

06

Données locales



L'enregistrement de données locales

LocalStorage et SessionStorage sont des fonctionnalités de stockage web offertes par les navigateurs Web modernes pour stocker des données côté client de manière persistante (d'une page à l'autre). Ces données restent disponibles même après la fermeture du navigateur et peuvent être utilisées pour stocker des préférences utilisateur, des paramètres de l'application, des informations de session, etc.

Ces données ne sont pas protégées, donc elles ne devrait jamais contenir de données sensibles. LocalStorage et sessionStorage peuvent, entre autres, garder de l'information en mémoire lors d'un passage d'une page à une autre ou permettre à un utilisateur de rester connecté à une application en gardant un jeton généré par le backend de l'application qui sera vérifié à l'ouverture de chacunes des pages, conserver les préférences sur le mode sombreclair de l'utilisateur, etc.

LocalStorage, SessionStorage

LocalStorage et SessionStorage sont deux mécanismes de stockage côté client offerts par les navigateurs web modernes, mais ils ont des différences importantes en termes de durée de vie des données et de portée.

LocalStorage

Les données stockées avec localStorage ont une durée de vie persistante et restent enregistrées même après la fermeture du navigateur ou le redémarrage de l'ordinateur. Elles restent disponibles tant que l'utilisateur ne les supprime pas manuellement, que le cache n'est pas vidé ou que votre application ne les efface pas par programmation.

Les données stockées dans localStorage sont partagées entre toutes les fenêtres et les onglets du même domaine. Cela signifie que les données stockées dans une fenêtre seront également disponibles dans une autre fenêtre ou un autre onglet du même site.

Il est couramment utilisé pour stocker des données permanentes, telles que des préférences utilisateur, des informations de connexion persistantes et des données d'application qui doivent être conservées d'une session à l'autre.

SessionStorage

Les données stockées avec sessionStorage ont une durée de vie limitée à la session en cours du navigateur. Elles seront supprimées lorsque l'utilisateur ferme l'onglet ou la fenêtre du navigateur. Chaque nouvelle session (nouvel onglet ou fenêtre) crée un nouveau sessionStorage.

Les données stockées dans sessionStorage sont spécifiques à la fenêtre ou à l'onglet en cours. Elles ne sont pas partagées entre différentes fenêtres ou onglets du même domaine.

Il est souvent utilisé pour stocker temporairement des données spécifiques à une session, comme des états de navigation, des données de formulaire et d'autres données qui ne sont pertinentes que pendant la durée d'une session de navigation.

Visualiser les données conservées

L'onglet "Stockage" ou "Application" dans les outils de développement de navigateur est une fonctionnalité qui permet de visualiser, modifier et gérer les données stockées localement, notamment dans localStorage et sessionStorage.

Rendez-vous sur la page que vous voulez inspecter.

Ouvrez l'onglet Stockage ou Application et dépliez la section stockage de session ou stockage locale.

Si des données ont été enregistrées, elles seront disponibles sous forme clé/valeur". Ex: ici nous avons un jeton permettant de resté connecté à une application.

The screenshot shows the developer tools interface with the "Stockage" tab selected. The sidebar lists "Cookies", "Indexed DB", and two expanded sections: "Stockage cache" and "Stockage de session". Under "Stockage de session", there is an item for the domain "https://statistiques.technoscience.ca". Under "Stockage local", there is also an item for the same domain. A blue bar highlights the "Stockage local" section. Below the sidebar, a table displays stored data with columns "Key" and "Value". One row is highlighted with a red box, showing the key "jwtToken" and the value "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVi...".

Key	Value
jwtToken	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVi...

Stocker , récupérer, supprimer des données

Pour stocker des données dans localStorage, vous pouvez utiliser la méthode `setItem()` avec une clé (nom) et une valeur (contenu). La clé et la valeur sont des chaînes de caractères de votre choix.

Pour récupérer des données depuis localStorage, vous pouvez utiliser la méthode `getItem()` en fournissant la clé.

Pour supprimer une entrée spécifique de localStorage, utilisez la méthode `removeItem()`.

Pour effacer complètement toutes les données stockées dans localStorage, utilisez la méthode `clear()`.

