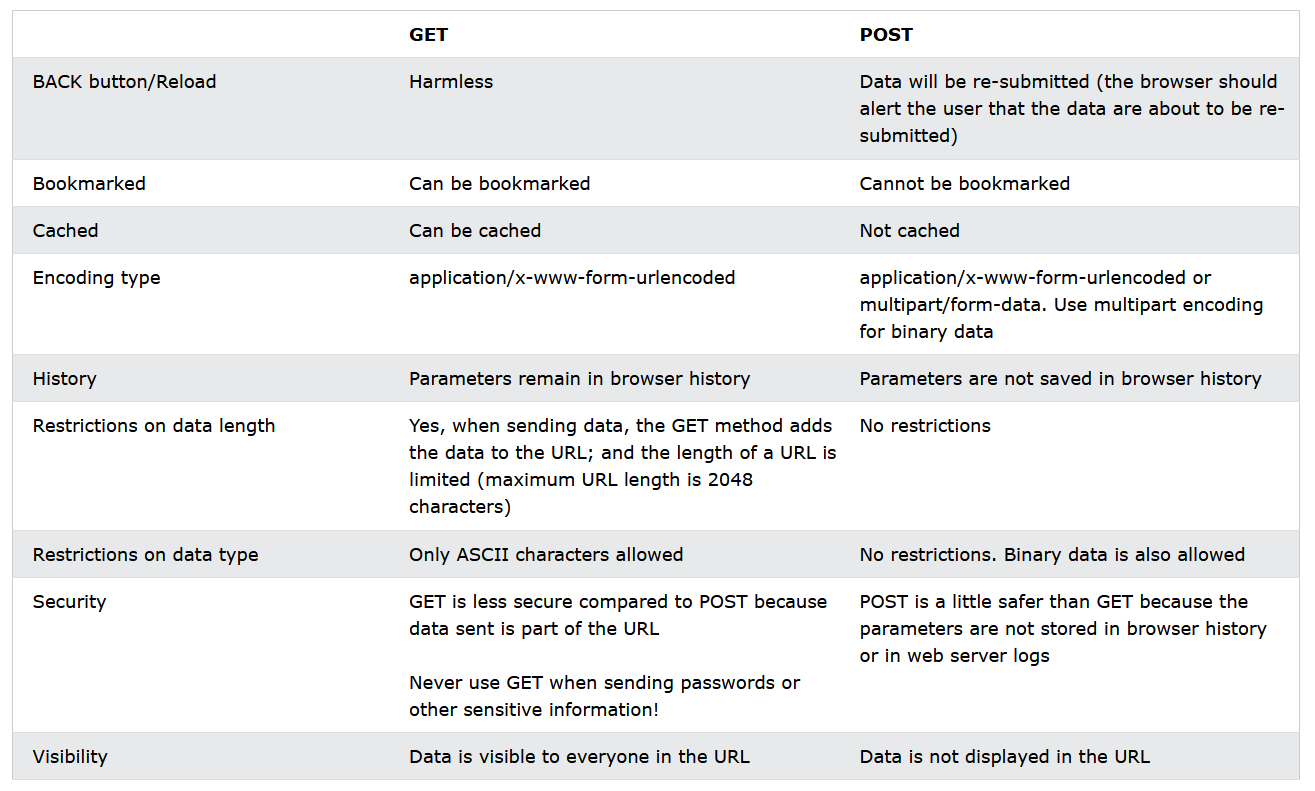
La superglobale POST

# Introduction

Comme la superglobale GET, la super globale POST est utilisé pour transmettre des informations au serveur via une requête HTTP(S).

Il y a cependant des différences entre c’est deux superglobale.

[Lien](https://www.w3schools.com/tags/ref_httpmethods.asp)



Si on résume la superglobale GET est principalement utilisé pour spécifier le contenu que l’on veut voir affiché. Tandis que POST a plus vocation à effectuer des opérations sur les données (ajout, modification, suppression). Contrairement a GET les informations ne sont pas transmissent via l’URL.

# Effectuer une requête http de type Post

Il existe plusieurs moyens d’effectuer une requête de type POST, mais le principal moyen consiste à envoyer un formulaire.

Chaque champs (input, select, textarea) doit disposer d’un attribut name est d’une valeur.

L’ors de l’envoi du formulaire les données seront transmises à la cible de l’attribut action de la balise form.

ATTENTION : l’attribut method de la balise form dois forcement avoir comme valeur POST.

Le fichier qui est la cible de l’action va alors recevoir les données transmissent sous la forme d’un array associatif où les clés sont les name des champs et les valeurs les values des champs.

Le processus qui vise à vérifier que les données envoyées sont bien correctes s’appelle la validation.

La validation doit forcement au minimum être faite du coté serveur mais il est également recommandé de la faire coté client afin d’évité d’envoyer un formulaire incorrect.

La validation coté serveur est également accompagné de la « sanitisaiton » un procesus qui vise a vérifié que les données ne contiennent pas d’injection de code.

La faille de sécurité XSS (Cross Site Scripting) (un nom plutôt mal choisi) consiste a envoyé des balises html surtout de balises scripts (JS) dans les champs du formulaire, ou en ce qui concerne GET directement dans l’URL.

Afin d’évité ce type d’injection on utilise la fonction filter\_var sur toutes informations reçus en GET ou POST avec un filtre de sanitize.

Par exemple :

$nom=filter\_var($\_POST["nom"],FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);

A partir de maintenant on ne manipulera plus $\_POST[‘nom’} mais $nom qui a était filtré est qui est maintenant sure.

<https://www.php.net/manual/fr/function.filter-var.php>

<https://www.w3schools.com/php/php_filter.asp>

Les filtres de nettoyages :

<https://www.php.net/manual/fr/filter.filters.sanitize.php>

Les filtres de validation :

<https://www.php.net/manual/fr/filter.filters.validate.php>

## Exercice 1 : Plusieurs pages on était ajouté au projet stagiaire :

* AdminStagiaire
* formAjoutStagiaire
* traitementAjoutStagiaire

Essayer de comprendre le role de chaque page.

## Exercice 2 : La page adminStagiaire a plusieurs boutons et form sur lesquelles il manque des hrefs. Créer les pages et ajouté les liens (certaine page aura besoin de l’id transmis en GET).

## Un formulaire de modification d’un stagiaire

## Une page pour consulter un stagiaire (déjà fait)

## Une page de traitement du formulaire de suppression

## Exercice 3 : A vous de jouer faite un tableau, formulaires et page de traitement pour les destinations (ou le projet de votre choix)

## 