

Le templating

Par templating, on entend la rédaction d'un code qui servira de modèle pour plusieurs pages. Il existe des moteurs de templates extrêmement bien réalisés pour PHP (et qui sont inclus dans les gros *frameworks* tels que **Laravel** ou **Symfony**), nous n'irons pas du tout jusque-là. Dans cet exercice on va apprendre à modulariser et réutiliser le code à travers plusieurs pages.

Travail à faire :

Au cours des 5 exercices précédents, vous avez développé toute la partie fonctionnelle de votre application. A aucun moment nous n'avons cherché à rendre les choses jolies ni même, de manière plus générale à obtenir une véritable interface web.

- Pour commencer, reprenez la page **login.php** que vous avez réalisée et ajoutez-lui un header et un *footer*. Faites en sorte que ces header et *footer* soient jolis, comme vous les ferez pour un vrai site. Vous pouvez inventer un nom pour le site, et ajouter un logo de votre choix. Il faut que ces *header/footer* soient réalistes ! Bien sûr, il vous faudra aussi écrire le CSS nécessaire. N'hésitez pas à utiliser **Bootstrap/Bulma** si vous le souhaitez.
- Maintenant nous allons faire en sorte que ces *header/footer* s'appliquent automatiquement dans toutes vos pages ! (home, signin, resetpassword, ...). Créez un fichier **header.php** et copiez dans celui-ci tout le code propre au header que vous avez écrit. Faites de même pour le *footer* dans un fichier **footer.php**
- Dans le fichier **login.php**, supprimez le code du *header* et « incluez » le fichier **header.php** à l'exact endroit où était situé le code de votre header.
- Faites exactement la même chose pour le footer.
- Généralisez ça à toutes vos pages
- Faites la même chose avec le contenu de la balise head avec les appels aux fichiers JS et CSS

Conclusion :

A l'avenir nous fractionnerons le code des pages que nous allons écrire en isolant dans des fichiers à part tous les « bouts de code » qu'on sera amenés à réutiliser sur diverses pages.

Les *frameworks* MVC (Modèle Vue Contrôleur) fonctionnent aussi sur le même principe. Lorsque vous fabriquerez vos vues, vous pourrez injecter ces fragments de page là où vous le souhaitez. Les vues seront constituées d'un ensemble de fragments de page que vous injecterez dans un certain ordre, avec potentiellement des données spécifiques à la page, que vous aurez préparées avant de générer la vue. Nous verrons tout ça dans quelques semaines lorsqu'on commencera à travailler avec le *framework* MVC **CodeIgniter**.