

Les API de HTML5

Michel Gagnon
École Polytechnique de Montréal



Stockage local

- Deux possibilités:
 - L'information disparaît après la fin de la session, c'est-à-dire lorsque l'onglet contenant la page est fermé, ou lorsque le navigateur est fermé (objet **localStorage**)
 - L'information est persistante (objet **sessionStorage**)
- Dans les deux cas, on stocke des paires *attribut/valeur*
- L'espace de stockage est associé au domaine du site web (deux pages du même domaine peuvent accéder à la même information du stockage local)

Stockage local – exemple simple

```
<!DOCTYPE html>
<html lang="en">
<head>...</head>
<body>
  <p> Cliquez <a href="/exemples/stockageLocal/autrepape.html">ici</a> pour
    charger une page qui partage le stockage local</p>
  <fieldset>
    <legend>Stockage local</legend>
    <label for="localData"> Ceci sera persistant:</label>
    <input id="localData"><br>
    <label for="sessionData"> Ceci sera conservé durant la session:</label>
    <input id="sessionData">
  </fieldset>
  <div>
    <button onclick="saveData()">Sauvegarder</button>
    <button onclick="loadData()">Récupérer</button>
  </div>
</body>
</html>
```

Stockage local – exemple simple

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  ...
  <script>
    function saveData() {
      localStorage.setItem("localData", $("#localData").val());
      sessionStorage.setItem("sessionData", $("#sessionData").val());
    }
    function loadData() {
      if (localData != null) {
        $("#localData").val(localStorage.getItem("localData"));
      }
      if (sessionData != null) {
        $("#sessionData").val(sessionStorage.getItem("sessionData"));
      }
    }
  </script>
</head>
<body>...</body>
</html>
```

Stockage local – exemple simple

Cliquez [ici](#) pour charger une page qui partage le stockage local

STOCKAGE LOCAL

Ceci sera
persistant:

Ceci sera conservé
durant la session:

Sauvegarder

Récupérer

<http://michelgagnon.herokuapp.com/exemples/stockageLocal/>

Adapté de M. McDonald, *HTML5, The Missing Manual*, O'Reilly, 2014

AJAX

- Principe: communiquer avec le serveur pour obtenir des données qui seront utilisées (en général) pour modifier le document sans recharger la page
- Deux manières courantes de l'implémenter:
 - Utilisation de l'objet **XMLHttpRequest**
 - Utilisation d'un élément **<script>**
- En général, le serveur envoie le contenu demandé dans un des formats suivants: texte, HTML, XML, JSON
- Technique asynchrone

XMLHttpRequest

1. On crée une instance de XMLHttpRequest
2. On attache un gestionnaire à l'événement **readystatechange**
3. On ouvre la connexion (méthode **open()**)
4. On envoie la requête (méthode **send()**)
5. Dans le gestionnaire, on teste la valeur de l'attribut **readyState**, et on traite les informations reçues lorsque la valeur de cet attribut est 4

XMLHttpRequest – Exemple GET

```
<script>
  const xhr = new window.XMLHttpRequest()
  xhr.open('GET', 'http://localhost:3000/comments')
  xhr.addEventListener('readystatechange', function () {
    if (xhr.readyState === window.XMLHttpRequest.DONE &&
    xhr.status === 200) {
      const comments = JSON.parse(xhr.responseText)
      // FAIRE QUELQUE CHOSE AVEC LA VARIABLE
    }
  })
</script>
```


XMLHttpRequest – Exemple POST

```
<script>
  const xhr = new window.XMLHttpRequest()
  xhr.open('POST', 'http://localhost:3000/comments')
  xhr.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded')
  xhr.send(`auteur=${author}&commentaire=${comment}`)
  xhr.addEventListener('readystatechange', function () {
    if (xhr.readyState === window.XMLHttpRequest.DONE &&
xhr.status === 200) {
      const comment = JSON.parse(xhr.responseText)
      // FAIRE QUELQUE CHOSE AVEC LE NOUVEAU COMMENTAIRE
    }
  })
</script>
```

AJAX avec jQuery

- Méthode **load()**:
 - On lui passe une URL
 - Le contenu envoyé par le serveur remplacera celui de l'élément sur lequel la méthode est exécutée
 - Exemple:
`$('#stats').load('rapport.html')`
- Arguments supplémentaires possibles:
 - Un ensemble de données à ajouter à la requête
 - Une fonction qui sera exécutée lorsque le traitement de la requête aura été complété

AJAX avec jQuery

- Fonction utilitaire **\$.getScript()**:
 - Télécharge et exécute un code Javascript
 - Pas régi par la contrainte de la même origine
- Fonction utilitaire **\$.getJSON()**:
 - Fait une requête avec l'URL fournie
 - Parse la réponse JSON et transmet l'objet résultant à la fonction passée en paramètre
- Fonctions **\$.get()** et **\$.post()**:
 - Font une requête avec la méthode correspondante
- Fonction générique **\$.ajax()**:
 - On lui passe un ensemble de données qui définissent le genre d'interaction désiré

AJAX avec jQuery - Exemple

```
<script>
$.get('http://localhost:3000/comments', function(comments) {
  // FAIRE QUELQUE CHOSE AVEC LA VARIABLE
})

$.post('http://localhost:3000/comments
http://localhost:3000/comments',
      { author, comment},
      function(comment) {
        // FAIRE QUELQUE CHOSE AVEC LA VARIABLE
      })
</script>
```