



PHP ENVOI DE DONNÉES DANS LA REQUÊTE



Aurélien Delorme
OCT 1



Envoie de données depuis l'URL

Récupérer les données envoyées dans l'URL

Les formulaires / La méthode POST

Envoyer des données dans un formulaire

Upload de fichiers

Session

Cookies

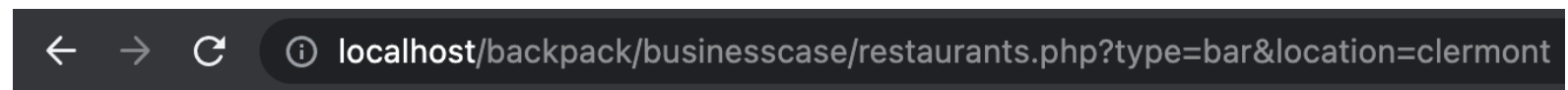




Envoi de données depuis l'URL

Consiste à envoyer des données depuis l'url

Exemple : Aller sur la page des restaurant mais affichez seulement les bars qui se situent à Clermont




Ici, nous transmettons 2 données :

- La première « type » qui vaut 'bar'
- Le ? Indique que c'est donnée envoyée
- La deuxième location qui vaut « clermont »
- Le & indique qu'il s'agit d'une donnée supplémentaire envoyée
- Nous pouvons en ajouter autant que l'on souhaite
- Dans la limite de la taille de la longueur maximale d'une URL (2048 caractères)



Récupérer les données envoyées dans l'URL

www > php-backpack > businesscase >  restaurants.php

```
<?php  
var_dump($_GET);  
?>
```

Utilisation d'une variable super globale

```
1 <?php  
2 var_dump($_GET);  
3 $type = $_GET['type'];  
4 $location = $_GET['location'];  
5 var_dump($type);  
6 var_dump($location);  
7 die();  
8
```

Il en existe plusieurs. Dans notre cas nous utiliserons \$_GET



Les formulaires / La méthode POST

Les données ne sont plus envoyés via l'URL mais dans le contenu du corps de la requête

←

→

↺

ⓘ localhost

Username :

Password :

Valider

localhost/connect.php

```
./var/www/html/connect.php:2
array (size=2)
  "login" => string 'aurelien' (length=8)
  "password" => string 'password' (length=8)
```

Bienvenue aurelien!

Network

connect.php

Headers

Preview

Response

Initiator

Timing

Cookies

Referer: http://localhost/
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.3987.132 Safari/537.36

Form Data
login: aurelien
password: password

▼ Form Data

view source

view URL encoded

login: aurelien

password: password



Les formulaires / La méthode POST

Envoi de données : Exemple (formulaire)

Création d'un formulaire avec : method='post'

```
<form method="post" action="connect.php">
  <label>Username : </label>
  <input name="login" type="text" placeholder="Utilisateur"> <br>
  <label>Password : </label>
  <input name="password" type="password" placeholder="Mot de passe"> <br>
  <input type="submit" placeholder="Mot de passe">
</form>
```

www > connect.php >

```
<?php
var_dump($_POST);
if($_POST['login'] != "" && $_POST['password'] != "") {
    echo("<h1> Bienvenue ". $_POST['login']. "!");
} else {
    echo('Probleme d\'authentification veuillez réessayer');
}
?>
```



Les formulaires / La méthode POST

Envoi de données : Exemple (utilisation de Postman)

http://localhost/conne + ... No Environment

POST ▼ http://localhost/connect.php Params Send ▼

Authorization Headers **Body** • Pre-request Script Tests

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

	Key	Value	Description
<input checked="" type="checkbox"/>	login	aurelien	
<input checked="" type="checkbox"/>	password	password	
	New key	Value	Description

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 36 ms

Pretty Raw Preview

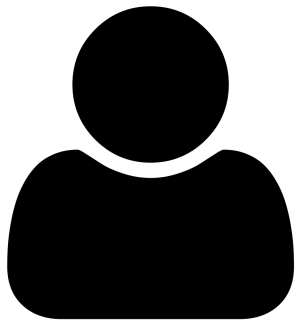
```
/var/www/html/connect.php:2:
array (size=2)
  'login' => string 'aurelien' (length=8)
  'password' => string 'password' (length=8)
```

Bienvenue aurelien!<https://www.postman.com/>



Session

Permet de stocker des éléments de manière temporaire



Utilisée notamment pour les sessions utilisateur. Lorsqu'un utilisateur se connecte à un site internet, ces informations sont stockées en session. Lorsqu'il se déconnecte, sa session est détruite.



