

# TD 02 : Échanges client-serveur de base

## Travaux Dirigés

Pascale Ho - 2019 - Ingénieur IMAC

---

## #01 Formulaires

### Table de multiplication

1. Écrivez un (seul) script qui saisit dans un champ texte un entier et qui **affiche la table de multiplication de base 10** de cet entier lors de l'envoi (vérifiez que la valeur soit bien un entier).

Par exemple, si je mets 5 dans mon formulaire, cela devrait m'afficher la table de multiplication de 5 (5x1, 5x2, ... 5x10) .

### Choix Multiples

1. Dans un fichier **formulaireFruits.html**, créez un formulaire avec des checkbox (cases à cocher) de fruits.

```
<form method="POST" action="afficheFruits.php">
...
</form>
```

2. Dans un fichier **afficheFruits.php**, faites en sorte d'afficher les fruits qui ont été cochés dans le formulaire. Affichez un message de ce type :

```
J'adore les pommes, les oranges, les bananes et les fraises.
```

Évidemment faites un affichage différent s'il n'y a aucune case de cochée par exemple.

## #02 Requêtes avec AJAX

1. Créez un fichier **formulaire.html** qui va contenir un formulaire avec un champ pour le prénom (firstname), le nom (lastname), la date de naissance (birthday), un bouton “Envoyer”.
2. Faites en sorte avec Javascript de lancer avec AJAX **une requête HTTP GET** dont le chemin de l'URL est “**salutation.php**”. Il va falloir composer les queries en fonction des valeurs renseignées dans le formulaire.

Le script ressemblera à ceci (ne faites pas de copier-coller svp, la copie ne va pas se faire correctement pour je ne sais quelle raison):

```
const button = ... ; // à compléter
button.onclick = event => {
    event.preventDefault(); // évite de recharger la page
    const form = ... ; // à compléter

    // construction des queries
    let params = {};
    if (form.firstname.value) params['firstname'] =
form.firstname.value;
    if (form.lastname.value) params['lastname'] = form.lastname.value;
    if (form.birthday.value) params['birthday'] = form.birthday.value;

    // création de l'URL et de ses queries (à modifier)
    // on indique en second paramètre du constructeur URL la racine du
    // site (ici http://localhost)
    let url = new URL("/chemin/vers/salutation.php",
"http://localhost");
    url.search = new URLSearchParams(params);
    console.log(url);

    // faire la requête ici (avec fetch ou xmlhttprequest)
    // ...
}
```

3. Créez un fichier **salutation.php** qui doit recevoir les informations du formulaire et renvoyer en JSON un message de ce type (que vous pouvez mettre en forme côté client quand il recevra la réponse HTTP):

```
Bonsoire, je m'appelle Pascale Ho et j'ai 25 ans.
```

**Conseils:** Des vérifications doivent être effectuées côté serveur:

- Est-ce qu'on a bien envoyé une requête **GET** ? Si ce n'est pas le cas retournez (toujours en JSON) le message suivant: *'This method is not allowed.'* (Cherchez dans les internets s'il n'existe pas de code statut correspondant à cette erreur).
- Est-ce qu'on a bien rempli notre formulaire ? **Est-ce les informations envoyées sont correctes ?** (bon format, non vides, ...) Si ce n'est pas le cas retournez un message du type: *'xx param is missing'* ou *'xx param is incorrect'* ou *'Incorrect parameters'*, ...
- Une fois que tout est correct, on doit trouver l'âge de la personne à partir de la date de naissance.
- Enfin on envoie le message (avec le bon code status bien sûr) !

**Bonus :** Faites en sorte que la première lettre des prénoms et des noms soient en majuscules, et pas les autres. Attention aux noms composés. Ces fonctions PHP peuvent vous aider : **strtolower**, **ucfirst**, **ucwords**

