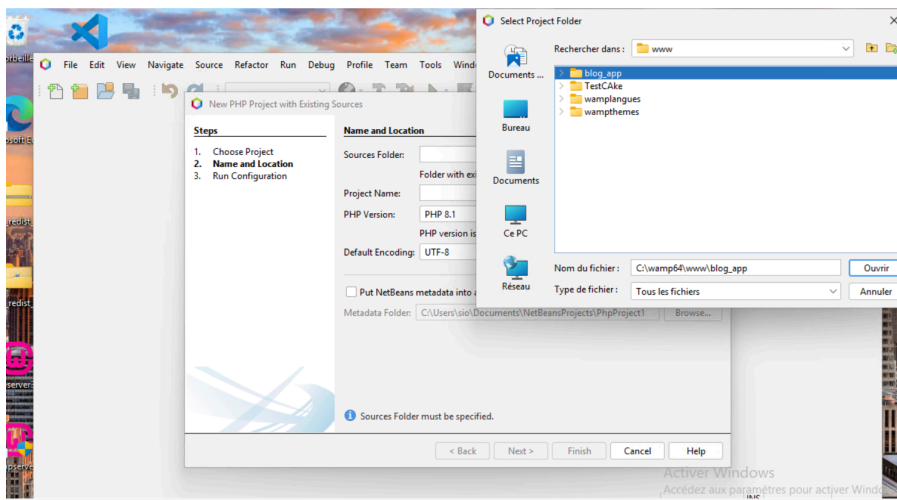
blog_app

Pour les permission mettre y

- Importation dans netbeans



- Aller sur netbeans dans new projet , cliquer sur php.



Arriver sur le document cliquer sur ce pc , puis cliquer sur blog_app

- Création de la base de données

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [version 10.0.26200.6901]
AnkhTech - (c) Microsoft Corporation. Tous droits réservés.
C:\wamp64\bin\mysql\mysql9.1.0\bin>create database
```

- Créer un utilisateur dédié avec tout les droit pour blog-app uniquement
create user blog_app identified by "king5498!!02#";


```

    ['title' => 'titre1', 'content' => 'sdfsdf'
     . 'dfsdfsdf',
     . 'sdfsffs', 'created' => date('Y-m-d H:i:s'), 'modified' => date('Y-m-d H:i:s')],
    ['title' => 'titre2', 'content' => 'content2', 'created' => date('Y-m-d H:i:s'), 'modified' => date('Y-m-d H:i:s')],
    ['title' => 'titre3', 'content' => 'content3', 'created' => date('Y-m-d H:i:s'), 'modified' => date('Y-m-d H:i:s')],
    ['title' => 'titre4', 'content' => 'content4', 'created' => date('Y-m-d H:i:s'), 'modified' => date('Y-m-d H:i:s')],
    ['title' => 'titre5', 'content' => 'content5', 'created' => date('Y-m-d H:i:s'), 'modified' => date('Y-m-d H:i:s')]
];
$articles->insert($rows)->saveData();
}

public function down(): void
{
    $this->execute('DELETE from articles');
}
}

```

Modifier les l'ancien code php vers le nouveaux

- Création modèle pour Articles

```

<?php
namespace App\Model\Table;

use Cake\ORM\Table;

class ArticlesTable extends Table {
}

```

- Création du controller
- Création template pour article/index

```

ArticlesController.php x detail.php x index.php x
Source History
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends ApplicationController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->all();
10        $this->set(compact('mesArticles'));
11    }
12    public function detail($id){
13        $leArticle = $this->Articles->get($id);
14        $this->set(compact('leArticle'));
15    }
16 }
17
18

```

- Autre action detail vue

```

<?=
    $this->html->link($article->title, [
        'controller' => 'articles',
        'action' => 'detail',
        $article->id]);
//l'url généré sera de la forme /articles/detail/...
?>

```

Le lien qui permet d'aller sur la vue détail

```

ArticlesController.php x detail.php x index.php x
Source History
<h1><?= h($leArticle->title) ?></h1>
2 <p><?= nl2br(h($leArticle->content)) ?></p>
3 <p>
4     <small>Created : <?= $leArticle->created->format(DATE_RFC850) ?></small>
5 </p>
6 <?=
7     $this->html->link('retour a la liste des articles', [
8         'controller' => 'articles',
9         'action' => 'index']);
10 //l'url généré sera de la forme /articles/detail/...
11 ?>

```

public function detail(\$id)

- Le try catch

```
ArticlesController.php x detail.php x index.php x AppController.php x
Source History
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends AppController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->all();
10        $this->set(compact('mesArticles'));
11    }
12
13    public function detail($id = null){
14        try{
15            $leArticle = $this->Articles->get($id);
16        } catch (\Exception $ex) {
17            if ($id == null) {
18                $this->Flash->error(__("L'action detail doit être appelé avec un identifiant"));
19            } else {
20                $this->Flash->error(__("L'article {0} n'existe pas", $id));
21            }
22            return $this->redirect(['action' => 'index']);
23        }
24        $leArticle = $this->Articles->get($id);
25        $this->set(compact('leArticle'));
26    }
27
28 }
```

Resultat sans id

CakePHP Documentation API

L'action detail doit être appelé avec un identifiant

Tous les articles du Blog

Id	Titre	Date de creation	Date de modifier
1	titre1	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
2	titre2	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
3	titre3	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
4	titre4	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
5	titre5	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC

Avec un id qui n'existe pas

L'article 1545 n'existe pas

Tous les articles du Blog

Id	Titre	Date de creation	Date de modifier
1	titre1	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
2	titre2	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
3	titre3	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
4	titre4	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC
5	titre5	Wednesday, 01-Oct-25 06:59:25 UTC	Wednesday, 01-Oct-25 06:59:25 UTC

Modification de la création d'article

```

43         $this->Flash->error(__("L'article {0} n'existe pas", $id));
44     }
45     return $this->redirect(['action' => 'index']);
46 }
47
48     $this->set(compact('leArticle', 'leNewComment'));
49 }
50
51 public function add() {
52     $leNewArticle = $this->Articles->newEmptyEntity();
53     if ($this->request->is('post')) {
54         $leNewArticle = $this->Articles->patchEntity($leNewArticle, $this->request->getData());
55         $leNewArticle->user_id = $this->request->getAttribute('identity')->id;
56         if ($this->Articles->save($leNewArticle)) {
57             $this->Flash->success(__("L'article a été sauvegardé."));
58             return $this->redirect(['action' => 'index']);
59         } else
60             $this->Flash->error(__("Impossible d'ajouter votre article."));
61     }
62     $this->set(compact('leNewArticle'));

```

8 kevin zeus 0

Friday, 21-Nov-25 05:27:02
UTCFriday, 21-Nov-25 05:27:02
UTC

MODIFIER

SUPPRIMER

Active Windows
Accédez aux paramètres pour activ

Créer un bouton:

```
add.php x detail.php x index.php x ArticlesController.php x
Source History
<h1>Tous les articles du Blog</h1>
2
3 <?=  
4 $this->html->link('ajouté un post', [  
5     'controller' => 'articles','action' => 'add'],  
6     ['class' => 'button']);  
7 //l'url généré sera de la forme /articles/detail/...  
8 ?>
```

La créer le bouton articles dans index pour sélectionner les articles demander

```
add.php x detail.php x index.php x ArticlesController.php x
Source History
<h1>Ajouter un post</h1>
2 <?php  
3 echo $this->Form->create($leNewArticle);  
4 echo $this->Form->control('title');  
5 echo $this->Form->control('content', ['rows' => '3']);  
6 echo $this->Form->button(__('Sauvegarder l'article'));  
7 echo $this->Form->end();  
8 ?>  
9 <?=  
10 $this->html->link('retour a la liste des articles', [  
11     'controller' => 'articles','action' => 'index'],  
12     ['class'=>'button']);  
13 //l'url généré sera de la forme /articles/detail/...  
14 ?>
```

Lien :

```
add.php x detail.php x index.php x ArticlesController.php x
Source History
<h1><?=  
2 <p><?=  
3 <p>  
4     <small>Created : <?=  
5 </p>  
6 <?=  
7 $this->html->link('retour a la liste des articles', [  
8     'controller' => 'articles',  
9     'action' => 'index']);  
10 //l'url généré sera de la forme /articles/detail/...  
11 ?>
```

Le lien entre l'articles qui vas etre créer

- Validation par défaut :

```
add.php x detail.php x index.php x ArticlesController.php x ArticlesTable.php x
Source History
<?php
2
3 namespace App\Model\Table;
4
5 use Cake\ORM\Table;
6 use Cake\Validation\Validator;
7
8 class ArticlesTable extends Table {
9
10     public function initialize(array $config): void {
11         $this->addBehavior('Timestamp');
12     }
13
14     public function validationDefault(Validator $validator):Validator {
15         $validator
16             ->notEmptyString('title')
17             ->notEmptyString('content');
18
19         return $validator;
20     }
21 }
22
```

- Edit

```
ArticlesController.php x add.php x detail.php x edit.php x index.php x ArticlesTable.php x
Source History
<h1>Modifier l'article "<?=$leArticle->titre->get($id=5) ?>"</h1>
2
3 <?php
4     echo $this->Form->create($leArticle);
5     echo $this->Form->control('title');
6     echo $this->Form->control('content', ['rows' => '3']);
7     echo $this->Form->button(__("Mettre à jour l'article"));
8     echo $this->Form->end();
9
10     ?>
11     <br/>
12     <?=$this->html->link('retour a la liste des articles', [
13         'controller' => 'articles', 'action' => 'index',
14         ['class' => 'button']);
15     //l'url généré sera de la forme /articles/detail/...
16     ?>
```

- Edite controleur

```
}
public function edit($id) {
    $leArticle = $this->Articles->get($id);
    if ($this->request->is(['post', 'put'])) {
        $this->Articles->patchEntity($leArticle, $this->request->getData());
        if ($this->Articles->save($leArticle)) {
            $this->Flash->success(__("Votre article a été mis à jour.));
            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__("Impossible de mettre à jour votre article.));
        }
    }
    $this->set(compact('leArticle'));
}
```

- Rajouté du try catch dans edit pour faire le trie

```

}
public function edit($id = null) {
    try{
        $leArticle = $this->Articles->get($id);
    } catch (\Exception $exc){
        if ($id == null) {
            $this->Flash->error(__("L'action detail doit être appelé avec un identifiant"));
        } else {
            $this->Flash->error(__("L'article (0) n'existe pas", $id));
        }
        return $this->redirect(['action' => 'index']);
    }

    $this->set(compact('leArticle'));

    if ($this->request->is(['post', 'put'])) {
        $this->Articles->patchEntity($leArticle, $this->request->getData());
        if ($this->Articles->save($leArticle)) {
            $this->Flash->success(__("Votre article a été mis à jour.));
            return $this->redirect(['action' => 'index']);
        } else
            $this->Flash->error(__("Impossible de mettre à jour votre article.));
    }
    $this->set(compact('leArticle'));
}

```

Activer Windows

- Bouton edit pour aller directement dans le controller articles

```

<?=  

$this->html->link('retour a la liste des articles', [  

    'controller' => 'articles', 'action' => 'edit',  

    ['class' => 'button']);  

//l'url généré sera de la forme /articles/detail/  

?>

```

- delete

```

5 | }  

6 |     public function delete($id = null) {  

7 |         $this->request->allowMethod(['post', 'delete', 'get']);  

8 |         $leArticle = $this->Articles->get($id);  

9 |         if ($this->Articles->delete($leArticle)) {  

10 |             $this->Flash->success(__("L'article (0) d' id {1} a bien été supprimé ! ",  

11 |                 $leArticle->title, $leArticle->id));  

12 |             return $this->redirect(['action' => 'index']);  

13 |         }  

14 |     }  

15 | }  

16 | }  

17 |

```

- Bouton supprimer

```

39     ?>
40     <?=  

41     $this->Form->postLink(  

42         __('Supprimer'),  

43         ['action' => 'delete', $article->id],  

44         ['confirm' => __("Vraiment supprimer {0} dont l'id vaut {1} ", $article->title, $article->id),  

45         'class'=>'button',  

46         'style'=>'background-color : black' ]);  

