

Méthodes GET et POST

Dans le cas de sites Web modernes et interactifs, les clients (c'est-à-dire les navigateurs Web) récupèrent non seulement un document HTML sur le serveur, mais lui envoient aussi souvent des informations comme :

- Le texte d'un terme de recherche que l'utilisateur a saisi dans le champ de recherche
- Le contenu d'un formulaire rempli
- La sélection des filtres dans une boutique en ligne
- Le tri d'une liste

Pour la transmission de ces informations au serveur, le protocole HTTP prévoit différentes **méthodes de requête**. Les deux plus importantes sont GET et POST. Bien que les deux aboutissent au même résultat, elles sont fondamentalement différentes. Voici dans cet article un aperçu de leurs différences et nos explications pour savoir quand les méthodes GET ou POST sont les plus appropriées.

GET

Avec la méthode GET, les données à envoyer au serveur sont **écrites directement dans l'URL**. Dans la fenêtre de votre navigateur, cela ressemble à ceci :

```
www.example.com/register.php?firstname=peter&name=miller&age=55&gender=male
```

Toutes les informations saisies par l'utilisateur (les paramètres dits URL) sont transmises aussi librement que l'URL elle-même. Cela présente des avantages et des inconvénients.

Avantages

Les paramètres de l'URL peuvent être enregistrés avec **l'adresse du site Web**. Cela permet de mettre une requête de recherche en marque-page et de la récupérer plus tard. Si nécessaire, la page peut également être récupérée via l'historique de navigation.

Cela est utile pour visualiser par exemple régulièrement une section de carte Google Maps ou pour enregistrer des pages Web contenant certains paramètres de filtrage et de tri.

Inconvénients

Le principal inconvénient de la méthode GET est **l'absence de protection des données**. Les paramètres URL envoyés sont non seulement visibles par tous dans la barre d'adresse du navigateur, mais sont également stockés sans chiffrement dans l'historique du navigateur, dans le cache et dans le fichier log du serveur.

Un deuxième inconvénient est **sa capacité limitée** : suivant le serveur Web et le navigateur, l'URL ne peut pas contenir plus de 2 000 caractères environ. De plus, les paramètres des URL ne peuvent contenir que des caractères ASCII (lettres, chiffres, caractères spéciaux, etc.), et non des données binaires telles que des fichiers audio ou des images.

POST

La méthode POST écrit les paramètres URL dans **la requête HTTP** pour le serveur. Les paramètres ne sont donc pas visibles pour les utilisateurs et la portée des requêtes POST est illimitée.

Avantages

Lorsqu'il s'agit de transmettre des données sensibles au serveur, par exemple un formulaire d'inscription avec nom d'utilisateur et mot de passe, la méthode POST permet de garder **la confidentialité** nécessaire. Les données ne sont pas mises en cache et n'apparaissent pas dans l'historique de navigation. **La flexibilité** est également de mise avec POST : non seulement des textes courts, mais aussi des données de toute taille et de tout type peuvent être transmis, comme des photos ou des vidéos.

Inconvénients

Si une page Web est mise à jour avec un formulaire dans le navigateur (par exemple, en utilisant le bouton « Précédent » / « Retour »), les données du **formulaire doivent être de nouveau soumises**. Vous avez certainement déjà vu des avertissements qui s'y réfèrent. Il existe un risque que les données soient envoyées plusieurs fois par inadvertance, ce qui peut déclencher des commandes en double par exemple. Toutefois, les programmes de boutiques en ligne les plus récents peuvent empêcher ces problèmes.

De même, les données transmises par la méthode POST **ne peuvent pas être sauvegardées sous forme de marque-page** avec l'URL.

Comparaison entre GET et POST

	GET	POST
Visibilité	Visible pour l'utilisateur dans le champ d'adresse	Invisible pour l'utilisateur
Marque-page et historique de navigation	Les paramètres de l'URL sont stockés en même temps que l'URL.	L'URL est enregistrée sans paramètres URL.
Cache et fichier log du serveur	Les paramètres de l'URL sont stockés sans chiffrement	Les paramètres de l'URL ne sont pas enregistrés automatiquement.
Comportement lors de l'actualisation du navigateur / Bouton « précédent »	Les paramètres de l'URL ne sont pas envoyés à nouveau.	Le navigateur avertit que les données du formulaire doivent être renvoyées.
Type de données	Caractères ASCII uniquement.	Caractères ASCII mais également données binaires.
Longueur des données	Limitée - longueur maximale de l'URL à 2048 caractères.	Illimitée.

Quand utiliser quelle méthode ?

POST est presque toujours favorisé lorsque l'utilisateur doit soumettre des données ou des fichiers au serveur, par exemple pour remplir des formulaires ou télécharger des photos.

GET est particulièrement bien adapté pour personnaliser les sites Web : les recherches des utilisateurs, les paramètres de filtrage et le tri des listes peuvent être mis en marque-page avec l'URL, de sorte qu'à la prochaine visite du site, l'utilisateur retrouvera la page telle qu'il l'a laissée.

Une simple « règle de base » pour finir :

- **GET** pour les paramètres d'un site Web (filtres, tri, saisies de recherche, etc.).
- **POST** pour la transmission des informations et des données de l'utilisateur.

Dans le cadre de mon stage j'ai utilisé ces méthodes pour le transfert des informations entre les pages du site Web

```
<?php
if (isset($_POST['valider'])) {

    $grade_cdc = htmlspecialchars($_POST['grade']);
    $nom_cdc = htmlspecialchars($_POST['nom']);
    $prenom_cdc = htmlspecialchars($_POST['prenom']);
    $biographie_cdc = htmlspecialchars($_POST['biographie']);
    $nom_photo = htmlspecialchars($_POST['nom_photo']);
    $date_entree_fonction = date("Y-m-d H:i:s", strtotime($_POST['date_entree_fonction']));

    $date = date_create();
    $sous_dossier = date("Ymd", strtotime($date_entree_fonction)).'_'.date_timestamp_get($date);

    //requête SQL:
    $sql = mysqli_query($bdd, "INSERT INTO 40rtrs_cdc(grade, nom, prenom, biographie, photo, date_entree_fonction)
        VALUES ('".$grade_cdc."','".$nom_cdc."','".$prenom_cdc."', '".$biographie_cdc."', '".$nom_photo."', '".$date_entree_fonction."')");
} else {
    echo 'Formulaire non reçu';
}
?>
```