PHP

COOKIES (-> côté client)

version 1.3 : revision 2021

LES COOKIES

- La version littéraire est *témoin de connexion*
- Fichiers texte (alphanumérique) courts (< 4 Ko)
- Stocké sur le disque dur de l'internaute
- Permet de garder les informations sur un utilisateur
- Ce fichier peut être récupérée sur l'ordinateur et envoyée au serveur.

Utilisation

- suivi de session (garder les informations d'une page à une autre)
- Date de la dernière visite
- Contenu d'un panier d'achat
- Et tout ce que vous jugerez utile pour le confort de l'utilisateur

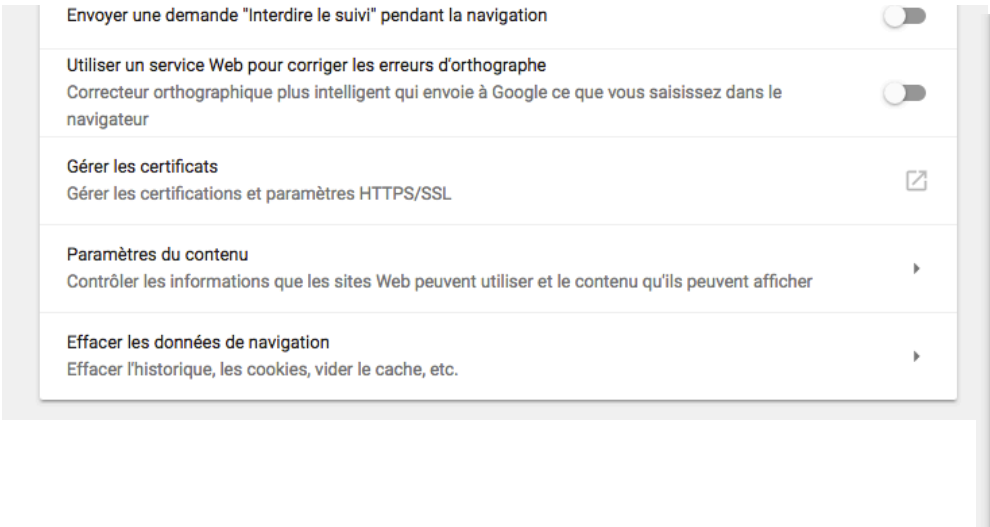
Utilisation

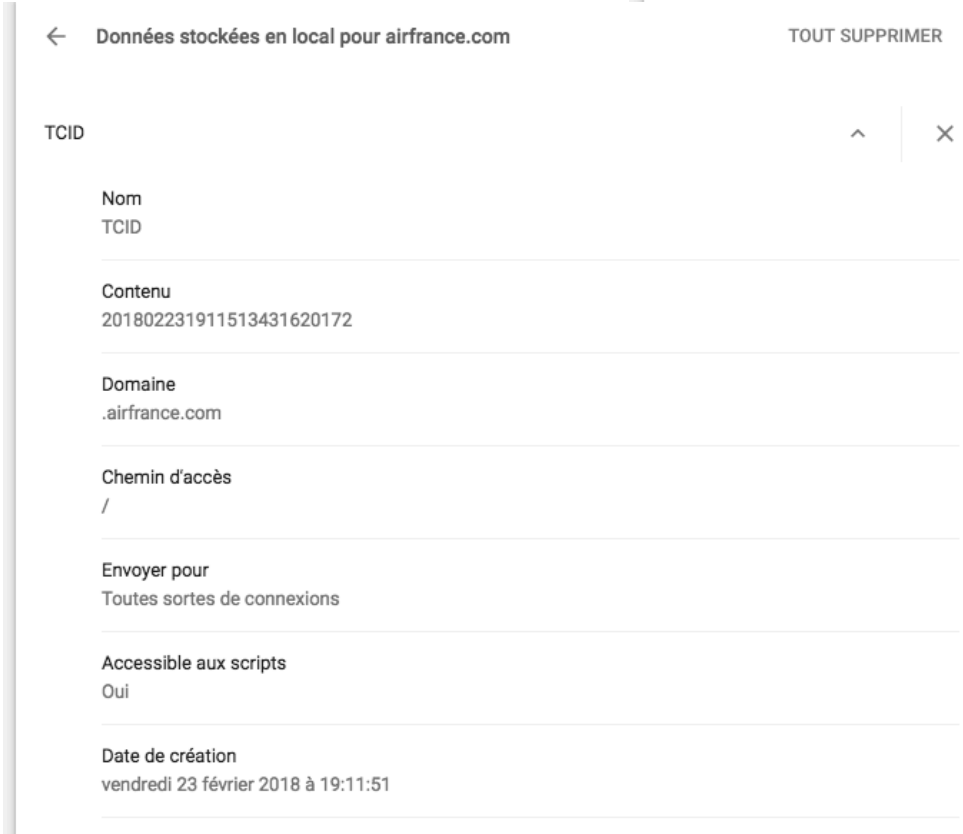
- Le but est d'assurer une transition agréable entre les pages et pour l'utilisateur
- Ne rien mettre d'important comme un pourcentage de réduction ! Car l'utilisateur peut modifier un cookie !
- Pour garantir la vie privée, seul le domaine qui a créé un cookie peut le relire.
- Mais une même page peut provenir de plusieurs domaines et chacun génère leur propres cookies. C'est ce qu'on appelle les cookies tierce et ils sont souvent issus de sociétés commerciales. On peut pour cette raison bloquer les cookies !
- Voir la vidéo

