



# W2- Projets MVC

---

W-WEB-120

## MVC Blog

---

Création d'un blog en utilisant le framework CakePHP

v1.0



# MVC Blog

Création d'un blog en utilisant le framework CakePHP

---

Nom du répertoire: **Piscine\_MVC\_MVC\_Blog**  
Droits de ramassage: ramassage-tek  
langage: PHP  
Taille du groupe: 1

---



# Jour 01

## Objectifs de la journée

- Prise en main de CakePHP
- Inscription et connexion fonctionnels

## Etapes

1. Installer "CakePHP": <http://book.cakephp.org/2.0/en/installation.html>
2. Installer "CakePHP Debug Kit": [https://github.com/cakephp/debug\\_kit](https://github.com/cakephp/debug_kit)
3. Créer une page d'inscription avec un formulaire comportant les champs :
  - (a) Nom d'utilisateur: username
  - (b) Mot de passe: password
  - (c) Nom: name
  - (d) Prénom: lastname
  - (e) Date de naissance: birthdate
  - (f) Adresse mail: email
4. Créer une page d'accueil avec:
  - (a) Lorsque l'on n'est pas connecté:
    - i. Un formulaire de connexion comportant les champs:
      - A. Nom d'utilisateur: username
      - B. Mot de passe: password
  - (b) Lorsque l'on est connecté:
    - i. Afficher le nom d'utilisateur ainsi qu'un bouton pour se déconnecter.
5. Router l'URL "/inscription" vers votre formulaire d'inscription.
6. Faire fonctionner les formulaires correctement.
7. Faire des tests, par exemple:
  - (a) Inscrire un nouvel utilisateur et se connecter.
  - (b) Supprimer cet utilisateur de la base de données.
  - (c) Vérifier que l'utilisateur n'est plus connecté.



Vous pouvez vous aider des tutoriaux:

<http://book.cakephp.org/2.0/en/tutorials-and-examples.html>



# Jour 02

## Objectifs de la journée

- Page d'accueil listant l'ensemble des billets.
- Système de "CRUD" fonctionnel pour les billets.

## Etapes

1. Créer une table dans la base de données, qui doit comporter les champs suivants:
  - (a) Identifiant: id
  - (b) Date de création: created
  - (c) Date de modification: updated
  - (d) Rédacteur: user\_id
  - (e) Titre: title
  - (f) Contenu: content
  - (g) Liste de tags: tags
2. Créer une page permettant de publier un nouveau billet, accessible depuis l'URL `"/billet/new"`, comportant un formulaire afin de remplir correctement la table précédemment décrite.
3. Implémenter un système de "CRUD", à travers lequel chaque utilisateur doit pouvoir:
  - (a) Lire tous les billets
  - (b) Publier de nouveaux billets
  - (c) Editer ses propres billets
  - (d) Supprimer ses propres billets
4. Créer les routes `"/billet/:id/edit"` et `"/billet/:id/delete"` de manière sécurisée.
  - (a) Vérifier que la page appelante est bien une page de votre site:
    - i. Utiliser la variable `$_SERVER['HTTP_REFERER']`
  - (b) Vérifier que la personne possède les droits d'effectuer l'action.
5. Implémenter un système de pagination pour la lecture de tous les billets.
6. Faire des tests, par exemple:
  - (a) Essayer d'éditer un billet qui n'existe pas.
  - (b) Essayer de poster du code HTML, PHP.
  - (c) Essayer les injections de type XSS, SQL.
  - (d) Essayer "F5" lors de la publication d'un billet.



# Jour 03

## Objectifs de la journée

- Système de commentaires fonctionnel pour les billets.
- Partie d'administration sur le site avec un CRUD complet.

