



Applications web et mobiles

TP n°1 : Backend en PHP et BD

Objectifs et préparations des TP

L'objectif des TP est de développer un forum permettant aux étudiants et aux enseignants d'échanger sur des sujets relatifs aux cours proposés dans le département informatique. Les TP sont organisés de la manière suivante :

- dans les TP n°1 à 9, vous réaliserez le *backend* de votre application en PHP et le *frontend* en *Angular*.
- dans les TP n°10 et 11, vous réaliserez un nouveau *backend* en Node/Express et vous modifierez votre *frontend Angular* afin qu'il converse avec ce nouveau *backend*.
- dans les TP n°12 à 14, vous adapterez votre *frontend* afin d'en faire une application pour smartphone. Pour cela, vous modifierez votre code *Angular* en un code *Ionic*.

Étape 1 – Serveurs Apache et MySQL

Si vous travaillez en salles de TP :

Votre serveur MySQL est situé sur la machine `mysql.pedaweb.univ-amu.fr`, comme vous l'a indiqué l'email de la DOSI que vous avez reçu après avoir fait votre demande d'espace web pédagogique, (cf. <https://dud.univ-amu.fr/votre-espace-web-pedagogique-amu>). C'est donc ce « host » qu'il faudra utiliser avec PDO.

Sous Linux, les machines de toutes les salles de TP du bâtiment A de Polytech administrées par la DOSI ont leur propre serveur web, qu'il vous faudra utiliser. Pour cela, dans votre *home directory*, si vous ne possédez pas encore de répertoire `public_html`, créez-en un. Encore mieux : créez un répertoire `public_html/forum-angular-php`. *In fine*, celui-ci contiendra un sous-répertoire `backend` qui contiendra votre *backend* en PHP et un sous-répertoire `frontend` qui contiendra votre *frontend* en *Angular*. Pour l'instant, créez uniquement le sous-répertoire `backend`.

En salle de TP, à chaque séance, vous devrez démarrer votre serveur via la commande `myapache2 start`. Le contenu de votre répertoire `public_html` sera alors accessible de votre browser via l'URL : `http://127.0.0.1:8080/`. Ainsi, pour visualiser le résultat de l'exécution d'un fichier `toto.php` de votre `public_html`, il vous faudra accéder à l'URL `http://127.0.0.1:8080/toto.php`. À la fin de chaque séance, arrêtez proprement votre serveur web via la commande `myapache2 stop`.

Si vous travaillez sur votre propre ordinateur :

Vous avez dû installer AMP sur votre ordinateur et donc vous avez vos propres serveurs web (`apache2`) et `Mysql` sur votre machine. A priori, le « host » qu'il faudra utiliser avec PDO est alors `127.0.0.1`.

À la racine de votre serveur web (ce qui correspond à l'URL `http://127.0.0.1`), créez un répertoire `forum-angular-php`. *In fine*, celui-ci contiendra un sous-répertoire `backend` qui contiendra votre *backend* en PHP et un sous-répertoire `frontend` qui contiendra votre *frontend* en *Angular*. Pour l'instant, créez uniquement le sous-répertoire `backend`.

Étape 2 – Installations de logiciels

Afin de tester aisément votre *backend*, installez l'application `postman`. Elle est téléchargeable sur <https://www.postman.com/downloads>. **Attention** : il existe également un plugin chrome, mais c'est bien l'application qu'il faut utiliser et non le plugin car ce dernier ne vous permettra pas d'exploiter les *cookies* qui serviront à vos authentifications.

Vous pouvez tester Postman en réalisant un POST sur `https://christophe-gonzales.pedaweb.univ-amu.fr/forum/backend/checkLogin.php`, qui est le *backend* du forum de démonstration, cf. la figure 1. Si vous transmettez dans le *body* de la requête un `form-data` avec des champs `login` et `password` ayant les valeurs correspondantes au login/password des utilisateurs du site (cf. AMeTICE), vous obtiendrez un message vous indiquant que votre authentification a été réussie.