Législation

- Les législations diffèrent suivant les pays. En France la loi impose d'obtenir l'accord de l'utilisateur pour utiliser des cookies sur son ordinateur.
- Cela peut apparaitre sous la forme d'un bandeau avec du genre « J'accepte »
- Les cookies s'échangent durant le transfert des entêtes, avant l'envoi du HTML réel d'une page Web.

Le stockage des cookies

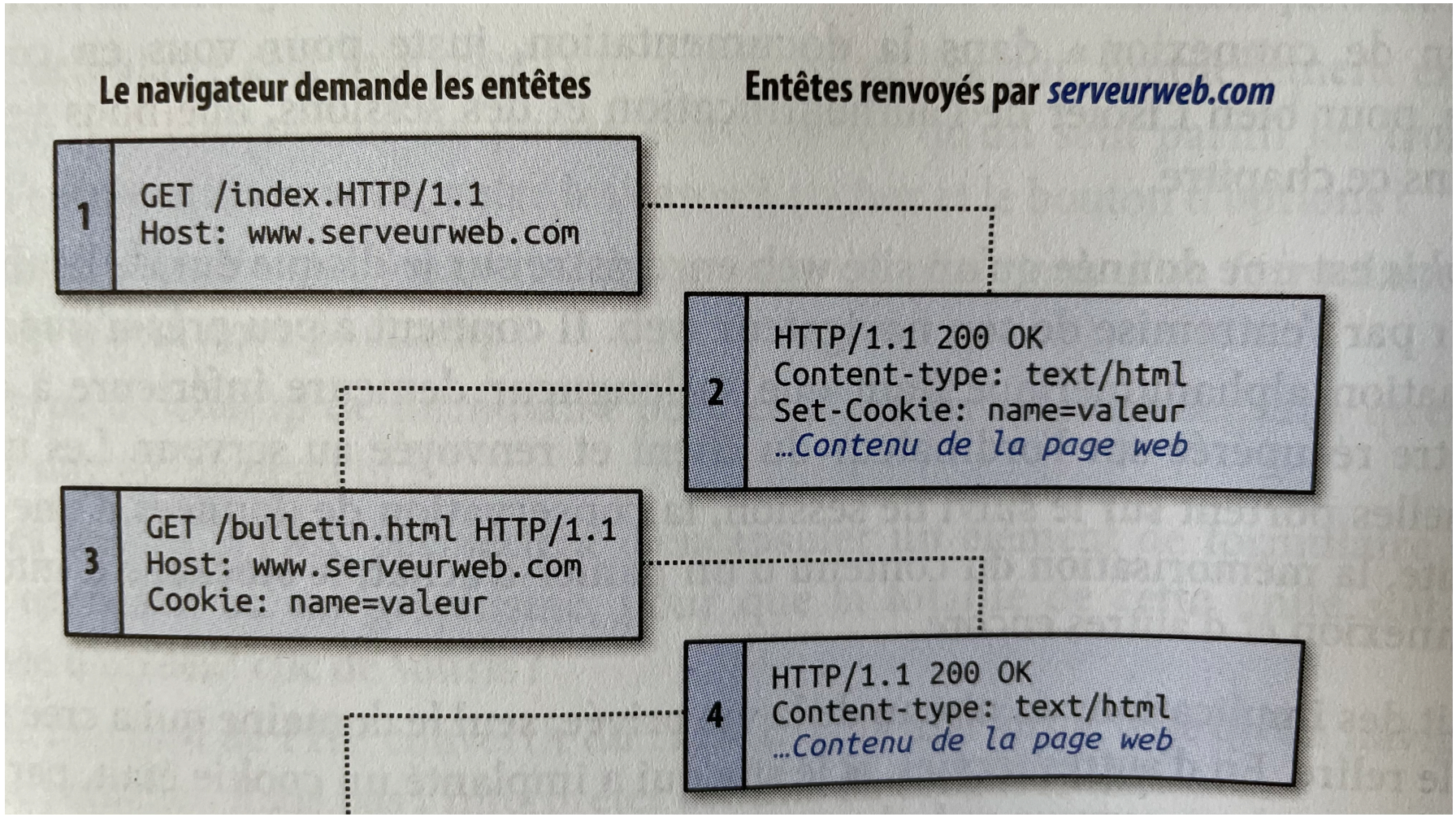