47  

48     ?>

```

- Route

```

ArticlesController.php x detail.php x edit.php x index.php x ArticlesTable.php x routes.php x
Source History
46     * Note that `Route` does not do any inflections on URLs which will result in
47     * inconsistently cased URLs when used with `{plugin}`, `{controller}` and
48     * `{action}` markers.
49     */
50     $routes->setRouteClass(DashedRoute::class);
51
52     $routes->scope('/', function (RouteBuilder $builder): void {
53         /*
54          * Here, we are connecting '/' (base path) to a controller called 'Pages',
55          * its action called 'display', and we pass a param to select the view file
56          * to use (in this case, templates/Pages/home.php)...
57          */
58         //$builder->connect('/', ['controller' => 'Pages', 'action' => 'display', 'home']);
59         $builder->connect('/', ['controller' => 'articles', 'action' => 'index']);
60
61         /*

```

Creation utilisateur

- Migration users

```

Creating file C:\wamp64\www\blog_app\config\Migrations\20251008064951_CreateUsers.php
Wrote 'C:\wamp64\www\blog_app\config\Migrations\20251008064951_CreateUsers.php'

```

```
C:\wamp64\www\blog_app\bin>cake migrations migrate
```

```

== 20251008064951 CreateUsers: migrating
== 20251008064951 CreateUsers: migrated 0.7236s
All Done. Took 2.5194s

```

Modification de la table

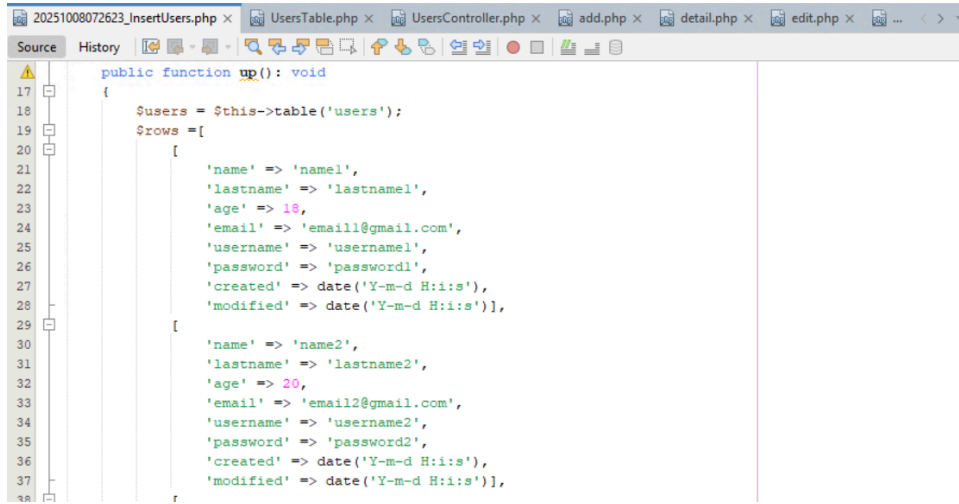
```
C:\wamp64\www\blog_app\bin>cake bake migration AlterArticlesAddFKUser
```

```

Creating file C:\wamp64\www\blog_app\config\Migrations\20251029064255_AlterArticlesAddFKUser.php
Wrote 'C:\wamp64\www\blog_app\config\Migrations\20251029064255_AlterArticlesAddFKUser.php'

```

- Insérer 5 user



```
public function up(): void
{
    $users = $this->table('users');
    $rows = [
        [
            'name' => 'name1',
            'lastname' => 'lastname1',
            'age' => 18,
            'email' => 'email1@gmail.com',
            'username' => 'username1',
            'password' => 'password1',
            'created' => date('Y-m-d H:i:s'),
            'modified' => date('Y-m-d H:i:s')],
        [
            'name' => 'name2',
            'lastname' => 'lastname2',
            'age' => 20,
            'email' => 'email2@gmail.com',
            'username' => 'username2',
            'password' => 'password2',
            'created' => date('Y-m-d H:i:s'),
            'modified' => date('Y-m-d H:i:s')],
    ];
}
```

- Index

```
public function up(): void
{
    $articles = $this->table('users');
    $rows = [
        ['name' => 'name',
        'lastname' => 'lastnam'],
        'age' => 'age',
        'email' => 'email',
        'username' => 'username',
        'password' => 'password',
        'created' => 'created',
        'modified' => 'modified'
    ];
    $articles->insert($rows)->saveData();
}
```

- validation

```
...hp index.php x view.php x UsersController.php x 20251008072623_InsertUsers.php x UsersTable.php x
Source History
12 return $hasher->hash($password);
13 }
14 public function initialize(array $config): void {
15     $this->addBehavior('Timestamp');
16 }
17 public function ValidationDefault(Validator $validator):Validator {
18     $validator
19         ->notEmptyString('username', __('username doit être non vide'))
20         ->notEmptyString('username', __('username doit être unique'))
21         ->notEmptyString('name', __('doit être non vide'))
22         ->notEmptyString('lastname', __('doit être non vide'))
23         ->notEmptyString('age', __('doit être non vide'))
24         ->notEmptyString('email', __('doit être au format email'))
25         ->notEmptyString('email', __('email doit être unique'));
26
27     return $validator;
28 }
29 }
30 }
```

- edit

```
edit.php x index.php x view.php x UsersController.php x 20251008072623_InsertUsers.php x UsersTable.php x
Source History
<h1>Modifier l'utilisateur "<?=$leUser->username?>" (id = <?=$leUser->id ?>)</h1>
2 <?php
3     echo $this->Form->create($leUser);
4     echo $this->Form->control('name');
5     echo $this->Form->control('lastname');
6     echo $this->Form->control('age');
7     echo $this->Form->control('email');
8     echo $this->Form->control('username');
9     echo $this->Form->control('password');
10    echo $this->Form->button(__('Mettre à jour l'utilisateur'));
11    echo $this->Form->end();
12 ?>
13 <br/>
14 <?=$this->html->link('Retour aux utilisateurs',
15     ['controller' => 'users', 'action' => 'index'],
16     ['class' => 'button']);
17 ?>
```

- Index

```
edit.php x index.php x view.php x UsersController.php x 20251008072623_InsertUsers.php x UsersTable.php x <> v
Source History
7 ?>
8
9 <table>
10 <tr>
11 <th>Id</th>
12 <th>Name</th>
13 <th>LastName</th>
14 <th>Age</th>
15 <th>Email</th>
16 <th>Password</th>
17 <th>Date de cr eacut;ation</th>
18 <th>Date de modification</th>
19 <th>Actions</th>
20 </tr>
21
22 <!-- Ici se trouve l'it eration sur l'objet query de notre $mesArticles, l'affichage des infos d
23 <?php foreach ($mesUsers as $user): ?>
24 <tr>
25 <td><?= $user->id ?></td>
26 <td><?= $user->name ?></td>
27 <td><?= $user->lastname ?></td>
28 <td><?= $user->age ?></td>
29 <td><?= $user->username ?></td>
30 <td><?= $user->password ?></td>
31 <td><?= $user->created->format (DATE_RFC850) ?></td>
32 <td><?= $user->modified->format (DATE_RFC850) ?></td>
33 <td>
34
```

- View

```
edit.php x index.php x view.php x UsersController.php x 20251008072623_InsertUsers.php x UsersTable.php x < >
Source History
<h1>Détail de l'utilisateur <?= h($leUser->id) ?></h1>
2
3 <p>
4 <small>Created : <?= $leUser->created->format(DATE_RFC850) ?></small><br/>
5 <small>Modified : <?= $leUser->modified->format(DATE_RFC850) ?></small>
6 </p>
7
8 <table>
9 <tr>
10 <th>Champs</th>
11 <th>Valeur</th>
12 </tr>
13
14 <?php foreach ($leUser->toArray() as $userKey => $userValue): ?>
15 <tr>
16 <td><?= $userKey ?></td><td><?= $userValue ?></td>
17 </tr>
18 <?php endforeach; ?>
19 </table>
20
21 <?=
22 $this->html->link('Retour à la liste des utilisateurs', [
23 'controller' => 'users',
24 'action' => 'index']);
25 ?>
```

- N'affiche pas le mot de passe

```
use Cake\Auth\DefaultPasswordHasher as AuthDefaultPasswordHasher;

protected function _setPassword(string $password): ?string {
    $hasher = new AuthDefaultPasswordHasher();
    return $hasher->hash($password);
}
```

```

7  ?>
8
9  <table>
10 <tr>
11 <th>Id</th>
12 <th>Name</th>
13 <th>LastName</th>
14 <th>Age</th>
15 <th>Email</th>
16 <th>Date de création</th>
17 <th>Date de modification</th>
18 <th>Actions</th>
19 </tr>
20
21 <!-- Ici se trouve l'itération sur l'objet query de notre $mesArticles, l'affichage des infos c
22 <?php foreach ($mesUsers as $user): ?>
23 <tr>
24 <td><?= $user->id ?></td>
25 <td><?= $user->name ?></td>
26 <td><?= $user->lastname ?></td>
27 <td><?= $user->age ?></td>
28 <td><?= $user->username ?></td>
29 <td><?= $user->created->format (DATE_RFC850) ?></td>
30 <td><?= $user->modified->format (DATE_RFC850) ?></td>
31 <td>
32 <?=
33 $this->html->link('Modifier',

```

Pour un pulgique authentication

```

C:\wamp64\www\blog_app>composer require "cakephp/authentication:^3.3"
./composer.json has been updated
Running composer update cakephp/authentication
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
 - Locking cakephp/authentication (3.3.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
 - Installing cakephp/authentication (3.3.2): Extracting archive
1 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
58 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.

```

Mettre a l'exterieur de bin pour l'installation

Pour l'entity

C:\wamp64\www\blog_app\bin>cake bake model Users

```

C:\wamp64\www\blog_app\bin>cake bake model Users
One moment while associations are detected.

Baking table class for Users...

File 'C:\wamp64\www\blog_app\src\Model\Table\UsersTable.php' exists
Do you want to overwrite? (y/n/a/q)
[n] > n
Skip 'C:\wamp64\www\blog_app\src\Model\Table\UsersTable.php'

Deleted 'C:\wamp64\www\blog_app\src\Model\Table\.gitkeep'

Baking entity class for User...

Creating file C:\wamp64\www\blog_app\src\Model\Entity\User.php
Wrote 'C:\wamp64\www\blog_app\src\Model\Entity\User.php'
Deleted 'C:\wamp64\www\blog_app\src\Model\Entity\.gitkeep'

Baking test fixture for Users...

Creating file C:\wamp64\www\blog_app\tests\Fixture\UsersFixture.php
Wrote 'C:\wamp64\www\blog_app\tests\Fixture\UsersFixture.php'
Deleted 'C:\wamp64\www\blog_app\tests\Fixture\.gitkeep'
Bake is detecting possible fixtures...

Baking test case for App\Model\Table\UsersTable ...