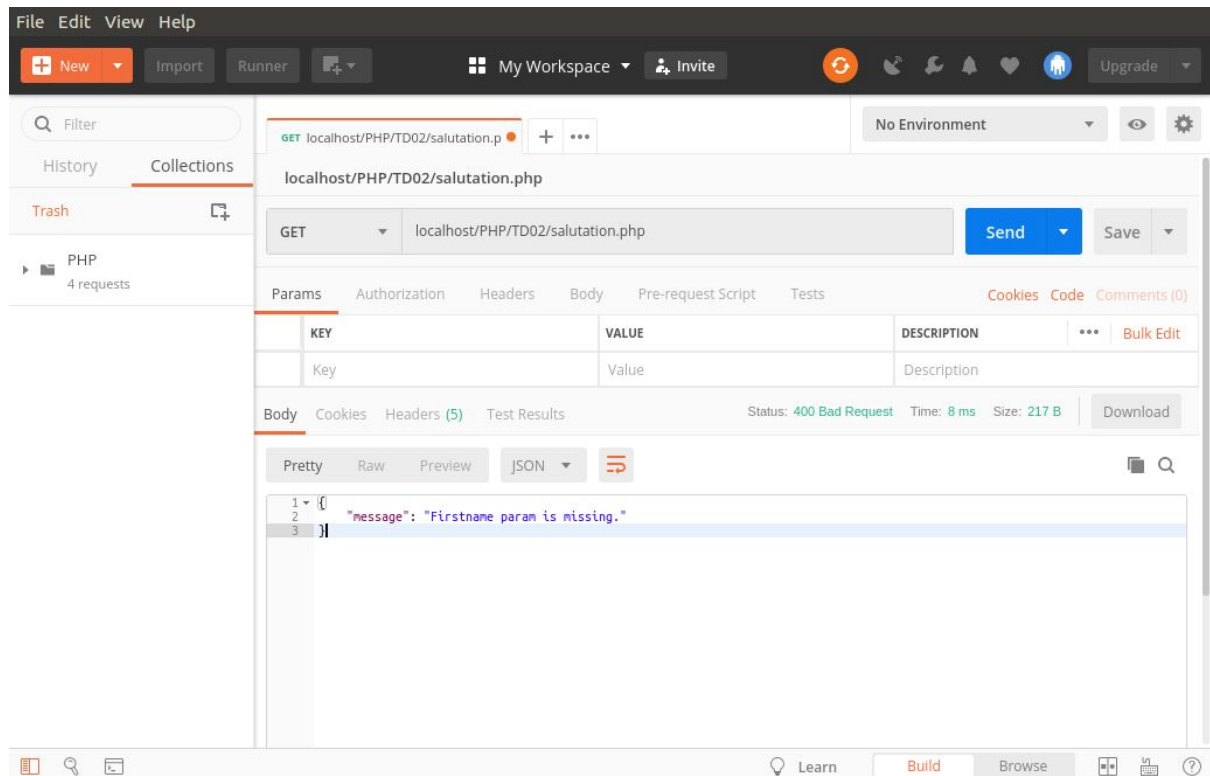
**Bonus :** Personnalisez encore plus le message, genre en mettant "Bonjour" en journée et "Bonsoir" en soirée (il suffit de récupérer l'heure).

**Bonus :** Côté client, vous pouvez donner un meilleur feedback pour le message final ou s'il y a une erreur (genre ajouter des couleurs, faire un pop-up, ...)