Stockage des produits ajoutés dans un panier pour un utilisateur non authentifié.



Session

```
<?php  
session_start();  
?>
```

Démarre la session. Est nécessaire à chaque fois que l'on va appeler nos variables de session

- L'utilisateur a déjà une session, elle est maintenant accessible
- L'utilisateur n'a pas de session : Une session lui sera créée

```
<?php  
session_destroy()  
?>
```

Ferme la session. La session sera automatiquement supprimée après une durée définie. Si l'utilisateur clique sur se déconnecter, il va néanmoins falloir la détruire manuellement grâce à `session_destroy`

La durée de vie d'une session est définie dans la configuration php (php.ini).



Lecture écriture dans la session

```
www > index.php >
```

```
session_start();  
$_SESSION['username'] = 'Aurélien';
```

Le script ci-dessus démarre la session et **enregistre la variable de session username avec la valeur Aurélien**

```
www > login.php >
```

```
<?php  
session_start();  
echo($_SESSION['username']);  
?>
```

Notre variable de session sera accessible sur **toutes les applications et pages** de notre serveur !

Ici : Mon fichier index.php crée la variable de session et le fichier login.php l'affiche !



Cookies

Quasiment similaire à la session en revanche :

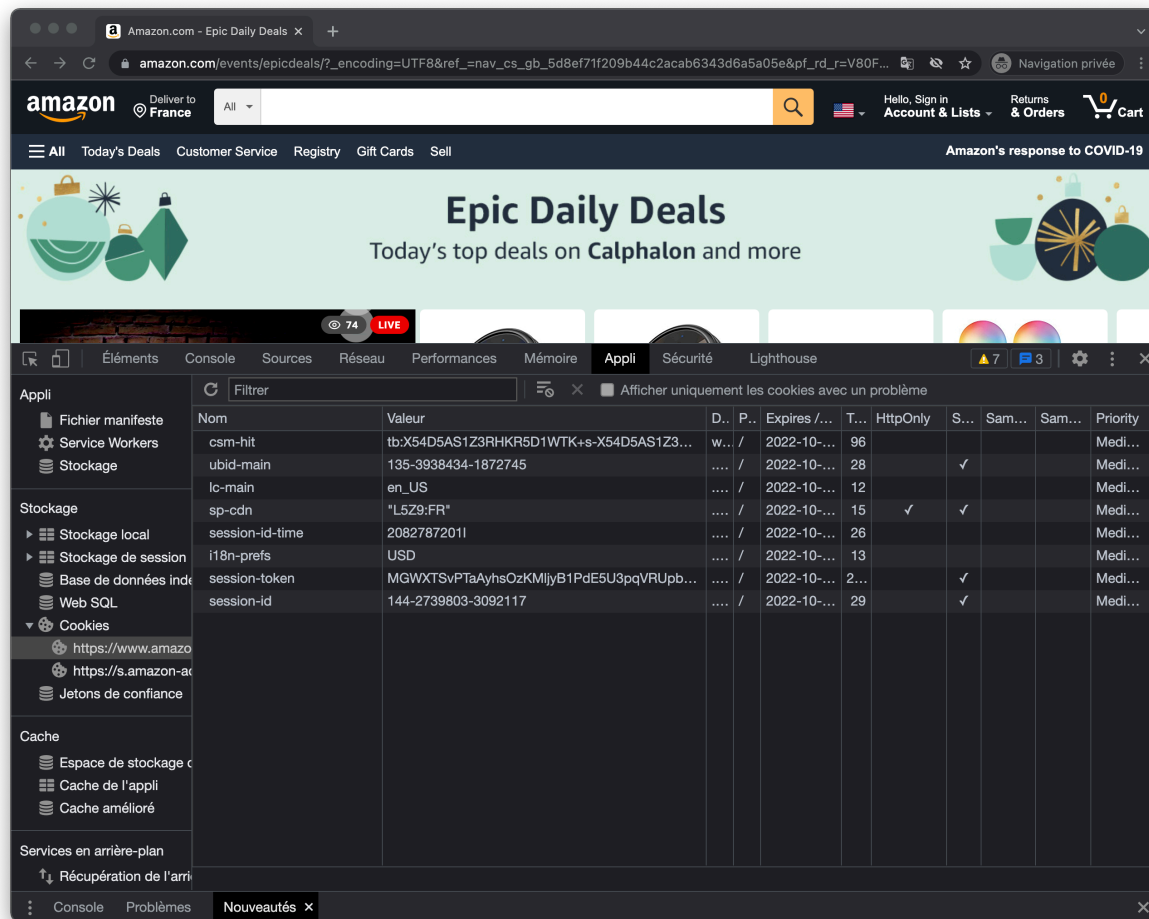
- La **session est stockée sur le serveur** alors que le **cookie est stocké sur le navigateur**.
- Il est distribué sous forme de fichier texte.
- Il a lui aussi un temps de vie maximum. Il sera **automatiquement supprimé à la fin de son temps de vie**
- Si l'utilisateur vide le cache de son navigateur, il sera aussi détruit.