Creating file C:\wamp64\www\blog_app\tests\TestCase\Model\Table\UsersTableTest.php
Wrote 'C:\wamp64\www\blog_app\tests\TestCase\Model\Table\UsersTableTest.php'
Done

```

Ajouté un utilisateur

```

<h1>Ajouter un post</h1>
2 <?php
3 echo $this->Form->create($leNewUser);
4 echo $this->Form->control('name');
5 echo $this->Form->control('lastname');
6 echo $this->Form->control('age');
7 echo $this->Form->control('email');
8 echo $this->Form->control('username');
9 echo $this->Form->control('password');
10 echo $this->Form->control('content', ['rows' => '3']);
11 echo $this->Form->button(__("Sauvegarder le utilisateur"));
12 echo $this->Form->end();
13 ?>
14 <?=
15 $this->html->link('retour a la liste des utilisateur', [
16     'controller' => 'Users', 'action' => 'index',
17     ['class'=>'button']);
18 //l'url généré sera de la forme /articles/detail/...
19 ?>

```

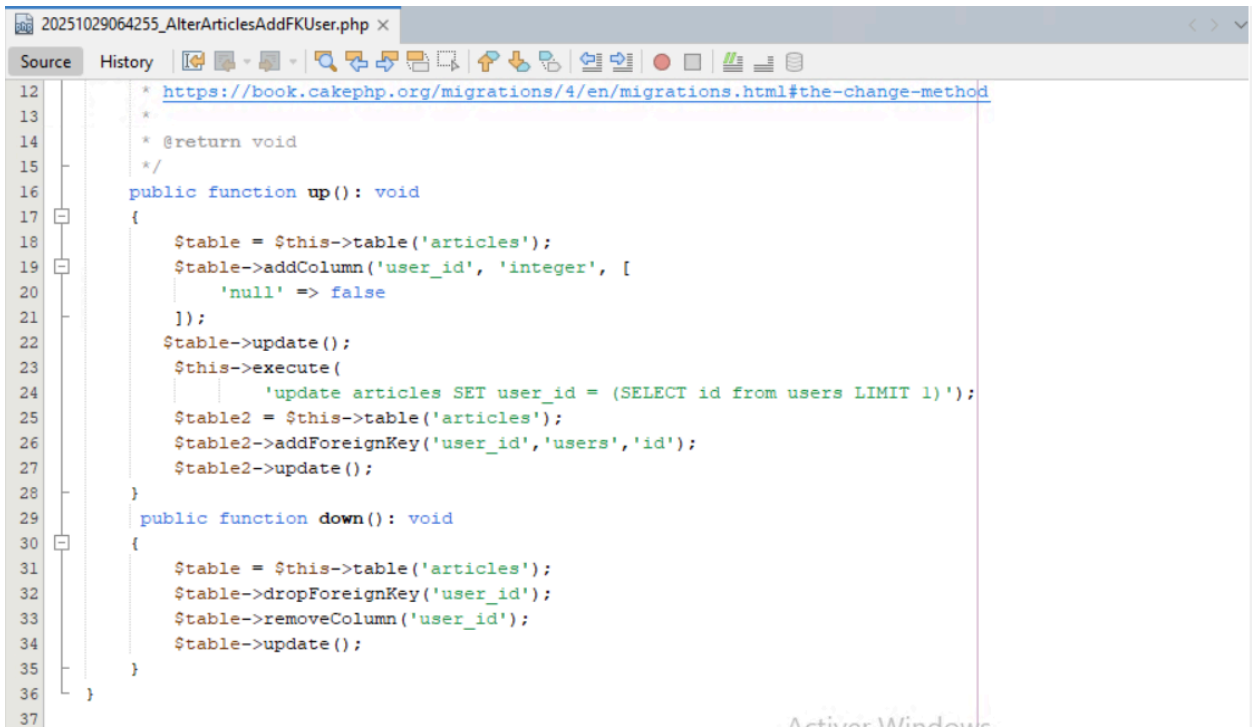
Modifier de migration et l'insertion d'un champs

```
C:\wamp64\www\blog_app\bin>cake migrations migrate

== 20251008072623 InsertUsers: migrating
== 20251008072623 InsertUsers: migrated 0.0198s

== 20251029064255 AlterArticlesAddFKUser: migrating
SQLSTATE[21000]: Cardinality violation: 1242 Subquery returns more than 1 row
Query: update articles SET user_id = (SELECT id from users LIMIT 1);
```

Création de la table articlesAddfkuser clé étrangère



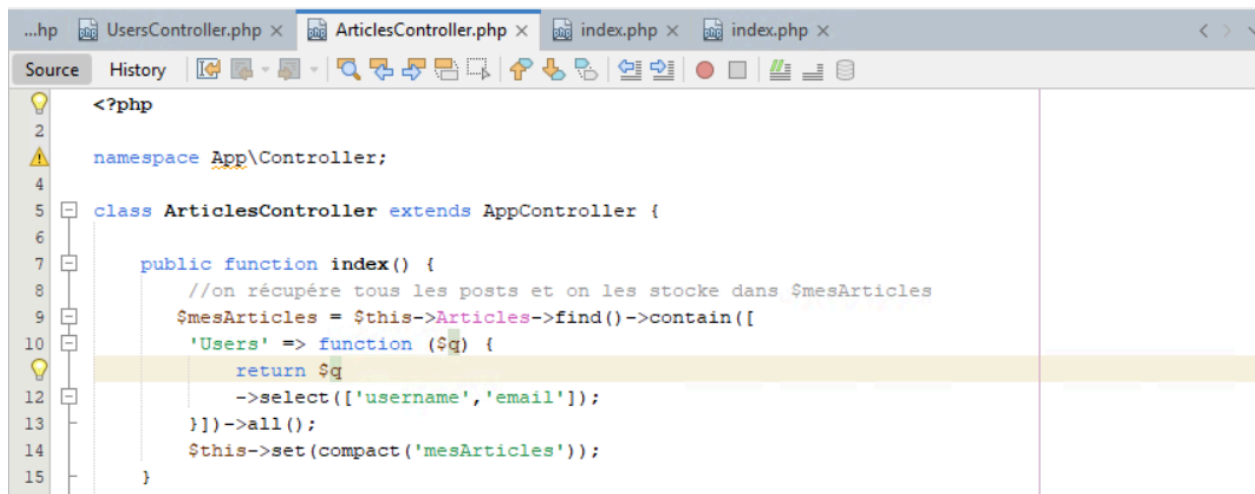
```
20251029064255_AlterArticlesAddFKUser.php x
Source History
12 * https://book.cakephp.org/migrations/4/en/migrations.html#the-change-method
13 *
14 * @return void
15 */
16 public function up(): void
17 {
18     $table = $this->table('articles');
19     $table->addColumn('user_id', 'integer', [
20         'null' => false
21     ]);
22     $table->update();
23     $this->execute(
24         'update articles SET user_id = (SELECT id from users LIMIT 1)';
25     $table2 = $this->table('articles');
26     $table2->addForeignKey('user_id', 'users', 'id');
27     $table2->update();
28 }
29 public function down(): void
30 {
31     $table = $this->table('articles');
32     $table->dropForeignKey('user_id');
33     $table->removeColumn('user_id');
34     $table->update();
35 }
36 }
37
```

Dans la fonction index de ArticlesController, on veut récupérer aussi les users des articles comme ceci :



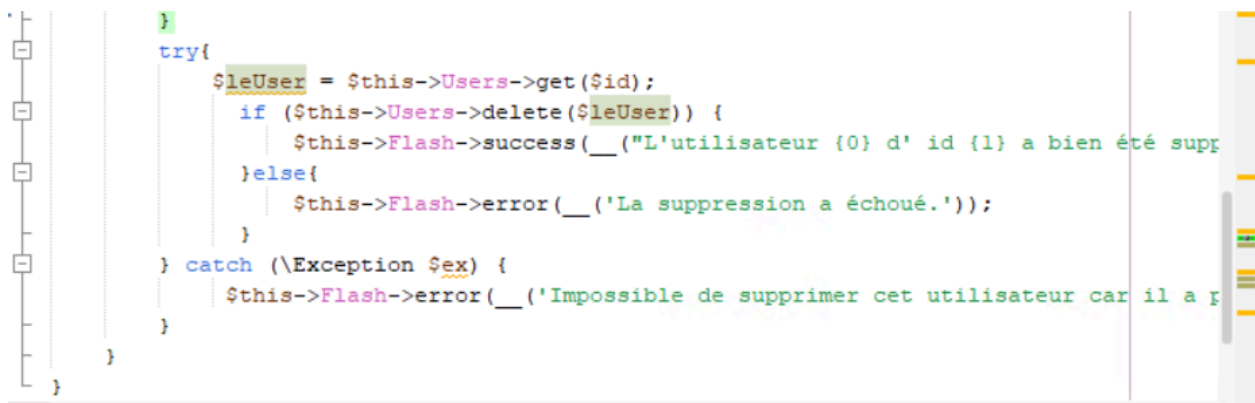
```
ArticlesController.php x UsersTable.php x ArticlesTable.php x index.php x
Source History
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends AppController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->contain(['Users'])->all();
10        $this->set(compact('mesArticles'));
11    }
12 }
```

En faisant cela, la jointure va prendre tous les champs de la table Users. On peut restreindre les champs à afficher comme ceci :



```
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends ApplicationController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->contain([
10             'Users' => function ($q) {
11                 return $q
12                     ->select(['username', 'email']);
13             }->all();
14         $this->set(compact('mesArticles'));
15     }
16 }
```

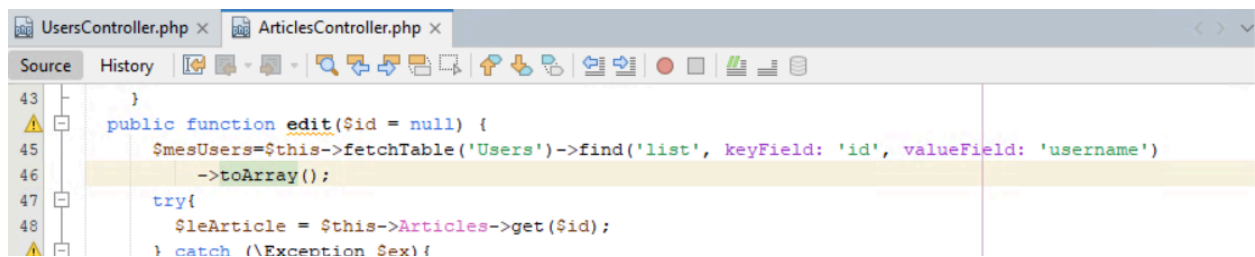
- Association avec le Users



```
    }
    try{
        $leUser = $this->Users->get($id);
        if ($this->Users->delete($leUser)) {
            $this->Flash->success(__("L'utilisateur {} d' id {} a bien été supprimé"));
        }else{
            $this->Flash->error(__("La suppression a échoué."));
        }
    } catch (\Exception $ex) {
        $this->Flash->error(__("Impossible de supprimer cet utilisateur car il a déjà des articles"));
    }
}
```

La suite TP3 :

Inseré une table



```
43     }
44     public function edit($id = null) {
45         $mesUsers=$this->fetchTable('Users')->find('list', keyField: 'id', valueField: 'username')
46         ->toArray();
47         try{
48             $leArticle = $this->Articles->get($id);
49         } catch (\Exception $ex){
50         }
```

- edite

```
...hp edit.php x ArticlesTable.php x index.php x detail.php x add.php x UsersTable.php x
Source History
<h1>Modifier l'article "<?=$leArticle->title?>" (id = <?=$leArticle->id ?></h1>
2 <?php
3     echo $this->Form->create($leArticle);
4     echo $this->Form->control('title');
5     echo $this->Form->control('content', ['rows' => '3']);
6     echo $this->Form->control('user_id',
7         ['options' => $mesUsers, 'label' => 'Selectionnez un user']);
8     echo $this->Form->button(__('Mettre à jour l'article'));
9     echo $this->Form->end();
10 ?>
11 <br/>
12 <?=$this->html->link('Retour aux articles',
13     ['controller' => 'articles', 'action' => 'index'],
14     ['class' => 'button']);
15 ?>
```

Rajouté une liste d'utilisateur

- Article table

```
UsersController.php x ArticlesController.php x edit.php x ArticlesTable.php x
Source History
<?php
2 namespace App\Model\Table;
4 use Cake\ORM\Table;
5 use Cake\Validation\Validator;
7 class ArticlesTable extends Table {
9     public function initialize(array $config): void {
11         $this->addBehavior('Timestamp');
12         $this->belongsTo('Users');
13     }
14     public function ValidationDefault(Validator $validator):Validator {
16         $validator
17             ->notEmptyString('title', __('Veuillez renseigner un titre'))
18             ->notEmptyString('content', __('Veuillez renseigner un contenu'))
19             ->notEmptyString('user_id', __('Veuillez renseigner un contenu'));
```

- Faire un commentaire

On a générer le commentaire et modifier

```
C:\wamp64\www\blog_app\bin>cake bake migration CreateComments title:String content:String created article_id:int
Creating file C:\wamp64\www\blog_app\config\Migrations\20251105043057_CreateComments.php
Wrote 'C:\wamp64\www\blog_app\config\Migrations\20251105043057_CreateComments.php'
```

Activer Windows

```
13
14 * @return void
15 */
16 public function change(): void
17 {
18     $table = $this->table('comments');
19     $table->addColumn('title', 'string', [
20         'default' => null,
21         'limit' => 255,
22         'null' => false,
23     ]);
24     $table->addColumn('content', 'string', [
25         'default' => null,
26         'limit' => 255,
27         'null' => false,
28     ]);
29     $table->addColumn('created', 'datetime', [
30         'default' => null,
31         'null' => false,
32     ]);
33     $table->addColumn('article_id', 'integer', [
34         'default' => null,
35         'limit' => 255,
36         'null' => false,
37     ]);
38     $table->create();
39 }
40 }
```

- Puis faire la migration createComments