## #03 Présentation Postman

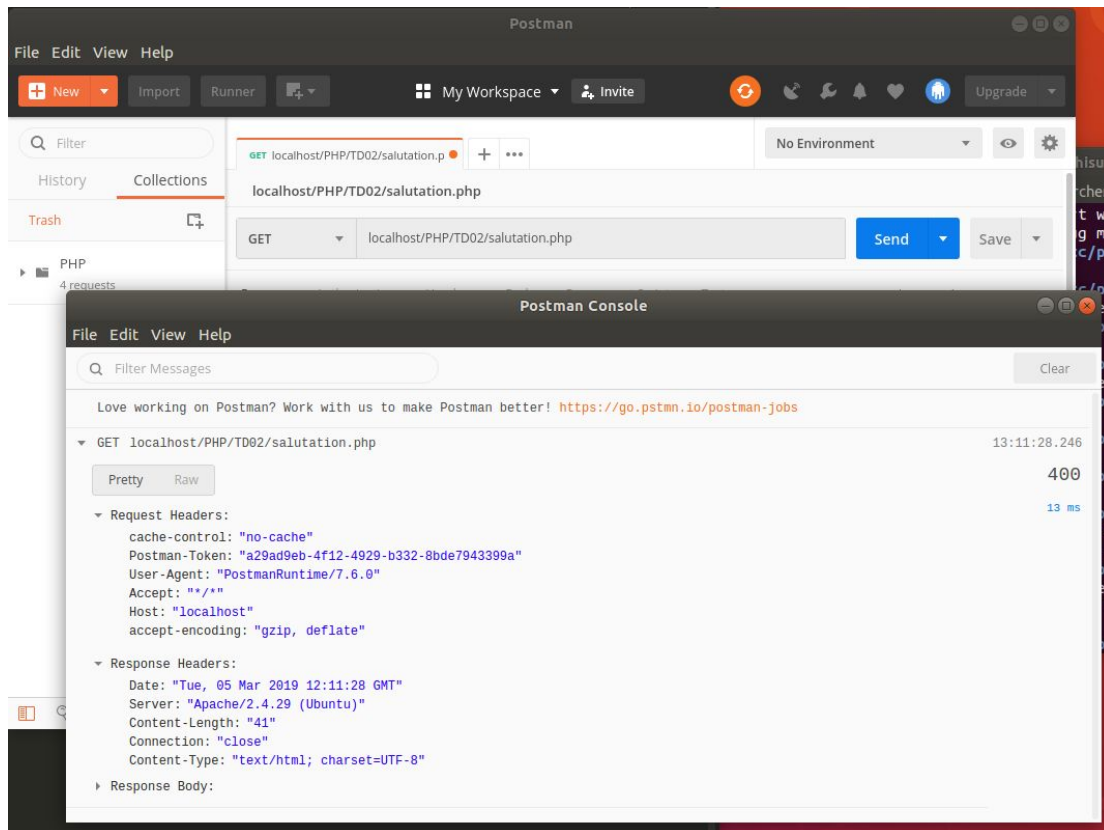
Avant de passer au fil rouge, on va regarder la structure de la requête GET de l'exercice précédent ainsi que sa réponse. On peut se servir du **moniteur Réseau** de nos navigateurs Internet (en faisant clic droit puis "Examiner l'élément") mais il existe des applications utiles pour le développement d'APIs Web qui permettent la construction et l'exécution des requêtes HTTP : le plus connu étant **Postman** qui est une application (Linux/Mac/Windows) mais aussi un module pour Chrome. Si vous voulez un module pour Firefox, il y a le module **RESTClient**. Sur ce TD je vais me servir de l'application.

- Postman (Linux/Mac/Windows, Chrome): <https://www.getpostman.com/>
- RESTClient (Firefox): <https://addons.mozilla.org/fr/firefox/addon/restclient/>

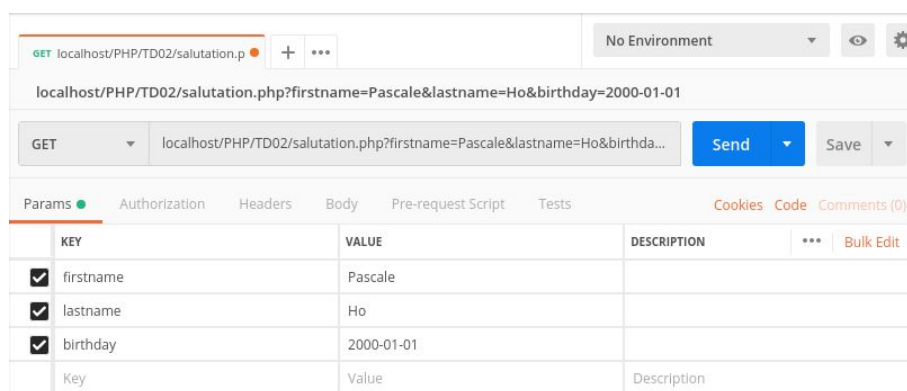


Voici ce qu'on peut observer lorsqu'on lance la requête **GET salutation.php** dans Postman. On sélectionne la méthode de requête HTTP, on saisit notre URI (des paramètres/queries si besoin dans les requêtes GET et des headers supplémentaires pour les requêtes non-GET), lorsqu'on clique sur "Send" on obtient la réponse dans la partie inférieure de l'application (choisissez JSON dans la liste déroulante) (cf image ci-dessus).

Dans l'application Postman, on peut accéder à la "console" (troisième bouton en bas à gauche) où on pourra voir différemment nos échanges HTTP :



Si on a réalisé l'expérience ci-dessus on devrait recevoir une erreur (car on a besoin d'un nom, d'un prénom et d'une date) ! Complétez donc l'URI pour que le script PHP renvoie bien le message de salutation. (Il est possible dans la dernière version de Postman d'indiquer les queries directement dans l'onglet "Params".)



