

La sauvegarde des données

- Les navigateurs offrent un espace de stockage dans lequel les applications Frontend peuvent sauvegarder des données
 - `sessionStorage` : vidé quand le navigateur est fermé
 - `localStorage` : permanent sans délai d'expiration (au delà de la fermeture du navigateur)
- Usages possibles :
 - Sauvegarder les préférences et données utilisateurs de façon persistante alternativement aux solutions classiques de
 - Session (coté serveur) avec une validé limitée à la durée de la session
 - Cookies (coté client) avec une validité limitée à la date d'expiration du cookies
 - Disposer d'un cache de sauvegarde dans le cas d'un usage hors ligne (pas de connexion avec le serveur)
 - Mémoriser facilement de petites quantités de données de l'utilisateur durant sa navigation pour les faire persister ensuite sur le serveur
 - Créer des applications JavaScript sur navigateur sans lien avec le serveur



La sauvegarde des données

- Avantages

Stocker rapidement des données en cache sans faire intervenir le serveur
- Retrouver des données immédiatement à la reconnexion ou les mémoriser pour éviter la perte s'il y a déconnexion
- Augmenter la performance ressentie par l'utilisateur et faciliter le développement (moins de requêtes avec le serveur)
- Éviter l'utilisation des cookies et réduire le coût du trafic engendré
- Disposer d'un espace de stockage supérieur à la limite imposée par les cookies (4 Ko)



La sauvegarde des données

Informations utiles

Les données sont associées au nom de domaine et au protocole (http, https) de la page web → pas de partage entre plusieurs domaines et différents protocoles

Les données d'un même domaine sont partagées entre plusieurs onglets du même domaine

L'espace de stockage par domaine est limité à 5 Mo

Les données sont au format clé/valeur → dictionnaire


Les paires clé/valeur sont de type chaîne de caractères

Les utilisateurs peuvent lire, supprimer et modifier les données stockées dans le navigateur

Les données ne sont pas cryptées → [NE PAS STOCKER D'INFORMATIONS SENSIBLES](#)

INT475 - Développement Web
Front End pour les débutants

26


IMT Atlantique
Brest - Nantes - La Rochelle
www.imt-atlantique.fr


La sauvegarde des données

	PC					Mobile					serveur	
Navigateur	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	Deno
version	4	12	3.5	10.5	4	18	4	11	3.2	1.0	37	1.16

Compatibilité des navigateurs

INT475 - Développement Web
Front End pour les débutants

27


IMT Atlantique
Brest - Nantes - La Rochelle
www.imt-atlantique.fr

La sauvegarde des données ▶ API

- Les éléments `sessionStorage` et `localStorage` sont des objets de type `Storage` accessibles à partir de la classe `window` ou directement
 - `let storage = window.localStorage;`
 - `let storage = localStorage;`
- Ils partagent la même API offrant les fonctionnalités de :
 - Sauvegarde
 - `localStorage.setItem("key", "value");`
 - `localStorage["key"] = "value";`
 - `localStorage.key="value";`
 - Lecture
 - `let value=localStorage.getItem("key");`
 - `let value=localStorage["key"];`
 - `let value=localStorage.key;`
 - Suppression
 - `localStorage.removeItem("key");` // supprimer une donnée
 - `localStorage.clear();` // supprimer toutes les données
 - Recherche
 - `localStorage.length;` // renvoie le nombre de paires clé/valeur
 - `let keyName= localStorage.key(0);` // renvoie la clé 'key' de l'indice 0



La sauvegarde des données ▶ API

- Pour obtenir toutes les clés :
 - ```
for(let i = 0; i < localStorage.length; i++){
 localStorage.key(i);
}
```
- Pour sauvegarder des valeurs structurées ou complexes (tableau, objet, etc.) il faut utiliser le format JSON (JavaScript Objet Notation) et procéder à des opérations de sérialisation et dé-sérialisation :
  - Exemple de tableau
    - `let obj = [1,2,3];`
  - Exemple d'objet
    - ```
let obj = {  
    prenom : "Pierre",  
    nom : "Durand"  
}
```
 - Sérialisation (transformation d'un objet en chaîne de caractères) avant la sauvegarde
 - `localStorage.setItem("key", JSON.stringify(obj));`
 - Dé-sérialisation (transformation d'une chaîne de caractères en objet) après la lecture
 - `let obj = JSON.parse(localStorage.getItem("key"));`



La sauvegarde des données ► l'événement StorageEvent

- L'API dispose d'un événement permettant à plusieurs pages (onglets) du même domaine de recevoir une notification quand une page (onglet) change des données du storage.
- Usage → Permettre à toutes les autres pages du domaine qui utilisent le stockage de se synchroniser avec tous les changements faits.
- Remarque → l'événement n'est pas propagé à la page qui a provoqué le changement.
- L'événement contient :
 - La clé,
 - L'ancienne valeur,
 - La nouvelle valeur,
 - L'URL de la page à l'origine de l'événement,
 - Le contenu du storage
- Exemple :



La sauvegarde des données ► l'événement StorageEvent

```
<article>
  <header><h1>Log Storage</h1></header>
  <div>Key: <span class="my-key"></span></div>
  <div>old: <span class="my-old"></span></div>
  <div>new: <span class="my-new"></span></div>
  <div>url: <span class="my-url"></span></div>
  <div>storage: <span class="my-storage"></span>
</div>
</article>

<script>
window.addEventListener('storage', function(e) {
  document.querySelector('.my-key').textContent = e.key;
  document.querySelector('.my-old').textContent = e.oldValue;
  document.querySelector('.my-new').textContent = e.newValue;
  document.querySelector('.my-url').textContent = e.url;
  document.querySelector('.my-storage').textContent =
e.storageArea;
});
</script>
```



Reprendre le code de l'exercice d'un annuaire de personnes et intégrer le code nécessaire pour

- 1. Sauvegarder l'annuaire dans le localStorage du navigateur quand l'utilisateur le sollicite via un bouton de sauvegarde
- 2. Charger l'annuaire du localStorage à l'ouverture de la page
- 3. Supprimer toutes les données de l'annuaire du localStorage via un bouton reset après confirmation explicite de l'utilisateur

ACTIVITÉ PRATIQUE