```
C:\wamp64\www\blog_app\bin>cake migrations migrate

== 20251105043057 CreateComments: migrating
== 20251105043057 CreateComments: migrated 1.6029s
All Done. Took 3.7502s

C:\wamp64\www\blog_app\bin>cake migrations status
```

Status	Migration ID	Migration Name
up	20251001044659	InitialMigration
up	20251001055324	AlterArticles
up	20251001060925	InsertArticles
up	20251008064951	CreateUsers
up	20251008072623	InsertUsers
up	20251029064255	AlterArticlesAddFKUser
up	20251105043057	CreateComments

Si tu a des doute fait cake migrate status pour voir si c'est en up

- Création du table commentaire

```
...hp ArticlesTable.php x CommentsTable.php x ArticlesController.php x CommentsController.php x
Source History
<?php
2
3 namespace App\Model\Table;
4
5 use Cake\ORM\Table;
6 use Cake\Validation\Validator;
7
8 class CommentsTable extends Table {
9
10 public function initialize(array $config): void {
11     $this->addBehavior('Timestamp');
12     $this->belongsTo('Articles');
13 }
14
15 public function ValidationDefault(Validator $validator):Validator {
16     $validator
17         ->notEmptyString('title', __('Veuillez renseigner un titre'))
18         ->notEmptyString('content', __('Veuillez renseigner un contenu'))
19         ->notEmptyString('article_id', __('Veuillez renseigner un contenu'));
20
21     return $validator;
22 }
23
24 }
```

- Rajouté dans la table article

```
...hp ArticlesTable.php x CommentsTable.php x ArticlesController.php x CommentsController.php x
Source History
<?php
2
3 namespace App\Model\Table;
4
5 use Cake\ORM\Table;
6 use Cake\Validation\Validator;
7
8 class ArticlesTable extends Table {
9
10 public function initialize(array $config): void {
11     $this->addBehavior('Timestamp');
12     $this->belongsTo('Users');
13     $this->hasMany('Comments');
14 }
15
16 public function ValidationDefault(Validator $validator):Validator {
17     $validator
18         ->notEmptyString('title', __('Veuillez renseigner un titre'))
19         ->notEmptyString('content', __('Veuillez renseigner un contenu'))
20         ->notEmptyString('user_id', __('Veuillez renseigner un contenu'));
21
22     return $validator;
23 }
24
25 }
```

Activer Windows

Ajouté commentaires dans le controller articles

```
...hp ArticlesTable.php x CommentsTable.php x ArticlesController.php x CommentsController.php x
Source History
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends ApplicationController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->contain([
10             'Users' => function ($q) {
11                 return $q
12                     ->select(['username', 'email']);
13             }, 'Comments']->all());
14         $this->set(compact('mesArticles'));
15     }
16 }
```

Index article

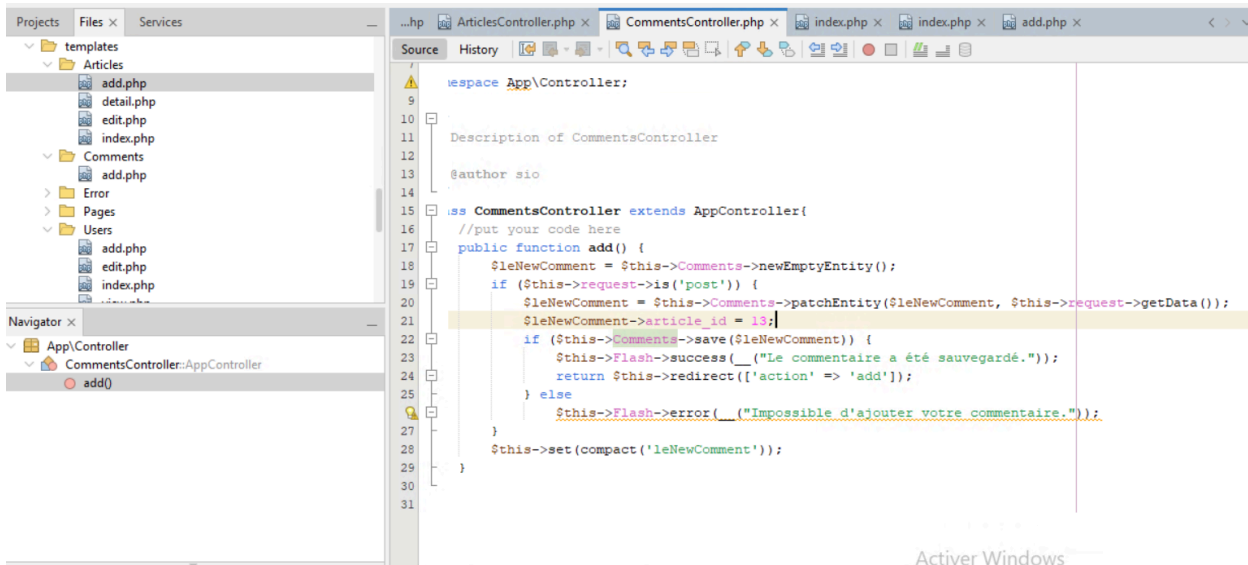
```
30 //l'url généré sera de la forme /articles/detail/...
31     ?>
32     </td>
33     <td><?= $article->user->username ?></td>
34     <td><?= count($article->comments) ?></td>
35     <td><?= $article->created->format (DATE_RFC850) ?></td>
36     <td><?= $article->modified->format (DATE_RFC850) ?></td>
37     <td>
```

```
<table>
  <tr>
    <th>Id</th>
    <th>Titre</th>
    <th>Créé par</th>
    <th>Nb commentaires</th>
    <th>Date de création</th>
    <th>Date de modification</th>
    <th>Actions</th>
```

Controller commentaire et modifier

```
...hp CommentsTable.php x ArticlesController.php x CommentsController.php x index.php x index.php x
Source History
espace App\Controller;
9
10
11 Description of CommentsController
12
13 @author sio
14
15 namespace CommentsController extends ApplicationController{
16 //put your code here
17 public function add() {
18     $leNewComment = $this->Comments->newEmptyEntity();
19     if ($this->request->is('post')) {
20         $leNewComment = $this->Comments->patchEntity($leNewComment, $this->request->getData());
21         if ($this->comments->save($leNewComment)) {
22             $this->Flash->success(__("Le commentaire a été sauvegardé."));
23             return $this->redirect(['action' => 'add']);
24         } else
25             $this->Flash->error(__("Impossible d'ajouter votre commentaire."));
26     }
27     $this->set(compact('leNewComment'));
28 }
29
```

- Rajoute ne nombre de commentaire



```
Projects Files x Services
templates
Articles
add.php
detail.php
edit.php
index.php
Comments
add.php
Error
Pages
Users
add.php
edit.php
index.php
Navigator x
App\Controller
CommentsController:AppController
add()
...hp ArticlesController.php x CommentsController.php x index.php x index.php x add.php x
Source History
espace App\Controller;
9
10
11 Description of CommentsController
12
13 @author sio
14
15 namespace CommentsController extends ApplicationController{
16 //put your code here
17 public function add() {
18     $leNewComment = $this->Comments->newEmptyEntity();
19     if ($this->request->is('post')) {
20         $leNewComment = $this->Comments->patchEntity($leNewComment, $this->request->getData());
21         $leNewComment->article_id = 13;
22         if ($this->comments->save($leNewComment)) {
23             $this->Flash->success(__("Le commentaire a été sauvegardé."));
24             return $this->redirect(['action' => 'add']);
25         } else
26             $this->Flash->error(__("Impossible d'ajouter votre commentaire."));
27     }
28     $this->set(compact('leNewComment'));
29 }
30
31
```

Affiche le nombre de commentaire du ligne 21

```
...hp CommentsTable.php x ArticlesController.php x CommentsController.php x index.php x index.php x ... < > v
Source History [Icons]
<?php
2
3 namespace App\Controller;
4
5 class ArticlesController extends ApplicationController {
6
7     public function index() {
8         //on récupère tous les posts et on les stocke dans $mesArticles
9         $mesArticles = $this->Articles->find()->contain([
10             'Users' => function ($q) {
11                 return $q
12                 ->select(['username','email']);
13             }, 'Comments' => function ($q) {
14                 return $q
15                 ->select(['article_id']);
16             }, 'Comments')->all();
17
18         $this->set(compact('mesArticles'));
19     }
}
```

- Rajouté une ligne dans articlescontroller

```
ArticlesController.php x CommentsController.php x index.php x
Source History [Icons]
13         }, 'Comments' => function ($q) {
14             return $q
15             ->select(['article_id']);
16         }, 'Comments')->all();
17
18         $this->set(compact('mesArticles'));
19     }
20
21     public function detail($id = null) {
22         $comments = $this->fetchTable('Comments');
23         $leNewComment = $comments->newEmptyEntity();
}
```

On renvoie la variable a la vue

```

ArticlesController.php x CommentsController.php x index.php x
Source History
13     }, 'Comments' => function ($q) {
14         return $q
15         ->select(['article_id']);
16     }, 'Comments')->all();
17
18     $this->set(compact('mesArticles'));
19 }
20
21 public function detail($id = null) {
22     $comments = $this->fetchTable('Comments');
23     $leNewComment = $comments->newEmptyEntity();
24     try {
25         $leArticle = $this->Articles->get($id);
26     } catch (\Exception $ex) {
27         if ($id == null) {
28             $this->Flash->error(__("L'action detail doit être appelé avec un identifiant"));
29         } else {
30             $this->Flash->error(__("L'article (0) n'existe pas", $id));
31         }
32         return $this->redirect(['action' => 'index']);
33     }
34
35     $this->set(compact('leArticle', 'leNewComment'));

```

- On ajouter et modifier pour le commentaire qui est demandé

```

24     if ($this->request->is('post')) {
25         $leNewComment = $comments->patchEntity($leNewComment, $this->request->getData());
26         $leNewComment->article_id = $id;
27         if ($comments->save($leNewComment)) {
28             $this->Flash->success(__("Le commentaire a été sauvegardé."));
29             return $this->redirect(['action' => 'détail', $id]);
30         } else {
31             $this->Flash->error(__("Impossible d'ajouter votre commentaire."));
32         }

```

- Récupérer tout les commentaire de l'articles concernée

```

32         $this->Flash->error(__("Impossible d'ajouter votre commentaire."));
33     }
34     try {
35         $leArticle = $this->Articles->get($id,
36             contain: ['Comments' => function ($q) {
37                 return $q
38                 ->order(['Comments.created asc']);
39             }]);

```

- Création du bouton comments

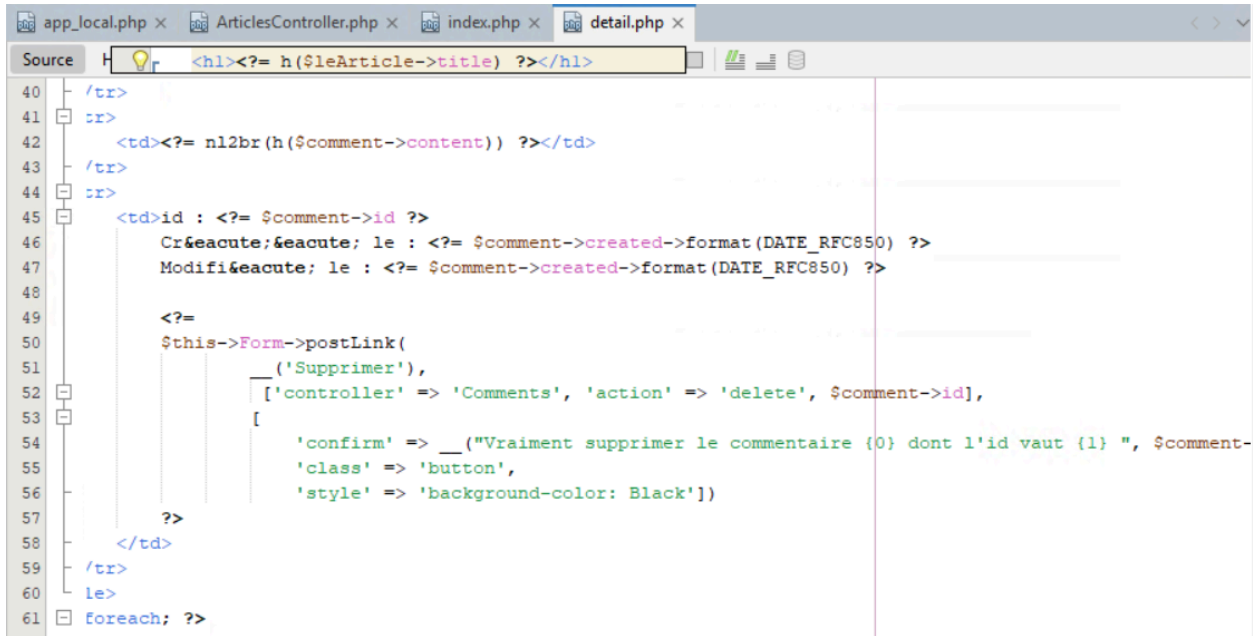
```