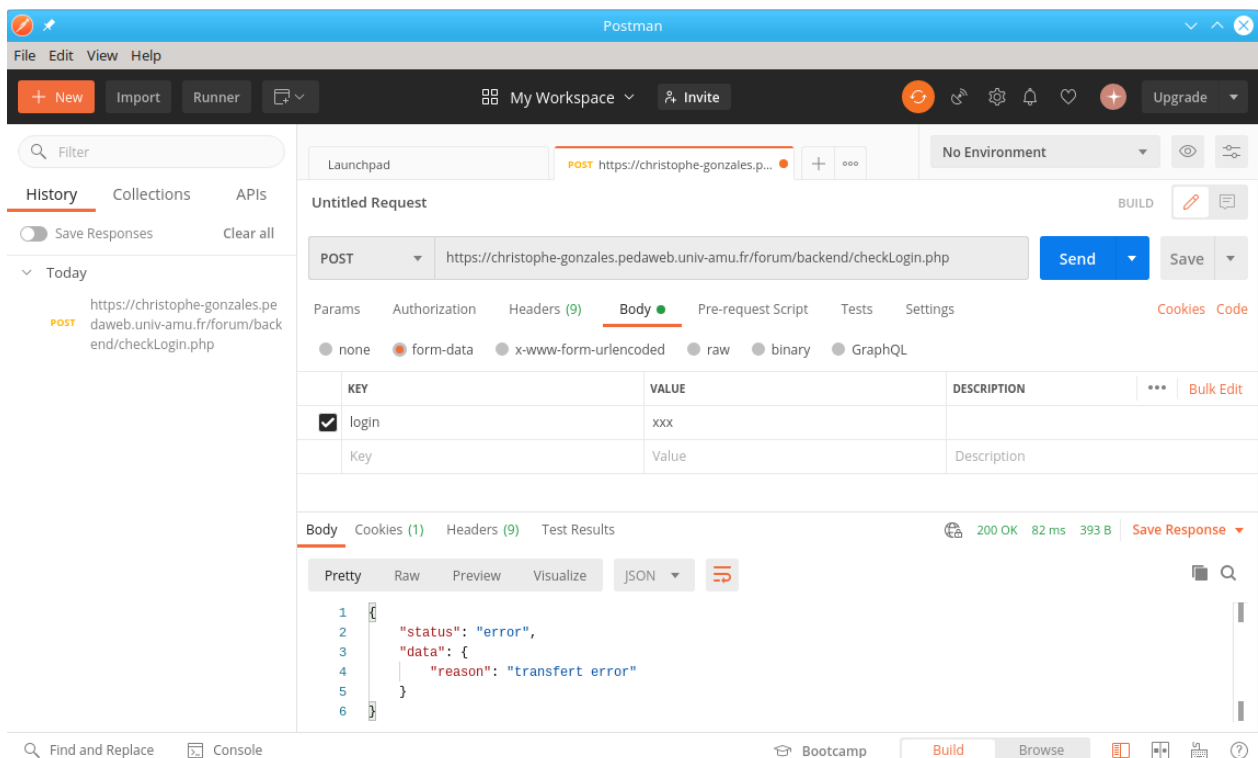


FIGURE 1 – Utilisation de postman.

Vous pourrez utiliser l'éditeur `phpStorm` afin d'écrire votre code PHP et l'éditeur `webStorm` afin d'écrire votre code *Angular*. Ces éditeurs sont déjà installés sur les machines de TP.

Développement du backend PHP

Étape 3 – Schéma de base de données

La première étape de développement de votre forum consiste à définir les tables de votre base de données. Pour cela, observez bien les pages du forum <https://christophe-gonzales.pedaweb.univ-amu.fr/forum>. Vous en déduirez les différentes tables dont vous avez besoin ainsi que leurs champs. Les *login* et *password* pour se connecter au forum sont indiqués sur AMeTICE.

Lorsque vous avez défini le schéma de votre base, appelez votre chargé de TP afin de le valider.

Étape 4 – Création de la base de données

Créez votre base sur votre espace web pédagogique de l'université si vous travaillez en salle de TP (cf. le mail que vous avez reçu lors de la création de cet espace ou l'URL <https://dud.univ-amu.fr/votre-espace-web-pedagogique-amu>) ou bien sur votre serveur Mysql si vous travaillez sur votre propre ordinateur. Créez vos tables et remplissez les avec deux utilisateurs, deux cours, *etc.* Cela vous permettra de tester votre *backend*.

Étape 5 – Organisation du code du backend

Comme dans toute application qui se respecte, il faut organiser le code de votre *backend* de manière cohérente et efficace. Maintenant que vous connaissez votre base, vous avez une bonne idée des scripts PHP qui seront appelés par votre *frontend* afin de récupérer les données dont il a besoin. Déterminez les noms des scripts PHP et des répertoires que vous comptez créer pour votre *backend*.

Lorsque vous pensez avoir terminé, appelez votre chargé de TP afin de valider cette organisation.

Étape 6 – Création du script de connexion à la base de données

Écrivez le script PHP de création de l'instance PDO, comme vu en cours, ainsi que le script de configuration contenant les paramètres de connexion (le nom du host, de la base, *etc.*). Personnellement, j'appelle traditionnellement ces scripts `mysqlConnect.php` et `config.php`.

Étape 7 – Test de la base de données

Écrivez un script PHP se connectant à votre base de données et affichant la liste de tous les utilisateurs. Ici, vous pourrez utiliser la fonction `fetchAll` de PDO. Plus généralement, pour la création de tout votre *backend*, vous aurez besoin d'utiliser les fonctions `fetchAll`, `fetchColumn` et `fetch` de PDO, qui sont très bien documentées dans l'URL suivante : <https://phpdelusions.net/pdo>. Vous pourrez aussi vous reporter au cours n°1 pour voir la différence entre ces fonctions.

Testez votre script via l'application `postman`.

Modifiez votre script afin qu'il attende un champ « login » et qu'il affiche également ce champ, et retestez-le avec `postman`.