

# Atelier pratique sur les méthodes GET et POST en PHP

## Objectif

Cet atelier a pour objectif de vous familiariser avec les deux méthodes HTTP GET et POST utilisées en PHP pour gérer les formulaires Web. Pour chaque exemple, suivez les étapes suivantes :

1. Écrivez le code dans un éditeur de texte (par exemple, VS Code ou Notepad++).
2. Enregistrez chaque fichier avec les extensions appropriées.
3. Testez vos fichiers dans un navigateur Web.

## 1 Configuration de l'environnement de travail

### 1.1 Installation et configuration de WAMP Server

Pour réaliser les exercices de ce TP, vous aurez besoin d'un serveur web local incluant PHP. Nous utiliserons WAMP Server qui fournit un environnement complet (Windows, Apache, MySQL, PHP).

#### Installation de WAMP Server

1. Téléchargez la dernière version de WAMP Server sur <https://www.wampserver.com>
2. Suivez les instructions d'installation
3. Vérifiez que l'installation est correcte en vous assurant que l'icône WAMP est verte dans la barre des tâches

### 1.2 Organisation des fichiers de travail

Les fichiers HTML et PHP doivent être placés dans le dossier racine du serveur web ou dans un sous-dossier de celui-ci.

```
1 C:\wamp64\www\           # Dossier racine de WAMP Server
2     tp_formulaires\      # Dossier de notre TP
3     index.php           # Page d'accueil du TP
```

```
4     recherche_get.php    # Exercice 1
5     inscription_post.php # Exercice 2
6     commande_upload.php # Exercice 3
```

#### Accès aux fichiers du TP

Pour accéder à vos fichiers :

1. Démarrez WAMP Server (icône verte dans la barre des tâches)
2. Ouvrez votre navigateur web
3. Accédez à l'URL : `http://localhost/tp_formulaires/`
4. Pour accéder à un exercice spécifique :  
`http://localhost/tp_formulaires/recherche_get.php`

### 1.3 Structure recommandée pour le TP

Pour faciliter le travail et l'organisation, voici une structure recommandée pour ce TP :

#### Structure recommandée

- Créez un fichier `index.php` qui servira de page d'accueil et contiendra :
  - Des liens vers chaque exercice
  - Des liens vers les questionnaires de compréhension
  - Une brève description de chaque partie
- Pour chaque exercice, créez un fichier PHP distinct
- Si nécessaire, créez un dossier `css` pour les feuilles de style
- Créez un dossier `uploads` avec les permissions d'écriture pour l'exercice avec upload de fichiers

### 1.4 Points importants pour les chemins dans les formulaires

Lorsque vous définissez l'attribut `action` de vos formulaires, soyez attentif aux chemins :

```
1 <!-- Si le traitement se fait dans le m me fichier -->
2 <form action="" method="post">
3
4 <!-- Si le traitement se fait dans un fichier du m me
   dossier -->
5 <form action="traitement.php" method="post">
6
```

```
7 <!-- Si le traitement se fait dans un fichier d'un sous-  
   dossier -->  
8 <form action="traitement/validation.php" method="post">  
9  
10 <!-- Si le traitement se fait dans un fichier du dossier  
    parent -->  
11 <form action="../../autre_traitement.php" method="post">
```

#### Attention aux droits d'accès

Si vous rencontrez des erreurs lors de l'upload de fichiers ou de l'écriture dans des fichiers :

- Vérifiez que le dossier `uploads` a les permissions d'écriture
- Sous Windows, faites un clic droit sur le dossier → Propriétés → Onglet Sécurité → Modifier les permissions
- Ajoutez l'utilisateur IUSR avec des droits d'écriture ou vérifiez que le groupe "Utilisateurs" a des droits d'écriture

## Exécuter les fichiers PHP et HTML avec XAMPP

### 1. Installer et démarrer XAMPP

- Télécharger XAMPP depuis le site officiel.
- Lancer XAMPP et démarrer les modules **Apache** (et **MySQL** si nécessaire).

### 2. Organiser les fichiers dans le dossier approprié

Placer les fichiers PHP et HTML dans le dossier :

```
1 C:\xampp\htdocs\tp_formulaires\
```

### 3. Accéder aux fichiers via le navigateur

Ouvrir un navigateur web et saisir :

```
1 http://localhost/tp_formulaires/
```

ou pour un fichier précis :

```
1 http://localhost/tp_formulaires/index.php
```

## 4. Vérifier le bon fonctionnement de PHP

Créer un fichier nommé `test.php` dans le dossier du projet avec :

```
1 <?php
2 phpinfo();
3 ?>
```

Accéder à ce fichier via :

```
1 http://localhost/tp_formulaires/test.php
```

## 5. Structure recommandée

Voici une structure organisée :

```
1 htdocs/
2 |-- tp_formulaires/
3     |-- index.php
4     |-- recherche_get.php
5     |-- inscription_post.php
6     |-- commande_upload.php
7     |-- css/ (optionnel)
8     |-- uploads/ (permissions d'écriture nécessaires)
```

## 6. Remarques importantes

- Toujours enregistrer les fichiers PHP avec l'extension `.php` et HTML avec l'extension `.html`.
- Attention à l'attribut `action` dans les formulaires :

**Exemple :**

```
1 <!-- Traitement dans le m e m e fichier -->
2 <form action="" method="post">
3
4 <!-- Traitement dans un autre fichier -->
5 <form action="traitement.php" method="post">
```

## 2 Vérification de l'environnement

Avant de commencer les exercices, créez un fichier `test.php` dans votre dossier de travail avec le code suivant :

```
1 <?php
2     phpinfo();
3 ?>
```

Accédez à ce fichier via `http://localhost/tp_formulaires/test.php` pour vérifier que PHP fonctionne correctement et pour connaître sa configuration.

### 3 Exemple 1 – Formulaire avec la méthode GET

#### Fichier HTML (form\_get.html)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formulaire GET</title>
5 </head>
6 <body>
7   <form action="resultat_get.php" method="get">
8     Nom: <input type="text" name="nom">
9     <input type="submit" value="Envoyer">
10  </form>
11 </body>
12 </html>
```

#### Fichier PHP (resultat\_get.php)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>R sultat GET</title>
5 </head>
6 <body>
7   Bonjour, <?php echo htmlspecialchars($_GET['nom']); ?> !
8 </body>
9 </html>
```

### 4 Exemple 2 – Formulaire avec la méthode POST

#### Fichier HTML (form\_post.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulaire POST</title>
</head>
<body>
  <form action="resultat_post.php" method="post">
    Nom: <input type="text" name="nom">
    <input type="submit" value="Envoyer">
  </form>
</body>
</html>
```

## Fichier PHP (resultat\_post.php)

```
<!DOCTYPE html>
<html>
<head>
  <title>Résultat POST</title>
</head>
<body>
  Bonjour, <?php echo htmlspecialchars($_POST['nom']); ?> !
</body>
</html>
```

## Exercices pratiques

1. Ajoutez à chacun des formulaires un champ supplémentaire « âge ».
2. Modifiez les pages résultats pour afficher le nom et l'âge envoyés par l'utilisateur.
3. Notez la différence observée dans les URLs des pages résultats selon que vous utilisez GET ou POST.

## Observation importante

- **Méthode GET** : Les données sont visibles directement dans l'URL.
- **Méthode POST** : Les données ne sont pas visibles dans l'URL, méthode recommandée pour des formulaires sensibles (connexion, inscription, etc.).

Bon travail !