...hp ArticlesController.php x index.php x UsersController.php x add.php x detail.php x index.php x
Source History
22     });
23     </script>
24
25     <?=
26     $this->Html->link(
27         'Ajoutez un commentaire', '#', ['class' => 'button', 'id' => 'showcom']
28     );
29     ?>
30

```

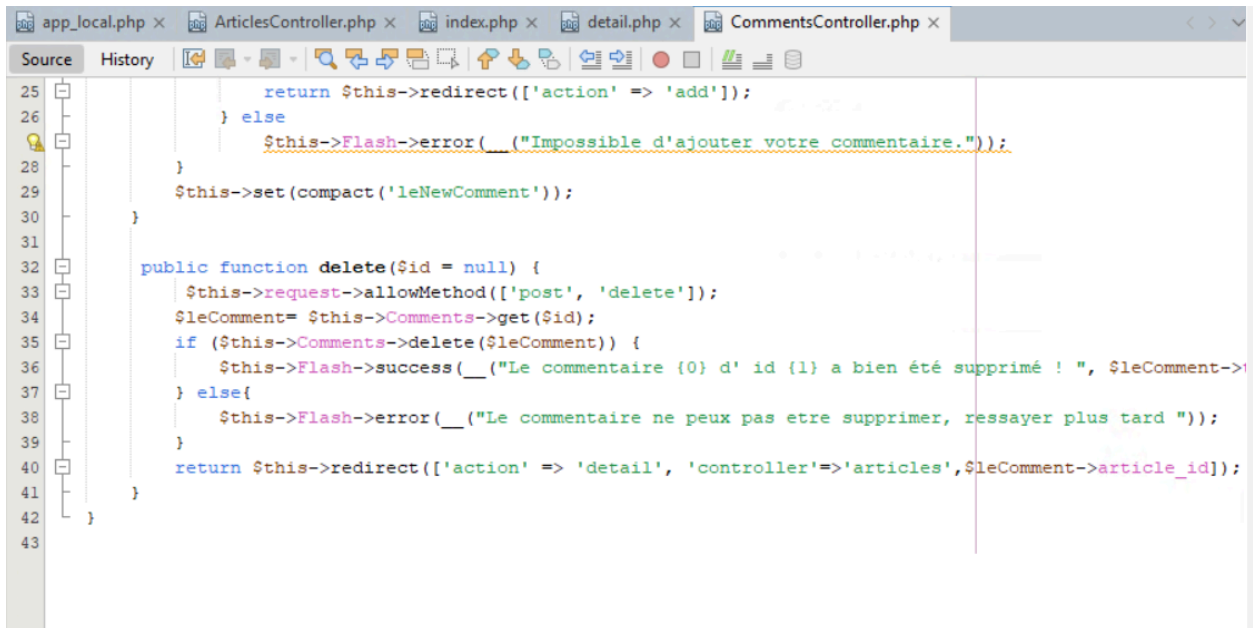
Pour créer des commentaires

- À côté de chaque commentaire créer un bouton supprimer



```
40 /tr>
41 <tr>
42 <td><?= nl2br(h($comment->content)) ?></td>
43 /tr>
44 <tr>
45 <td id : <?= $comment->id ?>
46 Créé le : <?= $comment->created->format(DATE_RFC850) ?>
47 Modifié le : <?= $comment->created->format(DATE_RFC850) ?>
48
49 <?=
50 $this->Form->postLink(
51     ('Supprimer'),
52     ['controller' => 'Comments', 'action' => 'delete', $comment->id],
53     [
54         'confirm' => __("Vraiment supprimer le commentaire {0} dont l'id vaut {1} ", $comment->id, $comment->id),
55         'class' => 'button',
56         'style' => 'background-color: Black; color: white; border: none; padding: 5px 10px;'
57     ]
58 )
59 /tr>
60 </table>
61 foreach; ?>
```

- Le débogage



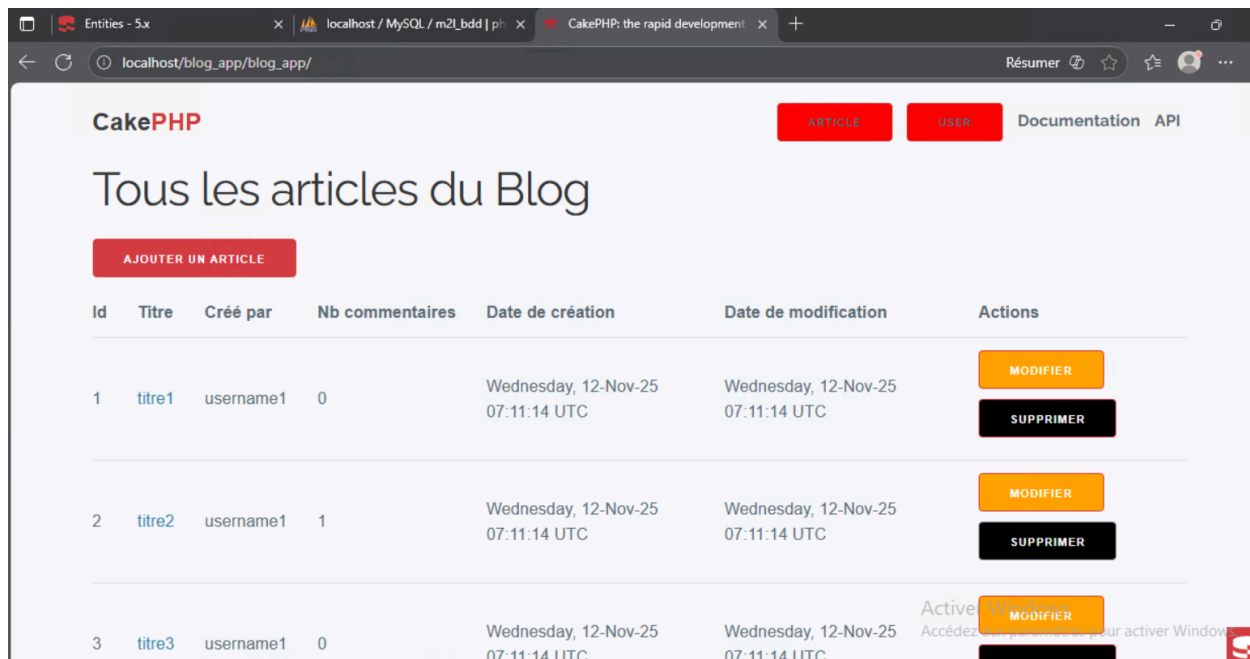
```
25     return $this->redirect(['action' => 'add']);
26 } else
27     $this->Flash->error(__("Impossible d'ajouter votre commentaire."));
28 }
29 $this->set(compact('leNewComment'));
30 }
31
32 public function delete($id = null) {
33     $this->request->allowMethod(['post', 'delete']);
34     $leComment = $this->Comments->get($id);
35     if ($this->Comments->delete($leComment)) {
36         $this->Flash->success(__("Le commentaire {0} d'id {1} a bien été supprimé ! ", $leComment->id, $leComment->id));
37     } else {
38         $this->Flash->error(__("Le commentaire ne peut pas être supprimé, réessayer plus tard "));
39     }
40     return $this->redirect(['action' => 'detail', 'controller' => 'articles', $leComment->article_id]);
41 }
42 }
43 }
```

Création de bouton pour passer d'utilisateur à articles

Bouton pour passer de utilisateur à un articles

En allant sur défaut et mettre un lien de bouton

```
app_local.php x default.php x index.php x default.php x
Source History
43 $this->html->link('Article',
44     ['controller' => 'articles', 'action' => 'index'],
45     [
46         'class' => 'button',
47         'style' => 'background-color: red']);
48 ?>
49 <?php
50 $this->html->link('users',
51     ['controller' => 'users', 'action' => 'index'],
52     [
53         'class' => 'button',
54         'style' => 'background-color: red']);
55 ?>
```



Creation des icones et réglage de la taille de l'icone





```
app_local.php x index.php x
Source History
52
53
54 <?=  
55     $this->Html->image('add.png', [  
56         'alt' => 'add',  
57         'width' => '40',  
58         'height' => '40',  
59         'class' => 'icon-sm'  
60     ])  
61 >?  
62  
63 <?=  
64     $this->Html->image('fountain-pen.png', [  
65         'alt' => 'fountain-pen',  
66         'width' => '40',  
67         'height' => '40',  
68         'class' => 'icon-sm'  
69     ])  
70 >?  
71
```

<?=
?>

```
$this->Html->image('fountain-pen.png', [  
    'alt' => 'fountain-pen',  
    'width' => '40',  
    'height' => '40',  
    'class' => 'icon-sm'  
])  
?>
```

Tous les articles du blog





AJOUTER UN ARTICLE

Id	Titre	Créé par	Nb commentaires	Date de création	Date de modification	Actions
1	titre1	username1	0	Wednesday, 12-Nov-25 07:11:14 UTC	Wednesday, 12-Nov-25 07:11:14 UTC	MODIFIER SUPPRIMER  
2	titre2	username1	1	Wednesday, 12-Nov-25 07:11:14 UTC	Wednesday, 12-Nov-25 07:11:14 UTC	MODIFIER SUPPRIMER  

Tous les utilisateurs du Blog

AJOUTER UN UTILISATEUR

UN UTILISATEUR

Id	Name	LastName	Nbarticle	Age	Email	Date de création	Date de modification	Actions
1	name1	lastname1	18	username1	5	Wednesday, 12-Nov-25 07:11:15 UTC	Wednesday, 12-Nov-25 07:11:15 UTC	<p>MODIFIER</p> <p>SUPPRIMER</p>  
2	name2	lastname2	20	username2	0	Wednesday, 12-Nov-25 07:11:15 UTC	Wednesday, 12-Nov-25 07:11:15 UTC	<p>MODIFIER</p> <p>SUPPRIMER</p>  

3 eme création deAuthentification

Sécurisation :

Appcontroller :

```
namespace App\Controller;  
use Cake\Event\Event;
```

```
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

usersController: permet de se connecter et deconnectée

```
AppController.php x UsersController.php x login.php x
Source History
public function beforeFilter(\Cake\Event\EventInterface $event)
83 {
84     parent::beforeFilter($event);
85     // Allow users to register and logout.
86     // You should not add the "login" action to allow list. Doing so would
87     // cause problems with normal functioning of AuthComponent.
88     $this->Auth->allow(['add', 'logout']);
89 }
90
91 public function login()
92 {
93     if ($this->request->is('post')) {
94         $user = $this->Auth->identify();
95         if ($user) {
96             $this->Auth->setUser($user);
97             return $this->redirect($this->Auth->redirectUrl());
98         }
99         $this->Flash->error(__('Mauvais username ou mot de passe, veuillez réessayer'));
100     }
101 }
102
103 public function logout()
104 {
105     return $this->redirect($this->Auth->logout());
106 }
107 }
```

src/Template/Users/login.php

```
AppController.php x UsersController.php x login.php x
Source History
<div class="users form">
2 <?=$this->Flash->render('auth') ?>
3 <?=$this->Form->create() ?>
4 <fieldset>
5 <legend><?=__("Merci de rentrer vos nom d'utilisateur et mot de passe") ?></legend>
6 <?=$this->Form->input('username') ?>
7 <?=$this->Form->input('password') ?>
8 </fieldset>
9 <?=$this->Form->button(__('Se connecter')); ?>
10 <?=$this->Form->end() ?>
11 </div>
12
```

Authentication

```

C:\wamp64\www\blog_app\blog_app>composer require cakephp/authentication
./composer.json has been updated
Running composer update cakephp/authentication
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 1 update, 0 removals
  - Upgrading cakephp/authentication (3.3.2 => 3.3.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 1 update, 0 removals
  - Downloading cakephp/authentication (3.3.3)
  - Upgrading cakephp/authentication (3.3.2 => 3.3.3): Extracting archive
Generating autoload files
58 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!

