

Présentation générale des solutions

Nous avons vu, dans le chapitre précédent, comment stocker de petites quantités d'informations dans des cookies (4 Ko maximum).

Il existe deux solutions, dites de *Web Storage*, alternatives à l'utilisation des cookies :

- le stockage de session,
- le stockage local.

Comme pour les cookies, le stockage avec ces deux interfaces nommées respectivement `sessionStorage` et `localStorage`, se fait du côté navigateur. Il ne faut donc pas confondre ces deux techniques avec un stockage distant en base de données. Nous verrons d'ailleurs dans le cadre du prochain chapitre comment mettre en œuvre ce type de solution en utilisant le protocole SOAP (*Simple Object Access Protocol*), une librairie PHP dédiée NuSOAP et le système de gestion de bases de données MySQL.

Le *Web Storage*, encore appelé *DOM Storage*, est apparu avec HTML 5. Ce mode de stockage est désormais accepté par la grande majorité des navigateurs.

L'avantage principal du *Web Storage* par rapport aux cookies est la possibilité de mémoriser des informations en plus grand volume (5 Mo maximum). Par ailleurs, contrairement à ce qui se passe pour les cookies, les interfaces `sessionStorage` et `localStorage` ne requièrent pas de trafic HTTP sur le réseau. Rappelons que les cookies sont inscrits sur le disque dur de l'utilisateur (donc un transfert HTTP est nécessaire) alors que les informations gérées par `sessionStorage` et `localStorage` sont purement locales (sans transfert HTTP).

1. Stockage par `sessionStorage`

La persistance des données est limitée à la durée de la session utilisateur, les données ne sont conservées que pour la fenêtre active (ou l'onglet). Les données stockées ne sont récupérables que par l'intermédiaire du navigateur ayant servi à les stocker.

Il faut noter à la décharge du stockage par `sessionStorage` (c'est également vrai pour `localStorage`) que les données ne sont pas cryptées et ceci peut avoir un impact sur les données mémorisées sur un poste de travail à utilisateurs multiples.

À la fermeture du navigateur, les données sont effacées.

2. Stockage par `localStorage`

L'interface `localStorage` est assez proche de la précédente. La persistance est gérée différemment. A priori il n'y a pas de limite temporelle au stockage. Les données sont conservées au-delà de la fermeture de la fenêtre active (ou de l'onglet). Il est bien sûr également possible d'utiliser les données stockées dans plusieurs fenêtres (ou onglets) d'un même domaine. Comme pour le stockage par `sessionStorage`, le mécanisme requiert d'utiliser le même navigateur.

Bien évidemment, comme pour `sessionStorage`, il existe des méthodes dédiées pour supprimer à la demande les données sauvegardées.

Point commun aussi entre `sessionStorage` et `localStorage` (comme pour les cookies d'ailleurs), ils ne permettent de stocker que des chaînes de caractères. Pour mémoriser des tableaux ou des objets, il faudra avoir recours à JSON (*JavaScript Object Notation*). Rassurez-vous un exemple complet va vous être présenté dans le cadre

de ce chapitre.



Vous trouverez un excellent article sur JSON sur le site AlsaCréations à l'adresse suivante :
<http://www.alsacreations.com/article/lire/1161-json-ajax-jquery-jsonp-getjson.html>