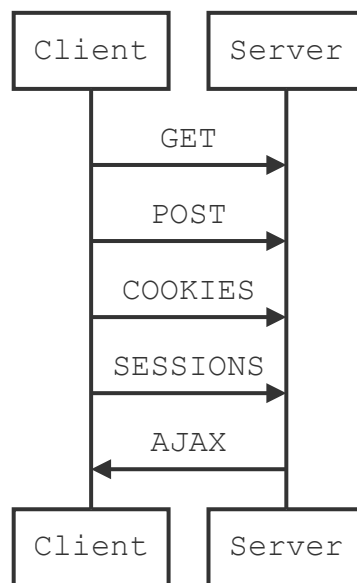


Accès Aux Données

La plus part des informations échangées entre un serveur et son client ne sont pas toujours sous la forme de documents php. Pour rappel, le php est *stateless*¹. Chaque requête doit donc être traitée de manière indépendante.

Dans la majorité des cas, c'est le client qui enverra ses informations à direction du serveur qui traitera la requête en fonction.



- **GET** : informations dans l'**adresse** de la requête
- **POST** : informations dans l'**entête** de la requête
- **Cookies** : informations **stockées chez le client**
- **Sessions** : informations **stockées sur le server**

Get et Post

GET

Get récupère les informations placées dans l'entête de l'url.

1 | `www.monsite.com/page? clef1 = val1 & clef2 = val2`

Attention, pas plus de 2048 caractères

-

- Méthode la moins sur car les données sont visibles dans l'url
- Données facilement interceptables
- Données retrouvables dans un historique

+

- Données retrouvables dans un historique
- Possibilité de partager un lien

Récupérer les données

Reprenons l'exemple :

```
1 | www.monsite.com/page? clef1 = val1 & petit = gregory
```

en php :

```
1 | $_Get['clef1'] //donne val1;  
2 | $_Get['petit'] //donne gregory;
```

Attention aux valeurs alphanumériques (espaces,=,&,/,;,...)

JS : cfr encodeURIComponent(string)

PHP : urlencode(e), urldecode(e)

\$_Get sont déjà urldécodés

POST

Les informations rentrées dans un formulaire html sont envoyées séparément et leur format n'est pas visible.

L'utilisateur remplit totalement / ou partiellement (aide de js) un formulaire HTML

CFR : Slide 15/70 : Formulaires HTML

-

- moins facile à mettre en place

+

- Pas de limite de Taille
- Laisse moins de traces(historique)

Exemple :

```
1 | <form method="post" action="destination.php">
```

Réception du post

code du formulaire :

```
1 | <input name="myNom" type="text" value="gregory"/>
```

code dans votre fichier de destination php

```
1 | $_POST["myNom"] // donnera gregory
```

Javascript et Formulaires

```
1 var myFormulaire = document.forms[index];
2 var myElem = myFormulaire.elements;
3
4 myElem[2].checked = true; //coche un radio button
5 myElem[5].disabled= true // empêche l'activation d'un élément (ex:bouton pour
  soumettre le formulaire)
6
7
```

On peut cacher certains champs et les révéler avec du JS pour les rendre interactifs

```
1 | <input type="hidden" ...>
```

on peut aussi valider un formulaire via du javascript

cfr slide 18 part 5

Types de formulaires

cfr Slide 20 _ part 5

1. Ne possède pas de mémoire [↗](#)