

TELECOM
ParisTech



Institut
Mines-Télécom

JAVASCRIPT et les applications web

APIs récentes ou en cours d'adoption

Télécom ParisTech

Jean-Claude Moissinac – Octobre 2014

Avec l'aide de Jean-Claude Dufourd

Et Thomas Bertrand

Mastère CPD





Javascripts, API récentes



Plan du cours

- **Tour d'horizon des API**
- **Développements HTML5**
 - Accès aux capteurs (accéléromètre, gyroscope et géolocalisation)
 - Détection online/offline
 - Web Storage (local et session storage)
 - Mode offline

TELECOM
ParisTech



Institut
Mines-Télécom

Tour d'horizon des API



Audio et vidéo

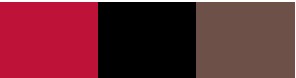
- **Permettre de lire des sons et des vidéos.**
- **Compléter les balises <audio> et <video>**
 - Possibilité de créer manuellement les commandes des lecteurs multimédia, on obtient un lecteur avec une charte graphique personnalisée.

Cf cours de Cyril Concolato

Géolocalisation

- **Déterminer la position de l'utilisateur :**
 - Par le GPS,
 - A défaut par la localisation IP moins précise (3G et fixe).

- **Interagir de façon simplifiée avec les API de cartographie (Google Maps) et possibilité de centrer la carte sur l'utilisateur.**
 - Ex: <http://html5demos.com/geo>



```
if (navigator.geolocation) {  
  navigator.geolocation.getCurrentPosition(success, fail);  
}
```

```
function success(position) {  
  alert('Latitude: '+ position.coords.latitude + ', Longitude:  
  '+ position.coords.longitude);  
}
```

Accès aux capteurs

- **Accéder aux capteurs présents sur les smartphones, tablettes et PC.**
 - Accéléromètre, Gyroscope
 - Micro, Caméra (<http://www.w3.org/TR/html-media-capture/>)
 - *(cf Cyril Concolato)*

- **→ Rapprochement fort avec les applications natives car on accède désormais à presque tout.**

Web storage

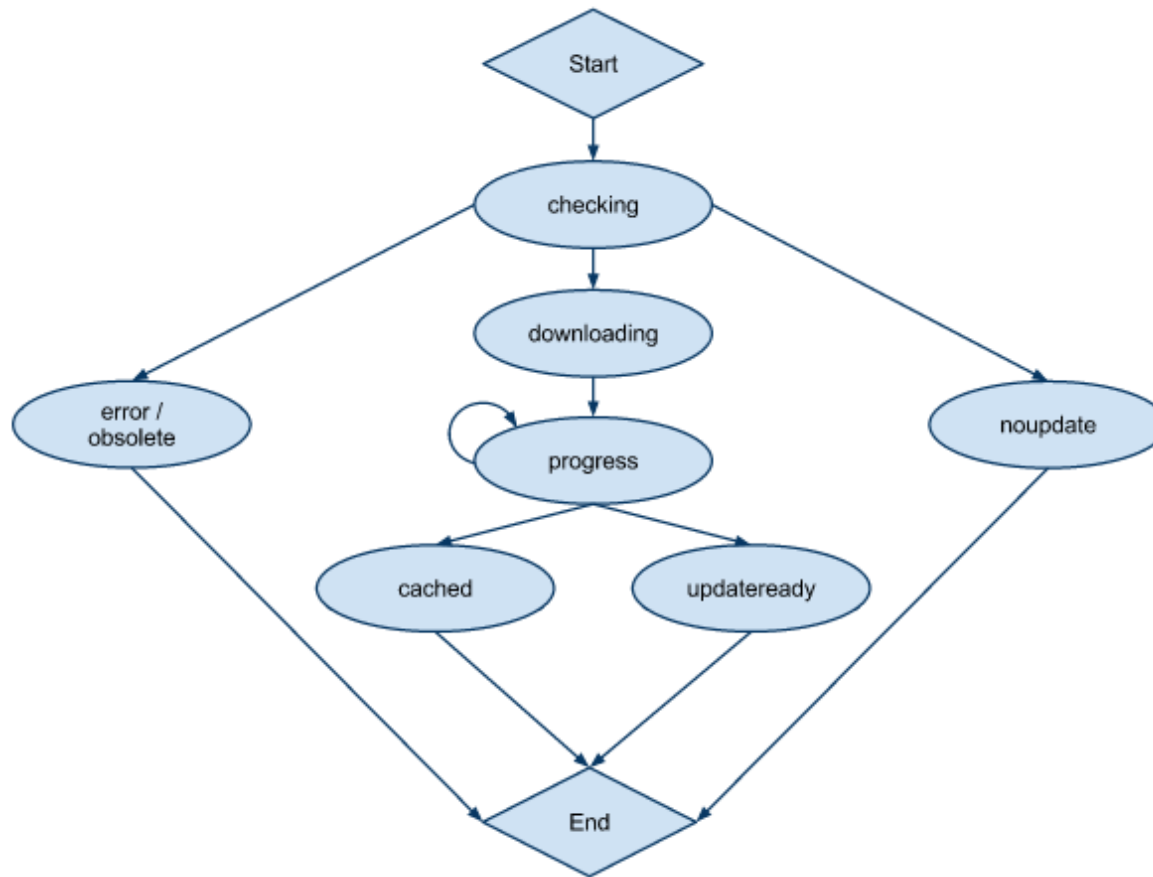
- **Stockage du côté du client**
 - => pas visible du serveur
- **Stocker des données dans une mémoire du navigateur (jusqu'à 5Mo extensibles).**
 - localStorage → stockage des données pérennes sous format textuel
 - sessionStorage → stockage des données le temps de la session sous format textuel
 - IndexedDB → stockage des données pérennes comme une base de données