© Thimothée / Femme Actuelle



Cookies

Afficher la liste des cookies dans un navigateur web

Le site web [amazon.com](https://www.amazon.com) stock entre autre la langue utilisé pour que lors de la prochaine visite, le site web apparaisse dans la langue choisie.

Si on change la valeur contenu dans ce cookie à la main, le site sera dans la langue choisie.

Attention, les cookies sont stockés sur le client !
Il ne faut donc pas **stocker d'informations confidentielles / représentant des failles de sécurité**

Pour utiliser des cookies, il faut **impérativement demander à l'utilisateur d'accepter** leur utilisation !
Et indiquer si les fins sont commerciales ou techniques.



Cookies

Ecrire un cookie

Utilisation de la fonction php setcookie.

Elle prend 3 paramètres :

- Le nom du cookie

- La valeur du cookie

- Son temps de vie

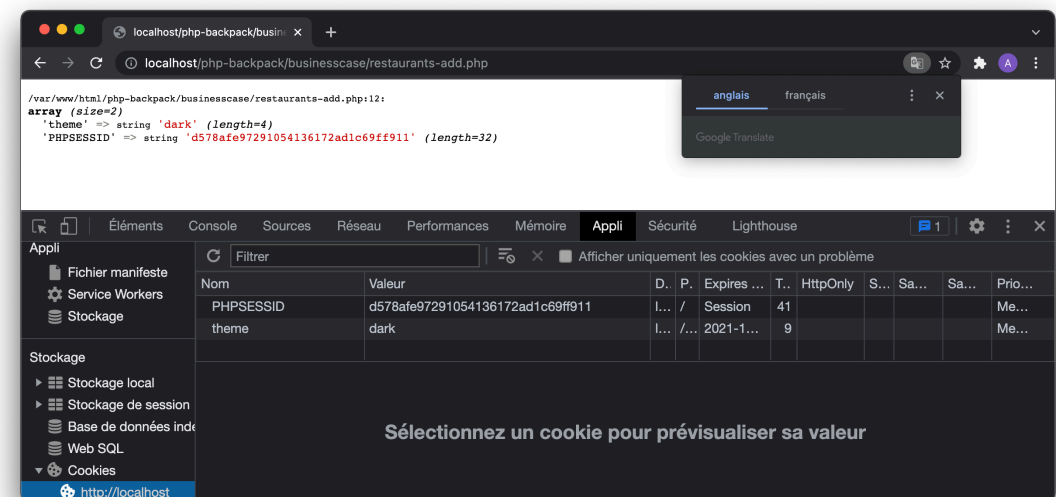
```
<?php
setcookie("theme", "dark", time()+3600);
?>
```

Ici nous créons un cookie nommé 'theme'. Il prendra la valeur 'dark' et sera valable 3600 secondes (1 heure)

<https://www.php.net/manual/fr/function.setcookie.php>

Lire un cookie

Utilisation de la variable super globale \$_COOKIE



```
<?php
var_dump($_COOKIE);
die();
?>
```



