



TD 7

# **STORAGE**



## OBJECTIFS

Découvrir le stockage d'informations dans le navigateur et faire un rappel de notions déjà abordées : fetch et MVC.

## STOCKAGE D'INFORMATION

### EXPLICATIONS

Les navigateurs proposent une API permettant de stocker des informations dans le navigateur sous la forme de paires [clé:valeur] :

```
//Enregistrer une donnée :  
localStorage.setItem("clé", "valeur");  
  
//Lire une information (si la clé n'existe pas, valeur vaudra null):  
const valeur = localStorage.getItem("clé");
```

La clé comme la valeur doivent toutes deux être des chaînes de caractères.

### PREPARATION

- Réalisez une page HTML simple comprenant un champ texte et un bouton.

### ENREGISTREMENT

- Associez un script JavaScript à votre page web.
- A partir des éléments vus précédemment, faites en sorte que le contenu du champ texte soit enregistré dans le LocalStorage lorsque l'utilisateur clique sur le bouton.
- Testez le bon fonctionnement en consultant la section « Application » ou « Stockage » des outils de développement de votre navigateur.

### CHARGEMENT

- A partir des éléments vus précédemment, faites en sorte que la valeur enregistrée dans le LocalStorage soit affichée dans le champ de saisie une fois le chargement de la page terminé.

Attention : il est possible que la clé n'existe pas. Pensez à prendre en compte ce cas.

- Actualisez la page pour tester le bon fonctionnement.
- Faites un nouveau test en chargeant votre page dans un nouvel onglet du navigateur.
- Faites un dernier test en fermant complètement votre navigateur (toutes les fenêtres) puis en le démarrant de nouveau. Chargez votre page.

## LOCALSTORAGE VS SESSIONSTORAGE

Les navigateurs proposent une seconde API pour le stockage de données : sessionStorage. Elle fonctionne de la même façon que localStorage :

```
//Enregistrer une donnée :  
sessionStorage.setItem("clé", "valeur");  
  
//Lire une information (si la clé n'existe pas, valeur vaudra null):
```



```
const valeur = sessionStorage.getItem("clé");
```

- Remplacez les appels à localStorage dans votre code par des appels à sessionStorage.
- Effectuez de nouveaux tests précédents : actualisation de la page, ouverture dans un nouvel onglet, fermeture complète du navigateur.
- Que pouvez-vous en déduire ?

## PROJET STARGATE

### PREPARATION

- Téléchargez et décompressez l'archive « td7-stargate.zip ».
- Ouvrez le dossier de ce projet dans visual studio code.
- Sélectionnez le fichier « index.html » situé à la racine et cliquez sur « Go Live ».

### EXPLORATION

#### RAPPELS

La fonction « fetch » permet de récupérer des ressources depuis un serveur sans avoir à recharger intégralement la page :

```
const response = await fetch(url, options);  
const textData = await response.text();
```

Les options données en paramètres de fetch permettent de :

- Définir la méthode http utilisée pour la requête

```
const options = {  
  method: "GET"  
};
```

- Transmettre des données

```
const formData = new FormData();  
formData.append("clé", "valeur");  
  
const options = {  
  body: formData  
};
```

### MISE EN PRATIQUE

- Etudiez le code du fichier « /controllers/controller.js »
- La méthode « engage » est appelée lorsque 7 symboles ont été saisis. Faites une requête POST auprès de l'url <http://51.83.35.197/api-stargate.php> en transmettant la séquence saisie sous la forme d'un tableau JSON associé à la clé « sequence » (JSON.stringify ?).  
L'API vous retournera le nom de la planète correspondant à la séquence fournie, ou une chaîne vide si aucune planète n'y est associée.  
La propriété « connected » du contrôleur doit passer à « true » si une planète est trouvée.



- Testez le bon fonctionnement.

## **HISTORIQUE**

### ENREGISTREMENT

- A chaque fois qu'une séquence correspond à une planète, ajoutez-la à l'historique du contrôleur.
- Puis enregistrez l'historique dans le LocalStorage.
- Testez et vérifiez avec les outils de développement que les données sont bien enregistrées.

### CHARGEMENT

- Chargez le contenu de l'historique mémorisé dans le LocalStorage au démarrage du contrôleur (JSON.parse ?).
- Testez le bon fonctionnement en réalisant un affichage console des données chargées.

### VUE

- Créez une nouvelle vue dédiée à la gestion de l'historique.
- Elle affichera les séquences mémorisées dans la section « history » de la page HTML
- Cliquez sur une séquence de l'historique déclenchera la saisie automatique de la séquence pour activer la porte.
- Testez en fermant puis relançant le navigateur.