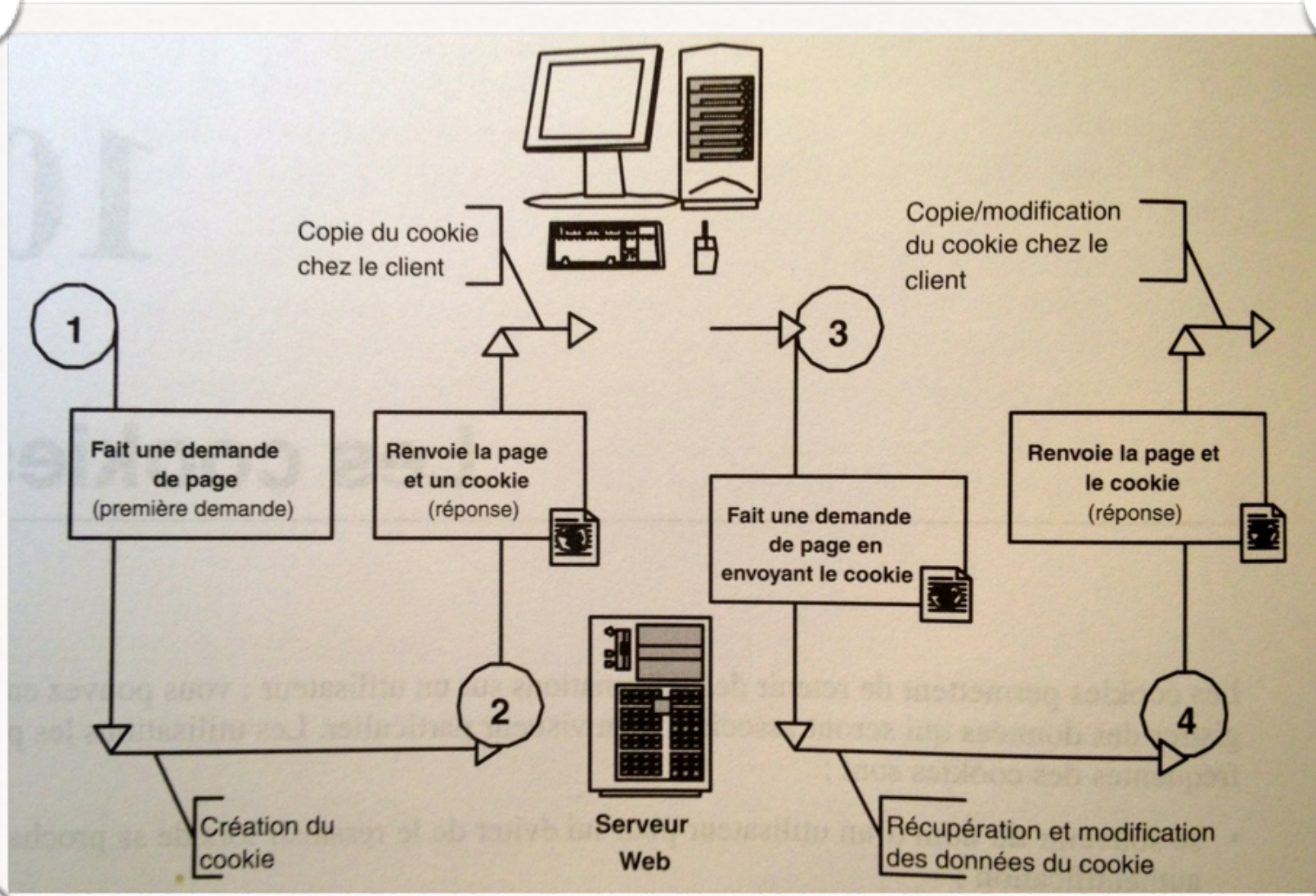
COOKIES

- Un cookie est un nom et une valeur.
- Instruction : setcookie()
- setcookie() définit un cookie qui sera envoyé avec le reste des en-têtes.
- Comme pour les autres en-têtes, les cookies doivent être envoyés avant toute autre sortie (c'est une restriction du protocole HTTP, pas de PHP). Cela vous impose d'appeler cette fonction avant toute balise <html> ou <head>.