On a juste défini pour la méthode GET. Si on change de méthode HTTP on devrait obtenir une autre erreur si vous avez fait la vérification de la méthode dans le script PHP (essayez pour voir !).

Pour ajouter des headers supplémentaires et un body dans une requête cela se fait dans la partie supérieure, dans les onglets correspondants.

## #04 Fil Rouge

**Rappel : Ce fil rouge n'est pas obligatoire.** Mais plus vous m'envoyez vos travaux plus vous pouvez avoir des **bonus sur votre note finale**. Vous avez donc plusieurs façons de me fournir vos TDs :

- soit en m'envoyant un fichier compressé (.zip, ...) contenant un répertoire avec tous les fichiers nécessaires à l'exercice,
- soit en m'envoyant un repo Git des TDs

À m'envoyer par mail ([hp.pascale.ho@gmail.com](mailto:hp.pascale.ho@gmail.com)) ou via messagerie 2.0 du digital des réseaux sociaux

**Date limite : 22 mars à 8h30 pétantes**

**Le style n'est pas important ! Et si vous avez des questions, n'hésitez pas à me les poser ou à vos camarades !**

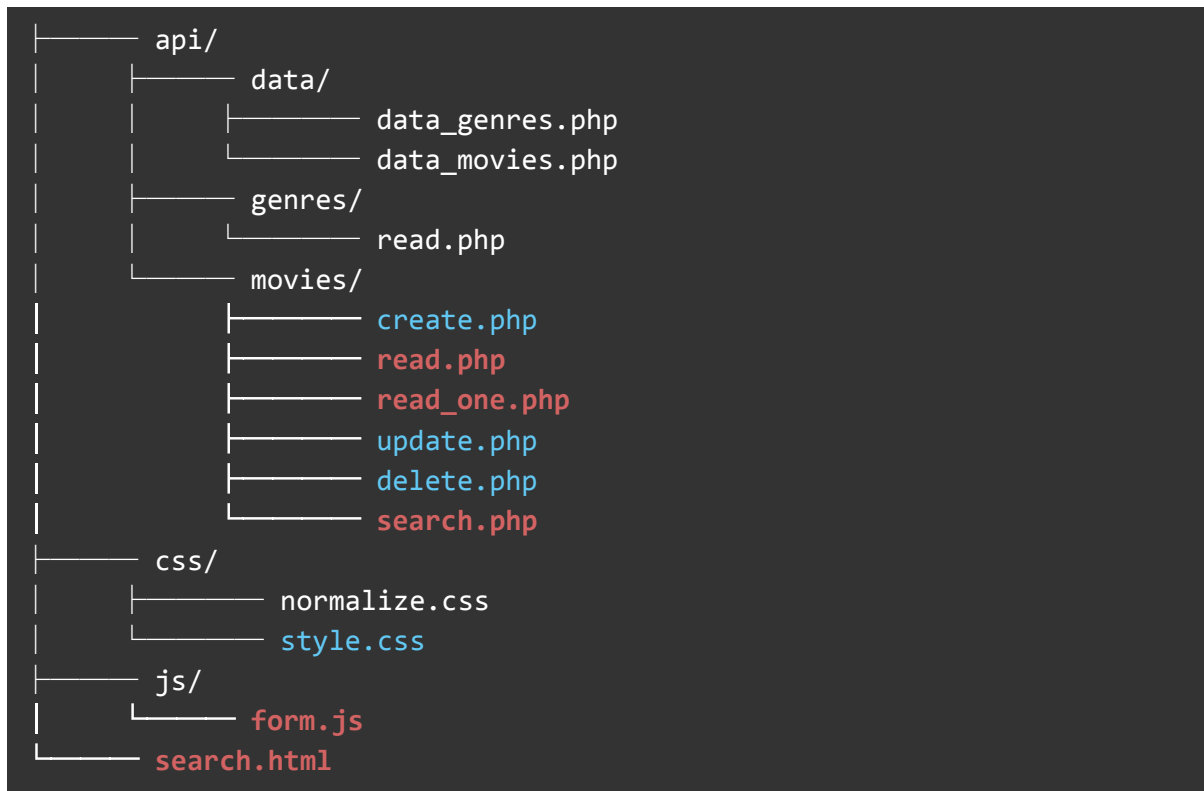
## Sujet

Tout d'abord j'ai modifié le fichier **data\_movies.php** (j'ai rajouté encore plus de films, eh ouais). Prenez ce nouveau fichier et regardez le contenu. Je vous demande de **NE PAS LE MODIFIER** thanks 😊