Upload de fichier

1) Le HTML

Utilisation de l'attribut '**enctype**' avec la valeur '**multipart/form-data**'
Utilisation d'un input HTML type file

```
<form method="post" action="connect.php" enctype="multipart/form-data">  
  <label>Username : </label>  
  <input name="login" type="text" placeholder="Utilisateur"> <br>  
  <label>Password : </label>  
  <input name="password" type="password" placeholder="Mot de passe"> <br>  
  Avatar: <input type="file" name="avatar"> <br>  
  <input type="submit">  
</form>
```

← → ↻ ⓘ localhost

Username :

Password :

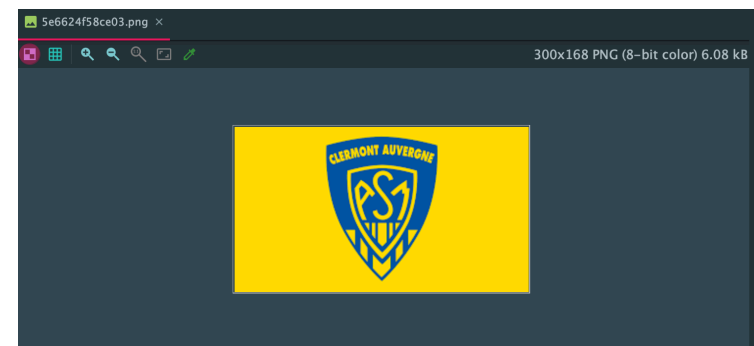
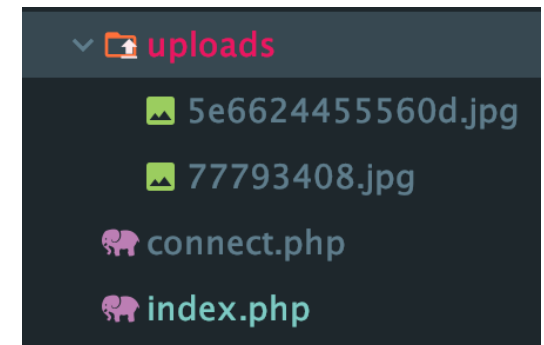
Avatar: Choisir un fichier Aucun fichier choisi



Upload de fichier

2) Le PHP

```
<?php
// Testons si le fichier a bien été envoyé et s'il n'y a pas d'erreur
if (isset($_FILES['avatar']) AND $_FILES['avatar']['error'] == 0) {
    // Testons si le fichier n'est pas trop gros
    if ($_FILES['avatar']['size'] <= 1000000)
    {
        // Testons si l'extension est autorisée
        $infosfichier = pathinfo($_FILES['avatar']['name']);
        $extension_upload = $infosfichier['type'];
        $extensions_autorisees = array('jpg', 'jpeg', 'gif', 'png');
        if (in_array($extension_upload, $extensions_autorisees))
        {
            // On peut valider le fichier et le stocker définitivement
            move_uploaded_file($_FILES['avatar']['tmp_name'], 'uploads/' . basename($_FILES['avatar']['name']));
            echo "L'envoi a bien été effectué !";
        } else {
            echo('J\'accepte que les images ...');
        }
    } else {
        echo('le fichier est trop lourd pour un petit serveur ... ');
    }
}
```





Upload de fichier

Utilisation de la variable super globale `$_FILE`

- Etape 1 : on vérifie que le fichier a bien été envoyé

- Etape 2 : On vérifie que le fichier n'est pas trop lourd pour éviter de surcharger le serveur

- Etape 3 : On vérifie que l'extension est bien celle que l'on attend en utilisant le mime type (Très important ! Sécurité !)

- Etape 4: La fonction `move_uploaded_file` déplace le fichier dans un dossier uploads précédemment créé

- Si toutes ces étapes sont respectés, notre fichier est désormais dans notre dossier upload grâce à la fonction PHP `move_uploaded_file` !

Si un fichier avec le même nom est ajouté, l'ancien fichier sera écrasé. La fonction php `uniqid` peut nous permettre de générer des noms uniques.

```
move_uploaded_file($_FILES['avatar']['tmp_name'], 'uploads/' .  
uniqid() . '.' . $extension_upload);
```




Echanges / Questions