Mode offline

- **Permettre de rendre un contenu web totalement accessible de manière déconnectée.**
- **Utiliser un ‘manifest’**
 - fichier dans lequel on référence tous les fichiers à sauvegarder dans une mémoire du navigateur : l'appCache
- **API importante pour créer des webApps.**

Mode offline – Fonctionnement de l'appCache



Source : : <http://blog.caplin.com/2011/02/14/taming-the-app-cache/>

Gestion d'historique

- **Permettre de contrôler l'historique de navigation du site.**
- **Eviter les erreurs dues aux aller-retour lors de la navigation notamment quand les pages utilisent AJAX pour recharger des blocs sans changer de page.**
 - On peut ainsi afficher l'ancien affichage des blocs sans pour autant quitter la page.

Gestion de fichiers

- **Offrir au JS la capacité d'interagir avec les fichiers.**
 - Analyse des fichiers (poids, dimensions, type...).
 - Préparation des fichiers en vue d'un upload sur le serveur.
 - Possibilité de créer des fichiers comme des vignettes d'images.



WebSockets

- **Permettre de simplifier et d'améliorer la technologie AJAX.**
- **Unifier les codes entre tous les navigateurs.**
- **Utiliser la méthode push :**
 - le serveur envoie les données
 - le navigateur est en attente pour les récupérer.

(nous y reviendrons)

WebWorkers et Messaging

- **Le JS fonctionne avec 1 seul processus.**
- **Les webWorkers sont des processus JS parallèles au processus principal qui vont pouvoir exécuter des traitements complexes et ainsi alléger le travail de ce dernier.**
- **L'API Messaging se charge des échanges entre les workers et le programme principal.**
- **Cela évite le blocage du reste de la page par un traitement lourd.**

Contenus éditables

- **Rendre tout élément HTML modifiable sans pour autant être un champ de formulaire.**
- **Possibilités**
 - Mettre en oeuvre des sortes de formulaires
 - créer un éditeur de texte pourvu des commandes traitement de texte classiques.
- **<http://www.w3.org/TR/html5/editing.html>**
 - Ex : <http://html5demos.com/contenteditable>



Plein écran

- **Activer un mode plein écran sur tous les éléments HTML souhaités.**
 - Très utile quand on crée ses propres commandes pour le lecteur vidéo.
 - Permet aussi de mettre au premier plan des blocs de pages.
 - (working draft)
 - <http://www.w3.org/TR/fullscreen/>