```

Suivre la doc cakephp version 3.x pour la création de login poue se connecter

src/Application.php

```

use Authentication\AuthenticationService;
use Authentication\AuthenticationServiceInterface;
use Authentication\AuthenticationServiceProviderInterface;
use Authentication\Identifier\AbstractIdentifier;
use Authentication\Identifier\IdentifierInterface;
use Authentication\Middleware\AuthenticationMiddleware;
use Cake\Http\MiddlewareQueue;
use Cake\Routing\Router;
use Psr\Http\Message\ServerRequestInterface;

```

```

public function middleware(MiddlewareQueue $middlewareQueue): MiddlewareQueue
{
    $middlewareQueue->add(new ErrorHandlerMiddleware(Configure::read('Error')))
    // Other middleware that CakePHP provides.
    ->add(new AssetMiddleware())
    ->add(new RoutingMiddleware($this))
    ->add(new BodyParserMiddleware())

    // Add the AuthenticationMiddleware. It should be
    // after routing and body parser.
    ->add(new AuthenticationMiddleware($this));

    return $middlewareQueue;
}

```

AuthenticationMiddleware: src/Application.php:

```

public function
getAuthenticationService(ServerRequestInterface
$request): AuthenticationServiceInterface
{
    $service = new AuthenticationService();

```

```
// Define where users should be redirected to when
they are not authenticated
$service->setConfig([
    'unauthenticatedRedirect' => [
        'prefix' => false,
        'plugin' => false,
        'controller' => 'Users',
        'action' => 'login',
    ],
    'queryParam' => 'redirect',
]);

// Define identifiers
$fields = [
    AbstractIdentifier::CREDENTIAL_USERNAME =>
'email',
    AbstractIdentifier::CREDENTIAL_PASSWORD =>
'password'
];

$passwordIdentifier = [
    'Authentication.Password' => [
        'fields' => $fields,
    ],
];

// Load the authenticators. Session should be
first.

$service->loadAuthenticator('Authentication.Session', [
    'identifier' => $passwordIdentifier,
```

```
]);  
$service->loadAuthenticator('Authentication.Form',  
[  
    'identifier' => $passwordIdentifier,  
    'fields' => $fields,  
    'loginUrl' => Router::url([  
        'prefix' => false,  
        'plugin' => null,  
        'controller' => 'Users',  
        'action' => 'login',  
    ]),  
]);
```

```
    return $service;  
}
```

```
// in src/Controller/AppController.php  
public function initialize()  
{  
    parent::initialize();
```

```
$this->loadComponent('Authentication.Authentication');  
}
```

```
// in src/Controller/UsersController.php  
public function login()  
{  
    $result = $this->Authentication->getResult();  
    // If the user is logged in send them away.  
    if ($result && $result->isValid()) {
```

```
        $target =
$this->Authentication->getLoginRedirect() ?? '/';
        return $this->redirect($target);
    }
    if ($this->request->is('post')) {
        $this->Flash->error('Invalid username or
password');
    }
}
```

```
public function beforeFilter(\Cake\Event\EventInterface
$event)
{
    parent::beforeFilter($event);
```

```
$this->Authentication->allowUnauthenticated(['login']);
}
```

```
// in templates/Users/login.php
<div class="users form content">
    <?= $this->Form->create() ?>
    <fieldset>
        <legend><?= __('Please enter your email and
password') ?></legend>
        <?= $this->Form->control('email') ?>
        <?= $this->Form->control('password') ?>
    </fieldset>
```

```
<?= $this->Form->button(__('Login')) ?>
<?= $this->Form->end() ?>
</div>
```

```
// in src/Controller/UsersController.php
public function logout()
{
    $this->Authentication->logout();
    return $this->redirect(['controller' => 'Users',
    'action' => 'login', 'add']);
}
```

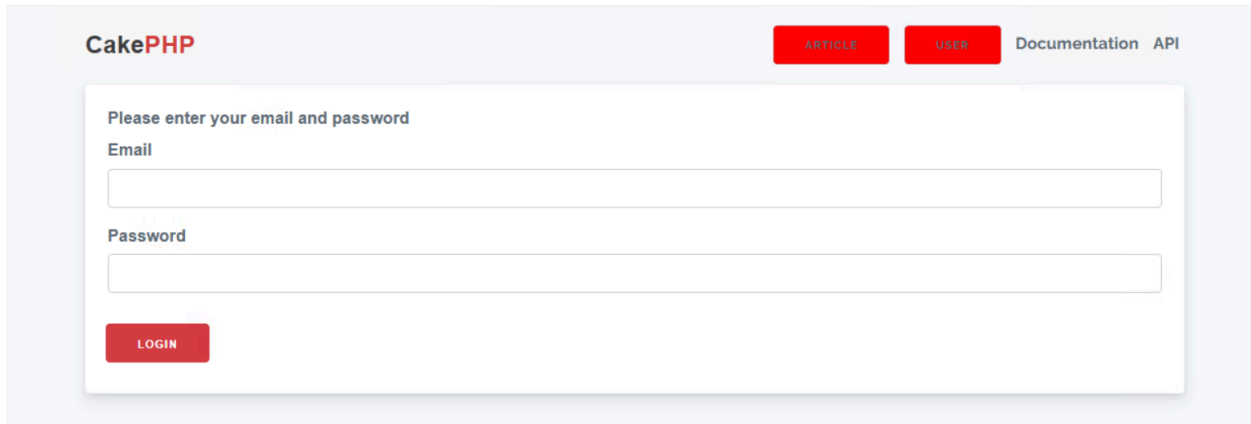
```
// in src/Model/Entity/User.php
use
Authentication\PasswordHasher\DefaultPasswordHasher;

class User extends Entity
{
    // ... other methods

    // Automatically hash passwords when they are
    changed.
    protected function _setPassword(string $password)
    {
        $hasher = new DefaultPasswordHasher();
        return $hasher->hash($password);
    }
}
```

Sans les code précédent l'authentification marche pas :

Resultat :

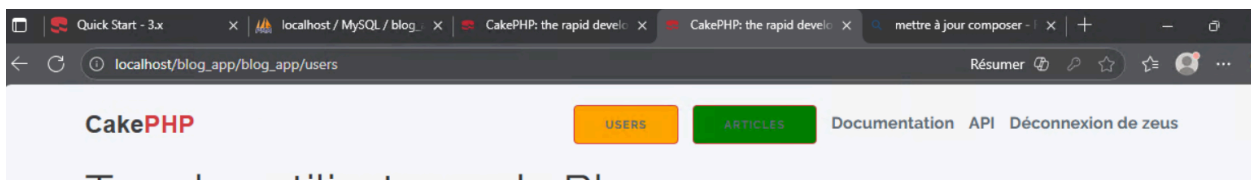


Autorisation pour se connecter

Modification du login dans le controlleur usercontroller

```
login.php x default.php x edit.php x AppController.php x add.php x UsersController.php x UsersTable.php x
Source History
70 $leUser = $this->Users->get($id);
71 try {
72     if ($this->Users->delete($leUser)) {
73         $this->Flash->success(__("L'utilisateur {0} d' id {1} a bien été supprimé ! ", $leUser->
74     } else {
75         $this->Flash->error(__("L'utilisateur ne peut pas être supprimé, réessayez plus tard"));
76     }
77 } catch (\Exception $ex) {
78     $this->Flash->error(__("L'utilisateur ne peut pas être supprimé, {0}", $ex->getMessage()));
79 }
80 return $this->redirect(['action' => 'index']);
81 }
82 public function beforeFilter(\Cake\Event\EventInterface $event)
83 {
84     parent::beforeFilter($event);
85     $this->Authentication->allowUnauthenticated(['login', 'add']);
86 }
87 }
88 }
```

Pour afficher la connection et la déconnection de l'utilisateur.



Btn inscription

```
login.php x default.php x edit.php x ApplicationController.php x add.php x UsersController.php x index.php...
Source History
38         'class' => 'boutton',
39         'style' => 'background-color: #2779F5'
40     })
41 }
42
43
44 </div>
45 <div class="form-group text-center">
46 <?php
47     $this->Html->link(
48         'Inscription',
49         ['controller' => 'Users', 'action' => 'add'],
50         ['class' => 'btn btn-primary']
51     )
52 >?>
```

Debug le controller articles :

```
ArticlesController.php x CommentsTable.php x UsersTable.php x 20251126051304_AlterCommentsAddFKUser.php x detail...
Source History
76 public function edit($id = null) {
77     // $mesUsers = $this->fetchTable('Users')
78     //     ->find('list', keyField: 'id', valueField: 'username')
79     //     ->toArray();
80     try {
81         $leArticle = $this->Articles->get($id);
82     } catch (\Exception $ex) {
83         if ($id == null) {
84             $this->Flash->error(__("L'action edit doit être appelé avec un identifiant"));
85         } else {
86             $this->Flash->error(__("L'article {0} n'existe pas", $id));
87         }
88         return $this->redirect(['action' => 'index']);
89     }
90
91     if ($this->request->is(['post', 'put'])) {
92         $this->Articles->patchEntity($leArticle, $this->request->getData());
93         $leArticle->user_id = $this->request->getAttribute('identity')->id;
94         if ($this->Articles->save($leArticle)) {
95             $this->Flash->success(__("Votre article a été mis à jour.));
96             return $this->redirect(['action' => 'index']);
97         } else {
98             $this->Flash->error(__("Impossible de mettre à jour votre article.));
99         }
100     }
101     $this->set(compact('leArticle'));
}
```

Activer Windows

Modification de l'articles controller :

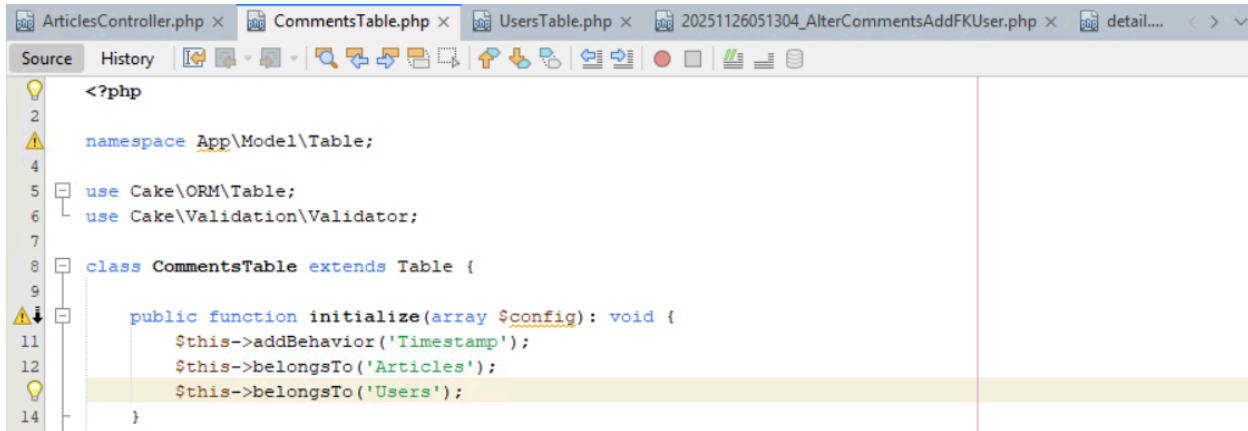
```
app_local.php x ArticlesController.php x
Source History
43         $this->Flash->error(__("L'article {0} n'existe pas", $id));
44     }
45     return $this->redirect(['action' => 'index']);
46 }
47
48     $this->set(compact('leArticle', 'leNewComment'));
49 }
50
51 public function add() {
52     $leNewArticle = $this->Articles->newEmptyEntity();
53     if ($this->request->is('post')) {
54         $leNewArticle = $this->Articles->patchEntity($leNewArticle, $this->request->getData());
55         //on recupere l'id du user connecter et on le rajoute en tant que clé étrangère
56         $leNewArticle->user_id = $this->request->getAttribute('identity')->id;
57         if ($this->Articles->save($leNewArticle)) {
58             $this->Flash->success(__("L'article a été sauvegardé."));
59             return $this->redirect(['action' => 'index']);
60         } else
61             $this->Flash->error(__("Impossible d'ajouter votre article."));
62     }
63     $this->set(compact('leNewArticle'));
64 }
```

Sa permet de rajouté un articles en tant que utilisateur connecter sans besoins de son id.

