

TP 3

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 *vues* chacune en lien direct avec l'action exécutée et son état après exécution (*SUCCESS*, *ERROR*, *NONE*). Comme vu dans le TP1, le dispatcher définit le nom de la *vue* et charge le *layout* ; ce dernier charge ensuite la *vue* attendue (stockée dans une variable temporaire). Ce schéma devra être modifié dans ce TP de façon à faire en sorte qu'un *layout* puisse charger plusieurs *vues* différentes, notamment pour permettre d'afficher des ensembles de contenus différents sur une seule et unique page.

Pour ce TP, seul le couple HTML/CSS est demandé, incluant l'utilisation du framework Bootstrap pour la mise en forme. Si du Javascript est nécessaire pour le traitement des données, nous l'implémenterons dans les séances futures.

- La mise en place du *layout* et des *vues* doit être réfléchie dans un contexte de site dynamique. En d'autres termes, les informations contenues dans le *layout* et les *vues* associées sont chargées dans leur globalité une première fois (dispatcher global), leur mise à jour se fera ensuite de manière partielle par des requêtes Ajax (dispatcher Ajax).
- Il est attendu que du PHP soit utilisé à l'intérieur du *layout* ou des *vues* 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 vues. 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>
```

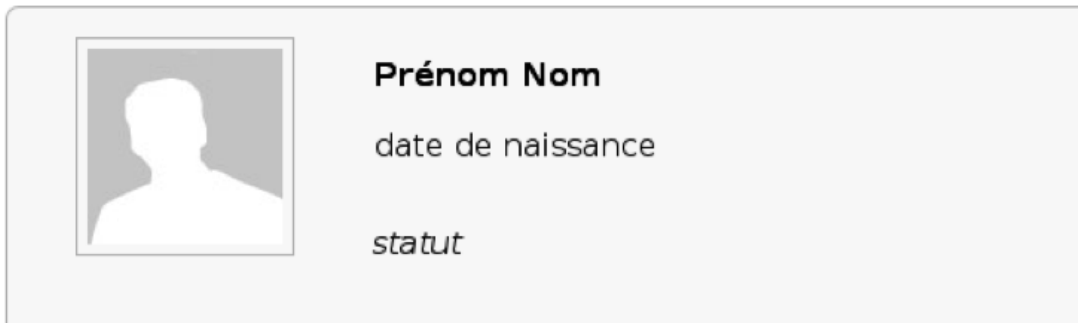
Il est rappelé que l'utilisation du PHP au niveau de la couche Vue doit se limiter **au strict minimum** touchant à la récupération d'informations à afficher.

Implémentation :

1. La fenêtre d'identification étant déjà implémentée (possibilité d'associer cette page à un *layout* différent), il faut à présent pouvoir afficher **le layout de l'application** après identification ie la page *ourface* de l'utilisateur connecté. Ce *layout* devra contenir (au minimum) différentes vues permettant :

- d'afficher le profil de l'utilisateur connecté ;
- d'afficher un bouton de déconnexion ;
- d'afficher la liste des amis ;
- d'afficher le mur ;
- d'afficher la fenêtre de chat ;
- d'afficher un bandeau de notification.

Note : Comme précisé précédemment, cela signifie que le layout devra charger non plus une seule vue stockée dans la variable \$template_view mais un ensemble de vues. De même, cela signifiera que les informations dynamiques présentes dans chacune des vues devra faire l'objet de



*Illustration 1: Exemple de vue pour l'affichage du profil
l'exécution d'une action du contrôleur (à prévoir dans les séances suivantes) permettant la récupération des données correspondantes.*

2. La figure 1 fournit un aperçu possible d'une vue de profil. Celle-ci devra 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 pour l'utilisateur connecté uniquement. La consultation d'un profil d'un autre utilisateur que celui connecté ne permettra pas cette modification.
 - un formulaire d'envoi de messages dont le propriétaire du profil sera le destinataire.
3. Lorsque l'on clique sur un utilisateur du réseau social (présent dans la liste d'amis), il est alors possible d'afficher sa page *ourface* contenant son profil, son mur, la liste d'amis (commune à tous), ... Le formulaire d'envoi de messages sera toujours à disposition sur la vue profil. Le destinataire des messages issus de ce formulaire sera alors l'utilisateur courant ie celui dont le profil est actuellement en consultation.
4. La vue du mur permettra de visualiser les messages du profil consulté (ou un sous-ensemble restreint suivant le nombre de messages dans la base). La visualisation d'un message comprendra toutes les informations qui s'y rapportent : émetteur, destinataire, date, texte... Devront être également affichés 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.

Note : Un message pourra être considéré comme une sous-vue, la vue du mur contenant alors un ensemble de sous-vues de type message à l'intérieur de la vue mur.
5. Finalement, le *layout* devra inclure une fenêtre de chat fournissant les fonctionnalités d'une messagerie instantanée, notamment l'affichage des (derniers) messages du chat. Afin d'accroître l'ergonomie de cette messagerie, la fenêtre de chat devra répondre aux exigences suivantes : (1) la fenêtre sera une zone déplaçable (type drag&drop) et redimensionnable, (2) elle sera munie d'un onglet (ou autre forme) permettant de la réduire ou de l'agrandir, (3) même réduite, la mise à jour des messages devra être effective dans la fenêtre et un élément visuel (clignotement de l'onglet, changement de couleur, ...) permettra de faire savoir à l'internaute que de nouveaux messages ont été postés. Vu le nombre important de messages qui pourront être postés, nous limiterons le nombre de messages à afficher à 10. Il est aussi possible, depuis cet espace, d'envoyer un message de type *chat* (n'ayant pas de destinataire contrairement à un message

posté sur un mur). Un formulaire d'envoi de messages devra donc être mis en place. Il s'agit ici de prévoir les différents éléments visuels, le codage des différentes fonctionnalités sera prévu dans les séances à venir.