

## TP 4 : Utilisation d'Ajax dans un projet d'application Web

L'objectif de ce TP est d'utiliser le javascript au travers du framework jQuery et les requêtes Ajax afin de recharger certaines parties de vos pages en fonction des actions demandées par l'utilisateur.

### Description :

- L'appel à une grande majorité des fonctionnalités de votre application (ex : le post d'un message sur un mur d'un utilisateur) conduisant à une action sur le modèle de données et/ou à l'affichage/mise à jour d'une sous-vue particulière dans le layout devra se faire par le biais d'une requête Ajax.
- Le code javascript exécuté côté client se fera par le biais de la librairie jQuery (jquery.com).
- La finalisation d'une action côté serveur pourra faire l'objet d'un retour au client sous la forme d'un message de notification (ex : « message posté avec succès » ou « erreur lors de l'envoi du message » lors d'un post de message sur le chat). Ce message pourra s'afficher dans un bandeau de notification commun à toutes les actions.

### Implémentation :

1. les requêtes Ajax que vous devrez écrire seront toutes associées à une action appelée au travers d'un dispatcher (présent côté serveur). Ce dispatcher sera légèrement différent de celui que nous vous avons fourni dans le squelette de l'application. En effet, celui que vous utilisez actuellement permet de renvoyer un layout complet, incluant une ou plusieurs vues. Le dispatcher Ajax devra renvoyer, non plus le layout, mais une simple vue (ou sous-vue) qui viendra mettre à jour partiellement le layout initialement chargé, si nécessaire. Votre premier travail est donc d'écrire ce deuxième dispatcher. Attention, le premier dispatcher sera toujours utilisé pour l'affichage complet du (ou des) layout lors de la première requête à votre application.
2. Toutes les fonctionnalités liées à la gestion du mur (écriture d'un message, vote, partage, ...) devront être associées à du code jQuery qui fera intervenir si nécessaire une (ou plusieurs) requête Ajax. Par exemple, l'écriture d'un message sur le mur, induisant une action sur la base de données (au travers du modèle de données) ainsi qu'une mise à jour des données affichées sur le mur fera l'objet d'une requête Ajax. Par ailleurs, un élément visuel devra permettre d'identifier la réception du ou des derniers messages, dès lors que ceux-ci ont été postés dans un laps de temps fixé (la dernière heure). Cette fonctionnalité fera l'objet d'une requête automatique toutes les n minutes permettant de désactiver cet élément visuel dès lors que ce laps de temps sera dépassé.
3. Toutes les fonctionnalités liées à la gestion du chat devront être associées à du code jQuery qui fera intervenir si nécessaire une (ou plusieurs) requête Ajax. Ces fonctionnalités comprennent :
  - le rafraîchissement automatique des messages du chat (rafraîchissement toutes les n secondes<sup>1</sup>). Dans ce cadre, il faudra conserver côté client l'identifiant et/ou la date et l'heure du dernier message de type chat afin de récupérer uniquement, côté serveur, les messages ultérieurs ;
  - l'envoi d'un message de type chat au travers de la fenêtre. Cet envoi passera par la validation d'un formulaire, côté client, et l'enregistrement dans la base côté serveur.

---

<sup>1</sup> Soyez raisonnable dans le réglage de ce temps en secondes afin de ne pas surcharger le serveur de requêtes. Vous n'êtes pas tout seul !

- Afin d'accroître l'ergonomie de cette messagerie instantanée, 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.
4. La gestion du profil utilisateur et notamment la mise à jour de son statut et avatar devra être associée à du code jQuery qui fera intervenir une requête Ajax. Cette requête devra se charger de la mise à jour des données dans la base de données et de la mise à jour de la vue dans le layout.

### **Répartition du travail**

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

- Réflexion collective sur le nouveau dispatcher dédié aux requêtes Ajax et développement commun
- deux parties distinctes (1 par étudiant) :
  - gestion du mur (personnel ou amis) et du profil personnel
  - gestion du chat
- Tout comme pour le premier TP, des commentaires bien fournis devront expliciter l'auteur de chaque morceau de codes et accompagner l'ensemble des sources fournies.