

TP 2 : La couche Vue d'un projet d'application Web

L'objectif de ce TP est de développer les différents composants de la couche Vue du projet ourface dans le cadre de l'architecture MVC présentée dans le TP1. Dans ce TP, vous devez comprendre et implémenter ces composants.

Description :

- L'architecture MVC présentée lors du TP1 fait intervenir plusieurs composants de la couche Vue : un *layout* unique et plusieurs *templates* dépendant directement de l'action exécutée et de son état après exécution (*SUCCESS*, *ERROR*, *NONE*). Comme vu dans le TP1, le dispatcher définit le nom du *template* et charge le *layout*, ce dernier charge ensuite le *template* attendu (stocké dans une variable temporaire). Ce schéma pourra être modifié dans ce TP de façon à faire en sorte qu'un *layout* puisse charger deux *templates* différents, notamment pour une fonctionnalité toujours présente sur la page quelle que soit l'action demandée (visualisation du chat par ex.).

Pour ce TP, seul le couple HTML/CSS est demandé. Si du Javascript est nécessaire pour le traitement des données, nous l'implémenterons dans les séances futures. Vous pouvez néanmoins faire appel à des framework javascript pour le design de vos pages. La mise en place du *layout et des templates* doit être réfléchie dans un contexte de site dynamique. En d'autres termes, les informations contenues dans le *layout* et les *templates* associés sont chargées dans leur globalité une première fois, leur mise à jour se fera ensuite de manière partielle par des requêtes Ajax.

- Il est attendu que du PHP soit utilisé à l'intérieur du *layout* ou des *templates* pour afficher des informations provenant du modèle **via le contrôleur** (pas d'accès direct au modèle par appel aux classes outils). Nous utiliserons alors une syntaxe PHP adaptée à l'écriture de templates. Un exemple utilisant une boucle *foreach* pour lister un ensemble de données est présenté ci-dessous.

```
<ul> <?php foreach( $context->tableData as $data ): ?>
<li><?php echo $data ?></li> <?php end foreach; ?>
</ul>
```

Il est rappelé que l'utilisation du PHP au niveau de la couche Vue doit se limiter **au strict minimum** touchant à la récupération d'informations à afficher.

Implémentation :

1. La fenêtre d'identification étant déjà implémentée, il faut à présent pouvoir afficher **votre profil personnel** après identification. La figure 1 fournit un aperçu possible d'une page de profil.

Pour cela, définissez le *layout* et le (ou les) *template* permettant de contenir :

- une photo de profil (attribut avatar ; afficher une image par défaut si pas d'avatar)
- les informations de l'utilisateur (nom, prénom, date de naissance, etc.)
- le statut de l'utilisateur (celui-ci pourra être modifié à tout moment à l'aide d'un formulaire)
- un formulaire d'envoi de messages (dont vous êtes le destinataire)

2. Toujours sur votre page de profil personnel, une zone de menus doit être définie, comportant un ensemble de liens. Les liens vont permettre d'accéder, entre autres, à :

- la liste des utilisateurs du réseau social. Chaque utilisateur de la liste est lui-même un lien,

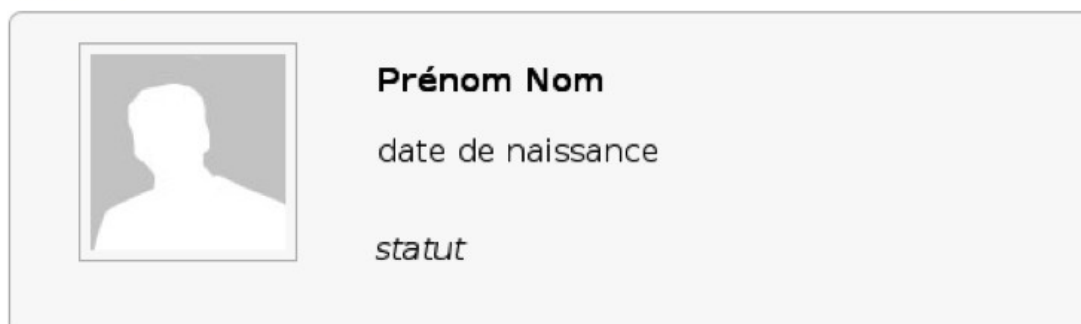


Illustration 1: Exemple de template pour l'affichage du profil

permettant d'accéder au profil de cet utilisateur (cf. 3).

- votre profil personnel

3. Lorsque l'on clique sur un utilisateur du réseau social, il est alors possible d'afficher son profil. Ce dernier n'est pas identique à notre profil personnel, car il n'est pas possible de modifier le statut d'un autre utilisateur. Un formulaire d'envoi de message est à mettre à disposition. Le destinataire des messages issus de ce formulaire sera l'utilisateur courant (dont le profil est actuellement consulté).

4. Définissez les éléments de la couche Vue nécessaires à l'affichage d'un message (expéditeur, destinataire éventuel, post, ...). Ce *template* doit contenir toutes les informations propres à un message. Il doit afficher le nombre de personnes ayant voté pour le message (champ *aime*) et des informations relatives à un éventuel partage de celui-ci. Par ailleurs, les dispositifs de vote et de partage devront être mis à disposition.

5. Ajoutez la liste des messages sur les *templates* de profils (le votre et celui des autres utilisateurs) en utilisant l'élément de *template* partiel implémenté précédemment (question 3). Lorsque vous êtes sur votre profil, seuls les messages dont vous êtes le destinataire sont à afficher. De la même manière, sur le profil d'un autre utilisateur, les messages qui lui sont destinés sont à afficher. Cependant, après identification, la page d'accueil peut afficher N messages aléatoires.

6. Toutes les pages doivent permettre, dans un espace séparé, d'afficher la liste de tous les (derniers) messages du *chat*. Définissez un *template* spécifique pour ces messages particuliers. Vu le nombre important de messages qui seront postés, nous limiterons le nombre de messages à afficher à 10¹. Il est aussi possible, depuis cet espace, d'envoyer un message de type *chat* (n'ayant pas de destinataire). Un formulaire d'envoi de message doit donc être mis en place.

Répartition du travail

Au sein de chaque binôme, le travail doit se répartir de la manière suivante :

- réflexion collective sur la charte graphique à définir pour le site
- Mise en place commune du menu et de la liste d'amis
- deux parties distinctes (1 par étudiant) :
 - gestion du mur (personnel ou amis)
 - gestion du chat et du profil personnel
- Tout comme pour le premier TP, des commentaires bien fournis devront expliciter l'auteur de chaque morceau de codes (sous peine d'une forte pénalité sur la note finale).

¹En bonus, vous pourrez implémenter un système de pagination permettant d'accéder aux messages plus anciens.