Jointure de modele

- Lier les modèles : users est comments

```
ArticlesController.php x CommentsTable.php x UsersTable.php x 20251126051304_AlterCommentsAddFKUser.php x detail...
Source History
11 public function initialize(array $config): void {
12     $this->addBehavior('Timestamp');
13     $this->hasMany('Articles');
14     $this->hasMany('comments');
15 }
```



```
<?php
2
3 namespace App\Model\Table;
4
5 use Cake\ORM\Table;
6 use Cake\Validation\Validator;
7
8 class CommentsTable extends Table {
9
10     public function initialize(array $config): void {
11         $this->addBehavior('Timestamp');
12         $this->belongsTo('Articles');
13         $this->belongsTo('Users');
14     }
15 }
```

Modifier la base de données

```
C:\wamp64\www\blog_app\blog_app\bin>cake migrations migrate
== 20251126051304 AlterCommentsAddFKuser: migrating
== 20251126051304 AlterCommentsAddFKuser: migrated 0.1759s
All Done. Took 0.2802s
```

```
C:\wamp64\www\blog_app\blog_app\bin>cake migrations rollback
== 20251126051304 AlterCommentsAddFKuser: reverting
== 20251126051304 AlterCommentsAddFKuser: reverted 0.0435s
All Done. Took 0.1470s
```

```
C:\wamp64\www\blog_app\blog_app\bin>cake migrations migrate
== 20251126051304 AlterCommentsAddFKuser: migrating
```

```
...hp detail.php x index.php x CommentsTable.php x UsersTable.php x 20251126051304_AlterCommentsAddFKUser.php x < > v
Source History
13 *
14 * @return void
15 */
16 public function up(): void
17 {
18     $table = $this->table('comments');
19     $table->addColumn('user_id', 'integer', [
20         'null' => false
21     ]);
22     $table->update();
23     $this->execute(
24         'update comments SET user_id = (SELECT id from users LIMIT 1)');
25     $table2 = $this->table('comments');
26     $table2->addForeignKey('user_id', 'users', 'id');
27     $table2->update();
28 }
29 public function down(): void
30 {
31     $table = $this->table('comments');
32     $table->dropForeignKey('user_id');
33     $table->removeColumn('user_id');
34     $table->update();
35 }
36 }
37 }
```

Activer Windows

Modifier le controller articles

Fonction detail prendre en compte user_id

```
ArticlesController.php x CommentsTable.php x UsersTable.php x 20251126051304_AlterCommentsAddFKUser.php x detail... < > v
Source History
31 } else
32     $this->Flash->error(__("Impossible d'ajouter votre commentaire.));
33 }
34 try {
35     $leArticle = $this->Articles->get($id,
36         contain: ['Comments.Users'=> function ($q) {
37             return $q
38                 ->select(['Users.username'])
39                 ->order(['Comments.created asc']);
40         },
41         'Users'=> function ($q) {
42             return $q
43                 ->select(['Users.username']);
44         },
45     );
46 }
47 }
48 ]);
```

Autorisation :

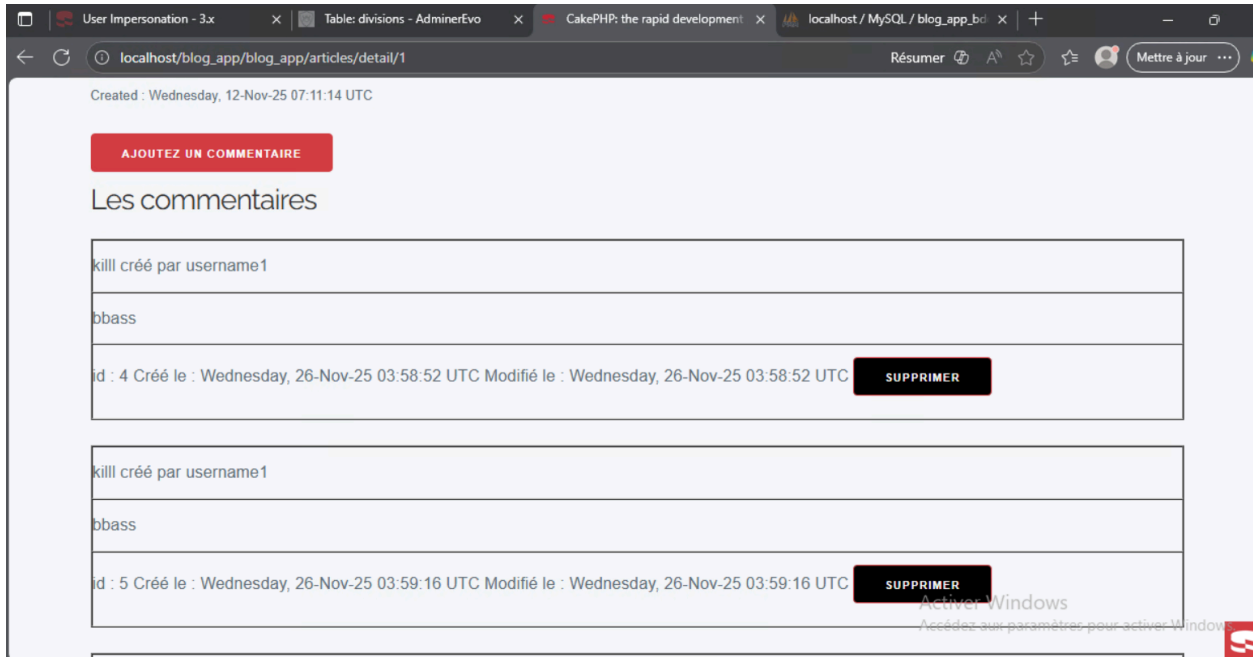
```
index.php x default.php x ArticlesController.php x AppController.php x
Source History
36 <td>
37 <?php
38 $identity = $this->request->getAttribute('identity');
39 if ($identity && ($article->user_id == $identity->id)) {
40
41     echo $this->html->link('Modifier',
42         ['controller' => 'articles', 'action' => 'edit', $article->id],
43         [
44             'class' => 'button',
45             'style' => 'background-color: Orange']);
46
47     echo $this->Form->postLink(
48         __('Supprimer'),
49         ['action' => 'delete', $article->id],
50         [
51             'confirm' => __("Vraiment supprimer {0} dont l'id vaut {1} ", $article->title),
52             'class' => 'button',
53             'style' => 'background-color: Black'];
54     }
55
56 ?>
```

Pas le droit :

```
index.php x default.php x ArticlesController.php x AppController.php x
Source History
77 public function edit($id = null) {
78     // $mesUsers = $this->fetchTable('Users')
79     //     ->find('list', keyField: 'id', valueField: 'username')
80     //     ->toArray();
81     try {
82         $leArticle = $this->Articles->get($id);
83         $identity = $this->request->getAttribute('identity');
84         if ($identity && ($leArticle->user_id != $identity->id)) {
85             $this->Flash->error(__("vous n'avez pas droit d'edité l'articles"));
86             return $this->redirect(['action' => 'index']);
87         }
88     } catch (\Exception $ex) {
89         if ($id == null) {
90             $this->Flash->error(__("L'action edit doit être appelé avec un identifiant"));
91         } else {
92             $this->Flash->error(__("L'article {0} n'existe pas", $id));
93         }
94         return $this->redirect(['action' => 'index']);
95     }
96 }
```

4 eme création des commentaire

Le commentaire créer par les utilisateur



```

43 |
44 |
45 | >id : <?= $comment->id ?>
46 | Créé le : <?= $comment->created->format(DATE_RFC850) ?>
47 | Modifié le : <?= $comment->created->format(DATE_RFC850) ?>
48 |
49 | <?php
50 |     $identity = $this->request->getAttribute('identity');
51 |     if ($identity && ($comment->user_id == $identity->id || $article->user_id == $identity->id)) {
52 |         echo $this->Form->postLink(
53 |             ('Supprimer'),
54 |             ['controller' => 'Comments', 'action' => 'delete', $comment->id],
55 |             [
56 |                 'confirm' => __("Vraiment supprimer le commentaire (0) dont l'id vaut {1} ", $comment->title),
57 |                 'class' => 'button',
58 |                 'style' => 'background-color: Black'];
59 |         }
60 |     }
  
```

5eme partie Création des tags

Creation fichier de migration avant de tout migré

```
C:\wamp64\www\blog_app\blog_app\bin>cake migrations migrate CreateTags
```

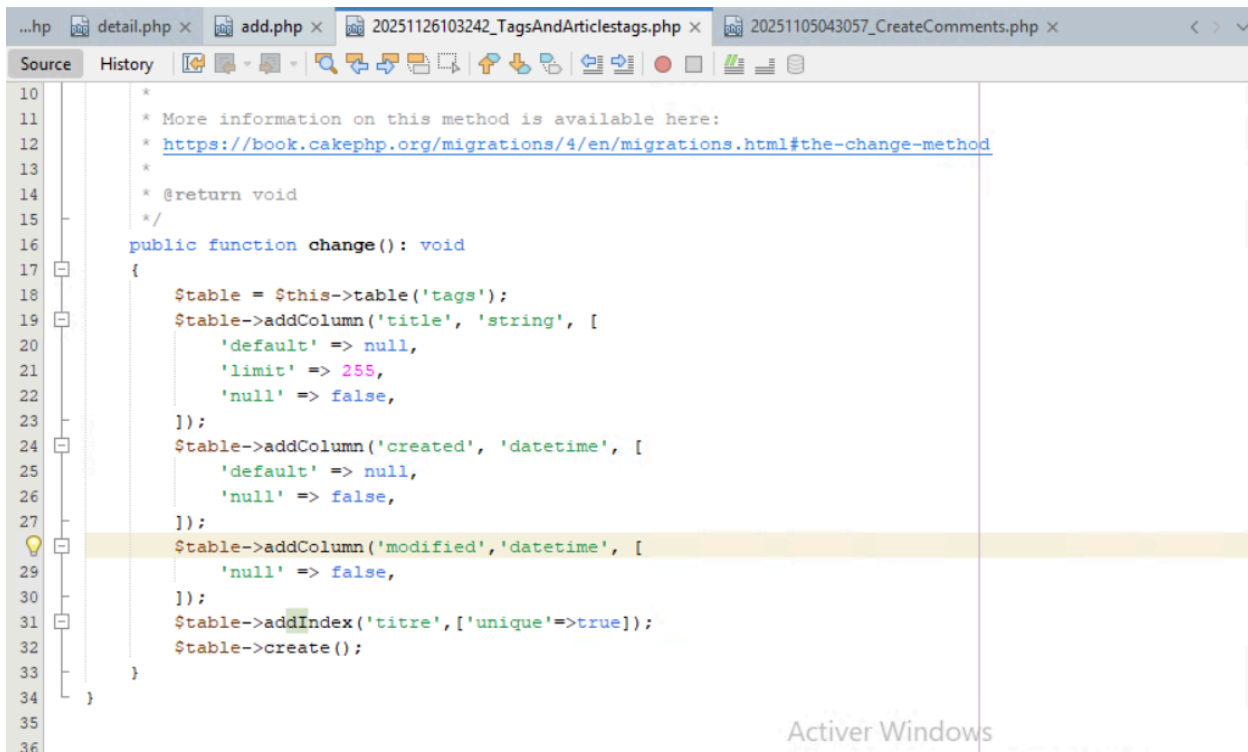
```
C:\wamp64\www\blog_app\blog_app\bin>cake migrations migrate
```

```
== 20251126103222 CreateTags: migrating
== 20251126103222 CreateTags: migrated 0.1678s

== 20251126113112 CreateArticlesTags: migrating
== 20251126113112 CreateArticlesTags: migrated 0.0811s
All Done. Took 0.3660s
```

```
C:\wamp64\www\blog_app\blog_app\bin>cake bake all tags
Bake All
```

```
C:\wamp64\www\blog_app\blog_app\bin>cake bake model articles_tags
One moment while associations are detected.
Bake cannot generate associations for composite primary keys at this time.
```



```
...hp detail.php x add.php x 20251126103242_TagsAndArticletags.php x 20251105043057_CreateComments.php x
Source History
10 *
11 * More information on this method is available here:
12 * https://book.cakephp.org/migrations/4/en/migrations.html#the-change-method
13 *
14 * @return void
15 */
16 public function change(): void
17 {
18     $table = $this->table('tags');
19     $table->addColumn('title', 'string', [
20         'default' => null,
21         'limit' => 255,
22         'null' => false,
23     ]);
24     $table->addColumn('created', 'datetime', [
25         'default' => null,
26         'null' => false,
27     ]);
28     $table->addColumn('modified', 'datetime', [
29         'null' => false,
30     ]);
31     $table->addIndex('titre', ['unique'=>true]);
32     $table->create();
33 }
34 }
35
36
```