Drag'n'drop

- Définir des zones « draggables » et « droppables ».
- → Le JS s'adapte au système sur lequel il se trouve et est capable de gérer les surfaces tactiles qui étaient jusqu'alors très difficiles à utiliser pour le drag'n'drop.
 - Ex : <http://html5demos.com/drag>
 - <http://www.w3.org/TR/2011/WD-html5-20110525/dnd.html>



Dessin 2D et 3D

- **Créer des formes géométriques 2D ou 3D au sein de la nouvelle balise <canvas>.**
- **Pour la 3D, l'API utilise les spécifications WebGL auxquelles les navigateurs accèdent.**
- **Concurrence directe des jeux flash.**
 - Ex : playcanvas.com

TELECOM
ParisTech



Institut
Mines-Télécom

Développements HTML5



Sélecteurs CSS

- **Accéder au éléments HTML pour les modifier et interagir avec eux.**

```
//En HTML 4 seule la sélection par ID est possible
var element = document.getElementById('identifiant');

//En HTML 5 la sélection se fait par ID, classe ou balise
var element = document.querySelector('#identifiant');
var element = document.querySelector('.classe');
var element = document.querySelector('balise');

//Avec jQuery
var element = $('#identifiant');
var element = $('.classe');
var element = $('balise');
```

Accès aux capteurs

■ Accéléromètre

```
window.addEventListener("devicemotion", function(event) {  
  //event.acceleration et event.rotationRate  
}, true);
```

■ Gyroscope

```
window.addEventListener("deviceorientation",  
function(event) {  
  //event.alpha, event.beta and event.gamma  
}, true);
```

■ GPS ou autre positionnement

```
navigator.geolocation.getCurrentPosition(function(position)  
{  
  //position.coords.latitude et position.coords.longitude  
}, errorHandler);
```

Détection online/offline

- Détecter si le navigateur (et donc l'appareil) est connecté à internet.

```
Navigator.onLine;  
//Retourne True si le navigateur est en ligne
```

- Utile si l'on veut échanger avec un serveur en ligne.

Web Storage

■ localStorage

- Sauvegarde les données même si l'on ferme le site (format chaîne de caractères).

■ sessionStorage

- Sauvegarde les données de la session courante (format chaîne de caractères).

```
//Stockage de valeurs classiques
localStorage['clé'] = 'valeur';
var valeur = localStorage['clé'];
//Stockage de valeurs classiques
localStorage['clé'] = JSON.stringify(object);
var objet = JSON.parse(localStorage['clé']);
```


Mode offline

■ Enregistrer la webApp dans l'Application Cache (appCache) avec un manifest et tester son accessibilité en mode offline.

1. Modifier la balise <html> de ses pages

```
<html manifest="manifest.appcache">
```

2. Créer un fichier .htaccess et ajouter la ligne suivante pour indiquer que le .appcache correspond au manifest du mode offline

```
AddType text/cache-manifest .appcache
```

<http://www.w3.org/TR/2011/WD-html5-20110525/offline.html>

Mode offline

3. Créer le fichier manifest.appcache et ajouter tous les fichiers à mettre en cache un à un (seules les parties CACHE MANIFEST ET CACHE sont obligatoires).

```
CACHE MANIFEST
# Version 01/01/2012 - v1
# Lise des entrées à mettre en cache
CACHE:
index.html
styles/style.css
js/script.js
images/loading.gif

# Ressources nécessitant obligatoirement un mode online
NETWORK:
mail.php

# Ressources affichées à la place des éléments non chargés
# Ici offline.html s'affichera à la place de tous les fichiers html
FALLBACK:
*.html /offline.html
```

Pour aller plus loin

- **Working draft de la spécification HTML5**
 - W3C : <http://www.w3.org/TR/html5/>
 - WHATWG : <http://www.whatwg.org/specs/web-apps/current-work/multipage/>
- **Sites présentant le HTML5 et ses API**
 - <http://www.html5rocks.com>
 - <http://slides.html5rocks.com>
 - <http://www.html5-css3.fr/>
 - <http://bleedinghtml5.appspot.com/>
 - <http://html5demos.com/>
- **Comparatifs et tests de compatibilité des API HTML5 selon les navigateurs web**
 - <http://caniuse.com/>
 - <http://findmebyip.com/litmus/>
 - <http://html5test.com/>
- **Site présentant l'évolution du support du HTML5**
 - <http://html5readiness.com/>