Echange


Le navigateur demande les entêtes
Entêtes renvoyés par serveurweb.com
1
GET /index.HTTP/1.1
Host: www.serveurweb.com
2
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=valeur
…Contenu de la page web
3
GET /bulletin.html HTTP/1.1
Host: www.serveurweb.com
Cookie: name=valeur
4
HTTP/1.1 200 OK
Content-type: text/html
…Contenu de la page web

Principe


Copie du cookie
chez le client
Copie/modification
du cookie chez le
client
1
3
Fait une demande
de page
(première demande)
Renvoie la page
et un cookie
(réponse)
Fait une demande
de page en
envoyant le cookie
Renvoie la page et
le cookie
(réponse)
2
4
Création du
cookie
Serveur
Web
Récupération et modification
des données du cookie

COOKIES

- setcookie("btssio","slam",time()+3600);
- nom du cookie :“btssio”
- valeur “slam”
- durée de vie 3600 secondes soit 1 heure
  time() retourne l'heure courante, mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).
- Si pas de temps d’indiqué -> durée de vie de la session (fermeture du navigateur)

Code

```php
<?php
setcookie("btssio","slam",time()+3600);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Document sans nom</title>
</head>

<body>

<a href="cookie2.php">PAGE SUIVANTE</a>
<br/>
<a href="supcookie.php">SUPPR COOKIE</a>

</body>
</html>
```

Code

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Document sans nom</title>
</head>

<body>

<?php
if (isset($_COOKIE['btssio'])) $contenu = $_COOKIE['btssio'];
else $contenu = "Il n'existe pas !";
echo "Le contenu du cookie est  : ".$contenu;
?>

</body>
</html>
```

Code

```php
<?php
if(isset($_COOKIE['btssio']))
{
    setcookie("btssio");
    unset($_COOKIE['btssio']);
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Document sans nom</title>
</head>

<body>
<?php

if(isset($_COOKIE['btssio'])) echo "VOTRE LOG EST : ".$_COOKIE['log'];
else echo 'La suppresion a été effectuée !';

?>
</body>
```

SUPPRESSION d'un COOKIE

```php
<?php
if (isset($_COOKIE['btssio']))
{
    setcookie("btssio");
    unset($_COOKIE['btssio']);
}
echo " Contenu du cookie est ". $_COOKIE['btssio'];
?>
```

setcookie(«btssio») supprime le cookie (temps de vie à 0 seconde)

unset($_COOKIE[‘btssio’]) supprime ce cookie du tableau

SUPPRESSION d'un COOKIE

Une autre solution pour supprimer un cookie est de l'envoyer à nouveau avec une date d'expiration située dans le passé.

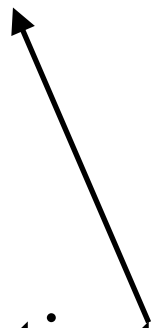
Attention tous les champs doivent être identique sauf l'horodatage.

Exemple si vous avez créer le cookie avec :
setcookie('pays','Canada', time() + 60 * 60 * 24 * 7)

pour le supprimer faire :
setcookie('pays','Canada', time() - 2592000)

Sécuriser avec httpOnly

- <?php setcookie('pseudo', 'SioBTS', time() + 365*24*3600, null, null, false, true); ?>



Activation de httpOnly

Ceci évite les risques de la faille XSS

Rappel : XSS (plus officiellement appelée Cross-Site Scripting) est une faille permettant l'injection de code HTML ou JavaScript dans des variables mal protégées.

Syntaxe complète

- setcookie(name, value, expire, path, domain, secure, httponly)
- path (falcultatif) permet d'indiquer où mettre le cookie
- domain (falcultatif) permet de restreindre à des sous domaine (ex: www/serveur.com/pub)
- secure : indique si le cookie doit passer par une connexion sécurisée de type https://
- httponly : indique si le cookie doit emprunter le protocole HTTP. Si la valeur est TRUE les langages comme javascript ne peuvent y accéder.