Modification des code dans le tags

Add de tags

```
add.php x TagsController.php x edit.php x
Source History
5 * @var \Cake\Collection\CollectionInterface|string[] $articles
6 */
7 >>
8 <div class="row">
9 <aside class="column">
10 <div class="side-nav">
11 <h4 class="heading"><?__('Actions') ?></h4>
12 <?=$this->Html->link(__('List Tags'), ['action' => 'index'], ['class' => 'side-nav-item'])
13 </div>
14 </aside>
15 <div class="column column-80">
16 <div class="tags form content">
17 <?=$this->Form->create($tag) ?>
18 <fieldset>
19 <legend><?__('Add Tag') ?></legend>
20 <?php
21 echo $this->Form->control('title');
22 //echo $this->Form->control('articles_ids', ['options' => $articles]);
23 ?>
24 </fieldset>
25 <?=$this->Form->button(__('Submit')) ?>
26 <?=$this->Form->end() ?>
27 </div>
28 </div>
29 </div>
30
```

Controller tags

```
add.php x TagsController.php x edit.php x
Source History
64 * @return \Cake\Http\Response|null|void Redirects on successful edit, renders view otherwise.
65 * @throws \Cake\Datasource\Exception\RecordNotFoundException When record not found.
66 */
67 public function edit($id = null)
68 {
69     $tag = $this->Tags->get($id, contain: ['Articles']);
70     if ($this->request->is(['patch', 'post', 'put']) {
71         $tag = $this->Tags->patchEntity($tag, $this->request->getData());
72         if ($this->Tags->save($tag) {
73             $this->Flash->success(__('The tag has been saved.));
74
75             return $this->redirect(['action' => 'index']);
76         }
77         $this->Flash->error(__('The tag could not be saved. Please, try again.));
78     }
79     // $articles = $this->Tags->Articles->find('list', limit: 200)->all();
80     $this->set(compact('tag'));
81 }
82
```

Edit tags

```
add.php x TagsController.php x edit.php x
Source History
10 <div class="side-nav">
11 <h4 class="heading"><?= __('Actions') ?></h4>
12 <?= $this->Form->postLink(
13     __('Delete'),
14     ['action' => 'delete', $tag->id],
15     ['confirm' => __('Are you sure you want to delete # {0}?', $tag->id), 'class' => 'side-nav-item']
16 ) ?>
17 <?= $this->Html->link(__('List Tags'), ['action' => 'index'], ['class' => 'side-nav-item']) ?>
18 </div>
19 </aside>
20 <div class="column column-80">
21 <div class="tags form content">
22 <?= $this->Form->create($tag) ?>
23 <fieldset>
24 <legend><?= __('Edit Tag') ?></legend>
25 <?php
26     echo $this->Form->control('title');
27     //echo $this->Form->control('articles_ids', ['options' => $articles]);
28 <?>
29 </fieldset>
30 <?= $this->Form->button(__('Submit')) ?>
31 <?= $this->Form->end() ?>
32 </div>
33 </div>
34 </div>
35
```

Resultat :

The screenshot shows a web browser window with the URL localhost/blog_app/blog_app/tags. The page features a navigation bar with 'USERS' and 'ARTICLES' buttons, and links for 'Documentation', 'API', and 'Déconnexion de zeus'. A green message box at the top states 'The tag has been saved.' Below this is a table titled 'Tags' with a 'NEW TAG' button. The table contains the following data:

Id	Title	Created	Modified	Actions
1	Model	11/26/25, 11:48 AM	11/26/25, 11:48 AM	View Edit Delete
2	Rencontre	11/26/25, 11:49 AM	11/26/25, 11:49 AM	View Edit Delete
3	magic	11/28/25, 3:38 AM	11/28/25, 3:38 AM	View Edit Delete
4	brawl start	11/28/25, 3:38 AM	11/28/25, 3:38 AM	View Edit Delete

At the bottom of the page, there are navigation links '< previous' and 'next >' and a footer indicating 'Page 1 of 1, showing 4 record(s) out of 4 total'.

Code add tags dans le controller articles

```
...hp ArticlesController.php x detail.php x add.php x TagsController.php x index.php x index.php x
Source History
67
68     $this->set(compact('leArticle', 'leNewComment'));
69 }
70
71 public function add() {
72     $mesTags = $this->fetchTable('Tags')
73         ->find('list', keyField: 'id', valueField: 'title')
74         ->toArray();
75     $leNewArticle = $this->Articles->newEmptyEntity();
76
77     if ($this->request->is('post')) {
78
79         //on recupere le data du formulaire pour recreeer un tableau tag/priorite
80
81
82
83         //newData contient les données à enregistrer dans la base (articles et articles_tags
84         $newData['title'] = $data['title'];
85         $newData['content'] = $data['content'];
86
87         //Pour enregistrer les tags en meme temps, il faut la structure tags._joinData
88         $newData['tags'] = [];
89
90         // Nettoyer les données des tags
91         if (!empty($data['tags']['_ids'])) {
92             foreach ($data['tags']['_ids'] as $tagId) {
93                 //On crée la structure qui permetre d'enregistrer dans la table de jointure grace à
94                 foreach ($data['tags']['_joinData'] as $tag => $prio) {
95                     $stab = [];
96                     if ($tag == $tagId) {
97                         $stab = ['id' => $tagId, '_joinData' => $prio];
98                         $newData['tags'][] = $stab;
99                     }
100                 }
101             }
102         }
103     }
104 }
```

Pour le validateur limité les tags :

```
->add('rating', 'validValue', [
    'rule' => ['range', 1, 5],
]);
```

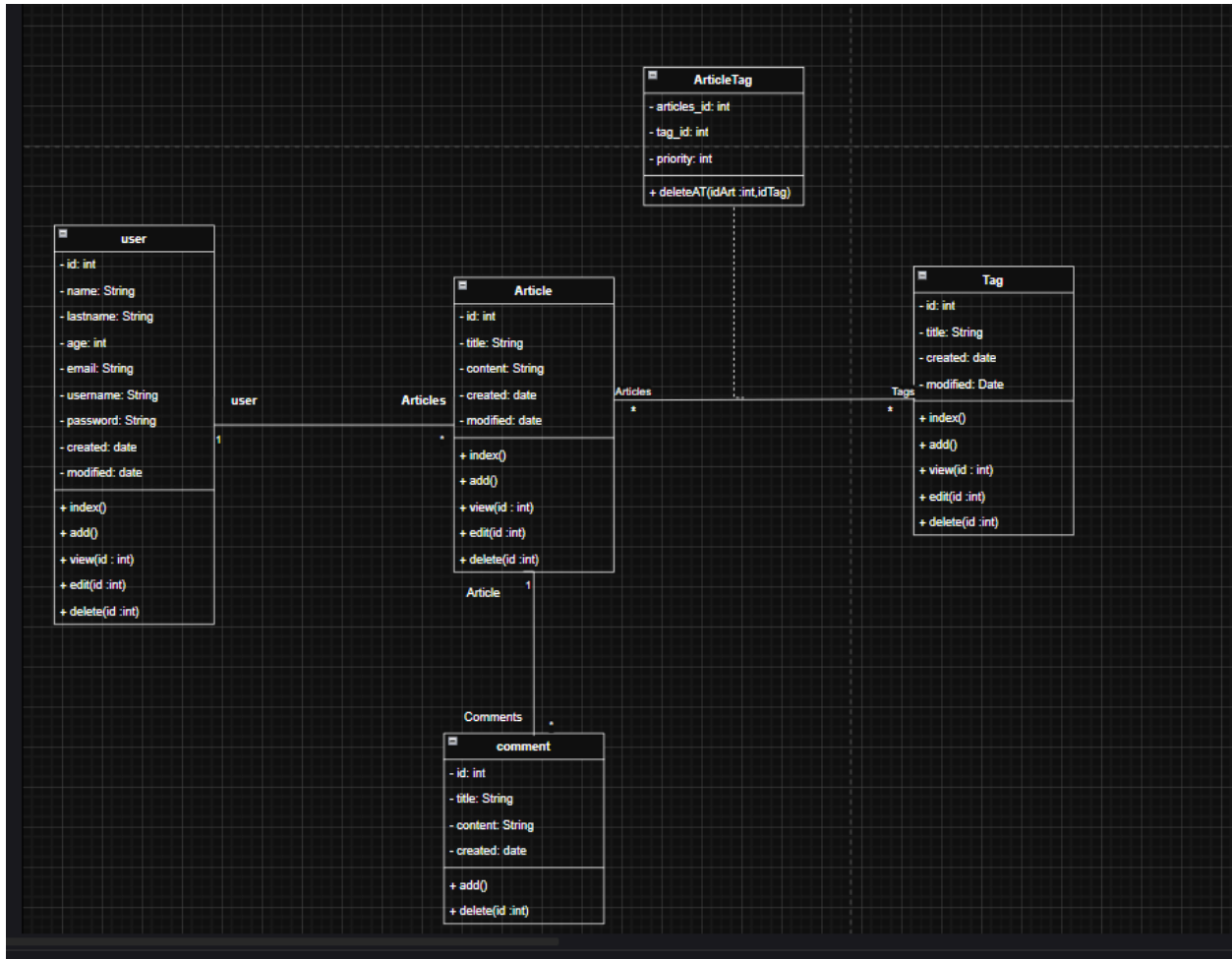
Afficher un tags associer, pour un tags qui n'est pas associer on affiche rien
:

```

<div class="related">
  <?php if(count($leArticle->tag)==0): ?>
  <h4><?= __('aucun Tags associés') ?></h4>
  <?php else: ?>
  <h4><?= __('Tags associés') ?></h4>
  <?php if (!empty($leArticle->tags)) : ?>
  <div class="table-responsive">
    <table>
      <tr>
        <th><?= __('Id') ?></th>
        <th><?= __('Title') ?></th>
        <th><?= __('Modified') ?></th>
        <th><?= __('Priority') ?></th>
        <th class="actions"><?= __('Actions') ?></th>
      </tr>
      <?php foreach ($leArticle->tags as $unTag) : ?>
      <tr>
        <td><?= h($unTag->id) ?></td>
        <td><?= h($unTag->title) ?></td>
        <td><?= h($unTag->modified->format(DATE_RFC850)) ?></td>
        <td><?= h($unTag->ArticlesTags['priority']) ?></td>
        <td class="actions">
          <?= $this->Html->link(__('View'), ['controller' => 'Tags', 'action' => 'view', $unTag

```

Diagramme de classe final du TP blog_app:



Recap de Base :

1. Structure du projet

- **src/Controller** : contient les contrôleurs (logique appliqueer).
- **src/Model** : gère les entités est tables (accès aux données).
- **templates/** : regroupe les vues (HTML + PHP).
- **config/routes.php** : définit les routes de l'application.

2. MVC en action

- **Model** : représente la base de données via des classes `Table` et `Entity`.
- **View** : affiche les données (fichiers `.php` dans `templates`).
- **Controller** : reçoit les requêtes, appelle le modèle et renvoie la vue.