Cette fois-ci on va construire un formulaire de recherche de films. Puis on va faire quelques requêtes avec Postman. Il s'agit d'une préparation à REST 😊

## Architecture du site

**1.** Téléchargez le zip dans le dossier partagé. L'arborescence du site se trouve ci-dessous. Les fichiers en **rouge** seront les fichiers sur lesquels vous allez travailler. Ceux en **bleu** seront à faire en bonus.



## Formulaire de recherche

**2.** Regardez le fichier **search.html**. À l'intérieur se trouve le formulaire de recherche de films avec des champs pour le titre, la date de sortie du film, un genre et un.e réal. La liste des genres est remplie dynamiquement grâce à la requête vers **/api/genres/read.php**.

Le but est de faire apparaître la liste de films en fonction des valeurs renseignées dans le formulaire.

Complétez le bas du fichier **form.js** en vous servant d'AJAX pour faire une requête **GET** vers **/api/movies/search.php**.

**2.** Complétez **search.php** qui doit retourner en JSON la liste de films correspondants aux critères choisies par l'utilisateur. Attention, cette fois-ci on va comparer des dates complètes avec jour, mois et année. Pour comparer deux dates vous pouvez vous servir de la classe **DateTime** de PHP.

**3.** Recomplétez le fichier **form.js** pour faire afficher les résultats de la recherche (vous pouvez réutiliser le travail du TD précédent pour vous inspirer de la mise en page). N'oubliez pas de faire un affichage quand la recherche a échoué.

### 4. Bonus :

- Utilisez des checkbox pour choisir plusieurs genres de film.

- Plutôt que de chercher exactement le terme exact du titre ou du cast, faites en sorte que la fonction de recherche vérifie que la valeur recherchée est bien une portion du titre du film ou du réalisateur/réalisatrice.
- Comme indiqué dans **search.php**, si aucun paramètre n'est renseigné dans la requête, retournez le contenu de **read.php** (cf exo suivant).

## Requêtes avec Postman

5. Servez-vous de Postman pour faire les requêtes GET suivants:

### GET api/movies/read.php

Liste tous les films. (en JSON !)

---

### GET /api/movies/read\_one.php?id={id}

Affiche le film à l'index **{id}**. On remplace {id} par un entier. Vérifier si cet indice existe bien.

---

#### Paramètres:

id : index de l'élément du film dans le tableau de films. Doit être un entier.

---

6. **Bonus** : On ne peut pas encore faire la création/modification/suppression de nos informations car on ne se sert pas encore de bases de données ! Mais on va faire une simulation. Faites les endpoints suivants:

### POST /api/movies/create.php

Création d'un film. Renvoie un message de confirmation avec les informations du film si tout s'est bien déroulé (un message d'erreur sinon).

---

#### Exemple de corps de requête:

```
{
  "title": "Titre d'un film",
  "date": "2019-01-01",
  "genre": "genre" OU ["genre1", "genre2", ...],
  "director": "Nom réal" OU ["Nom réal1", "Nom réal2", ...]
}
```

---

### PUT /api/movies/update.php

Modification d'un film. Renvoie un message de confirmation avec les informations du film modifié si tout s'est bien déroulé (un message d'erreur sinon).

---

**Exemple de corps de requête:** (les parties en italiques ne doivent pas être

---



---

obligatoirement présentes)

```
{  
  "id": 1,  
  "title": "Titre d'un film",  
  "date": "2019-01-01",  
  "genre": "genre" OU ["genre1", "genre2", ...],  
  "director": "Nom réal" OU ["Nom réal1", "Nom réal2", ...]  
}
```

---

#### **DELETE /api/movies/delete.php**

Suppression d'un film. Renvoie un message de confirmation avec les informations du film supprimé si tout s'est bien déroulé (un message d'erreur sinon).

---

#### **Exemple de corps de requête:**

```
{  
  "id": 1  
}
```

---