## Etapes

1. Créer la route `"/billet/:id"` qui doit afficher le billet suivi par:
  - (a) Tous les commentaires
  - (b) Un formulaire pour laisser un nouveau commentaire
2. Afficher sur la page d'accueil pour chaque billet, le nombre de commentaires laissés ainsi qu'un lien "Commenter" permettant d'accéder à l'URL précédemment décrite.
3. Créer une partie réservée à l'administration accessible depuis l'URL:
  - (a) `"/admin"` qui affiche un récapitulatif avec:
    - i. Les 10 derniers utilisateurs inscrits.
    - ii. Les 10 derniers billets publiés.
    - iii. Les 10 derniers commentaires laissés.
  - (b) `"/admin/users"` pour tous les utilisateurs (avec système de pagination).
  - (c) `"/admin/billets"` pour tous les billets (avec système de pagination).
  - (d) `"/admin/comments"` pour tous les commentaires (avec système de pagination).
4. Implémenter un système de type de compte (droits relatifs aux utilisateurs):
  - (a) Commentateur: L'utilisateur peut seulement commenter les billets.
  - (b) Blogueur: L'utilisateur peut poster des billets.
  - (c) Administrateur: L'utilisateur a accès à l'URL `"/admin"` et possède tous les droits.
5. Permettre à un administrateur de bloquer (ou débloquer) un compte.
6. Faire des tests.



# Jour 04

## Objectifs de la journée

- Système de billet avec contenu élaboré (vidéos, musiques, images...).
- Système de tags et de recherche fonctionnel pour les billets

## Etapes

1. Implémenter un système de mise forme des billets, au choix:
  - (a) Soit grâce aux "balises" BBCode (liste minimale):
    - i. [b] Gras [/b]
    - ii. [i] Italique [/i]
    - iii. [s] Souligné [/s]
    - iv. [color=red] Coloré (rouge) [/color]
    - v. Pour les liens:
      - A. [url] Lien [/url]
      - B. [url=Lien] Texte [/url]
      - C. [img] Image [/img]
  - (b) Soit grâce à une interface WYSIWYG.
2. Implémenter en haut de votre blog, un système de recherche, qui doit pointer vers l'URL "/search/:keywords" comportant les résultats de la recherche:
  - (a) Dont les tags des billets comportent approximativement le mot clé recherché.
    - i. Utiliser la fonction PHP "levenshtein".
  - (b) Dont les billets comportent exactement le mot clé recherché (hors code HTML).
3. Afficher sur la page du résultat de la recherche, pour tous les billets trouvés:
  - (a) Le titre, doit être un lien de la forme "/billet/:id/search/:keywords", qui permet par la suite de mettre en évidence (highlight) la recherche sur le billet concerné.
  - (b) Le début (200 premiers caractères) du billet concerné.
4. Tester encore. Il doit être possible de revenir en arrière sur les pages de recherche sans avoir à revalider les envois de formulaire.



# Jour 05

---

## Objectifs de la journée

- Page de contact
- Tests et debug

---

## Etapes

1. Implémenter un formulaire de contact accessible depuis l'URL `"/contact"`.
  - (a) Il faut être connecté pour arriver sur cette page.
  - (b) Le message doit être envoyé par mail à tous les administrateurs.
  - (c) On ne peut pas envoyer plus de 2 messages par heure.
2. Tester tout votre site et débbugger.
3. Finaliser le design de votre blog.



# Bonus

## Bonus

---

1. Faire en sorte qu'il soit possible pour chaque utilisateur de bénéficier d'un sous-domaine. Il doit être possible de changer de sous-domaine au maximum 1 fois par semaine.
2. Faire en sorte que chaque utilisateur puisse personnaliser le thème de son blog. Cela peut comporter pour le style de la page: couleurs, police d'écriture, images, disposition etc.
3. Faire en sorte d'implémenter un système de messagerie entre les membres. Avec le système de mise en forme que vous n'avez pas utilisé pour les billets.
4. Implémenter une interface de statistiques pour connaître:
  - (a) Le blog le plus consulté
  - (b) Le billet le plus consulté
  - (c) Le blogueur le plus actif
  - (d) Et trouver au moins 3 autres chiffres significatifs à présenter.
5. S'il vous reste encore du temps, implémentez d'autres fonctionnalités qui pourraient être utiles !