Les mêmes méthodes sont disponibles pour sessionStorage.

```
// Ajouter des données
localStorage.setItem("nom", "Maxime");
localStorage.setItem("age", "30");

// Récupérer des données
const nom = localStorage.getItem("nom");
const age = localStorage.getItem("age");

// Supprimer une entrée
localStorage.removeItem("nom");

// Efface toutes les données
localStorage.clear();

// Les mêmes méthodes sont disponibles
// pour sessionStorage.
sessionStorage.setItem("nom", "Maxime");
sessionStorage.setItem("age", "30");

// etc.
```

Limites

LocalStorage a une limite de stockage d'environ 5 à 10 Mo par domaine, ce qui signifie que vous ne devriez pas stocker de grandes quantités de données.

LocalStorage stocke les données sous forme de chaînes de caractères, vous devrez donc les convertir en objets JSON pour stocker des structures de données plus complexes

Convertir un objet en chaîne

Pour convertir un objet JSON en une chaîne JSON en JavaScript, vous pouvez utiliser la méthode `JSON.stringify()`. Cette méthode prend en entrée un objet JavaScript et renvoie une chaîne de caractères JSON correspondante.

Reconvertir une chaîne en objet

Pour reconvertir une chaîne JSON en un objet JavaScript, vous pouvez utiliser la méthode `JSON.parse()`. Cette méthode prend en entrée une chaîne JSON valide et renvoie un objet JavaScript correspondant.

```
const objet = { nom: "John", age: 30, profession: "Développeur" };
const chaineJSON = JSON.stringify(objet); // Maintenant une chaîne

localStorage.setItem("user", chaineJSON);

// Si on récupère un objet convertis en chaîne
const chaineUser = localStorage.getItem("user");

const objetUser = JSON.parse(chaineUser); // Maintenant un objet

// ** Attention, JSON prends des majuscules ici **
```

Sécurité

Les données stockées dans localStorage sont accessibles depuis le code JavaScript, donc évitez de stocker des informations sensibles comme les mots de passe.

localStorage est propre au domaine, ce qui signifie que les données ne sont accessibles que pour le domaine spécifique qui les a stockées. Il n'est pas possible de les consulter depuis un autre site.

Créer une classe pour sauvegarder les données

Maintenant que vous connaissez les classes, vous pourriez créer une classe disponible partout grâce aux fonctions statiques qui automatise le processus d'enregistrement local ou session.

Dans l'exemple droit, la classe ne comporte que les fonctions pour localStorage mais rien ne vous empêche d'ajouter des fonctions pour se connecter à une base de données, utiliser sessionStorage, etc.

```
//Dans le code principal
import GestionnaireDonnees from "./classes/GestionnaireDonnees.js";

GestionnaireDonnees.enregistrerDonneesLocales("test", "test2");
const valeur = GestionnaireDonnees.recupererDonneesLocales("test");
```

```
export default class GestionnaireDonnees {
    // fonction statique disponible partout
    // sans avoir à faire new

    //LocalStorage
    static enregistrerDonneesLocales(cle, valeur) {
        const cleChaine = cle.toString();
        const chaineJSON = JSON.stringify(valeur);
        localStorage.setItem(cleChaine, chaineJSON);
    }

    static recupererDonneesLocales(cle) {
        const cleChaine = cle.toString();
        const chaineJSON = localStorage.getItem(cleChaine);

        return JSON.parse(chaineJSON);
    }

    static supprimerDonneesLocales(cle) {
        const cleChaine = cle.toString();
        localStorage.removeItem(cleChaine);
    }

    static supprimerToutesDonneesLocales() {
        localStorage.clear();
    }
}
```

Références

LocalStorage - MDN

<https://developer.mozilla.org/fr/docs/Web/API/Window/localStorage>

SessionStorage - MDN

<https://developer.mozilla.org/fr/docs/Web/API/Window/sessionStorage>

Dark Mode with JS et LocalStorage

<https://javascript.plainenglish.io/build-a-dark-mode-toggle-with-javascript-and-localstorage-8022b492fb9e>

Cookie Apocalypse

<https://www.adviso.ca/blog/tech/apocalypse-des-cookies/>