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

INT475 - Développement Wel Front End pour les débutant

24



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

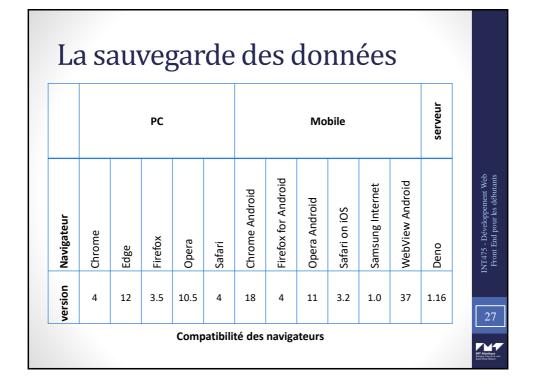
Disposer d'un espace de stockage supérieur à la limite imposée par les cookies (4 Ko)

NT475 - Développement V Front End pour les débuts

25







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;
- let value=localStorage
- 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



28



La sauvegarde des données > API

• Pour obtenir toutes les clés :

- 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 chaine de caractères) avant la sauvegarde

- localStorage.setItem("key", JSON.stringify(obj))
- Dé-sérialisation (transformation d'une chaine de caractères en objet) après la lecture
 - let obj = JSON.parse(localStorage.getItem("key"));

INT475 - Développement W Front Fnd nour les défuites

29



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 strorage
- Exemple :

NT475 - Développement We Front End pour les débutant

30



La sauvegarde des données l'événement StorageEvent

