Applications Web :
Compilation native ou Offline Apps

Programmation Web avancée et mobile – Mif38

Aurélien Tabard

Rich Internet Application (RIA)

- Définition (rappel) : application Web riche
- Utilise des fonctionnalités Web avancées pour rendre un service complexe
- Intérêts :
  - portabilité
  - environnements de développement Web classiques
- Inconvénients :
  - limitations techniques / de sécurité imposées par le navigateur
  - temps d’exécution

Application native (desktop ou mobile)

- Définition (rappel) : application installée sur l'OS de la machine
- Accès natif à toutes les fonctionnalités de la machine
- Intérêts :
  - APIs de développement complètes
  - Seules limitations : hardware / OS
  - temps d'exécution
- Inconvénients :
  - portabilité
  - environnements de développement parfois propriétaires

Entre les deux

- Adobe Air (Flash/Flex) principalement pour du jeu
- JavaFX peu utilisé
- Microsoft Silverlight deprecated
- Mozilla XULRunner peu soutenu par la fondation

La direction :

des applications Web accédant aux API natives

Aurélien Tabard - Université Claude Bernard Lyon 1

Les alternatives techniques pour le développement mobile

?

Les alternatives techniques pour le développement mobile

- Reponsive Web
  - HTML 5
- Natif mobile
  - iOS, Android, Windows, Tizen, Firefox OS…
- Cross-platform mobile
  - Xamarin Studio (C#), Adobe Air (Flex/Actionscript), Titanium (XML/Javascript)
- Web embarqué
  - Cordova

Les alternatives techniques pour le développement mobile

- **HTML 5**
  - Reponsive + device APIs
- Natif mobile
  - iOS, Android, Windows, Tizen, Firefox OS…
- Cross-platform mobile
  - Xamarin Studio (C#), Adobe Air (Flex/Actionscript), Titanium (XML/Javascript)
- Web embarqué
  - Cordova

Web applications

- Initiative du W3C, menée par le WebApp WG
- Différentes spécifications visant à standardiser la notion d'application Web
  - Web IDL
  - Web Components
  - Widgets (interfaces & packages)
  - DOM (Level 4, Events, shadow…)
  - Service Workers
  - …
- Spécifications / implémentations en cours

Gérer le offline : Application Cache

- Supporté par les navigateurs récents
- Fonctionne en parallèle du cache classique du navigateur
- Se base sur un manifest :

```
<!DOCTYPE html>
<html manifest="/cache.manifest">
<body>…</body>
</html>
```

- Un générateur pour de manifest pour les paresseux :

  https://appcache-generator.herokuapp.com/

Le manifest

```
CACHE MANIFEST
/clock.css
/clock.js
/clock-face.jpg
```

Commence toujours par cette ligne

Les fichiers à mettre en cache

Ne pas tout cacher

```
CACHE MANIFEST

NETWORK:
/tracking.js

CACHE:
/clock.css
/clock.js
/clock-face.jpg
```

Les fichiers à ne pas cacher

Fallback

Problème : on ne veut pas cacher tout un site mais seulement une partie (ex : les pages vues sur Wikipedia).

```
CACHE MANIFEST

FALLBACK:
/ fallback.html
/avatars/ avatars/v1.png
```

Si une requête échoue on affiche fallback.html

Sauf pour les requêtes sur /avatars/
ou on utilisera v1.png

Mettre à jour le manifest

1. Ne pas cacher le fichier manifest…

   -> Gérer le cache côté serveur

```
CACHE MANIFEST
# rev 43

clock.js
clock.css
```

Un numéro de version pour signaler un changement dans les fichiers cachés

Mettre à jour le manifest

2. la version cachée est chargée d'abord, il faut recharger pour mettre à jour la page.

```
window.applicationCache
  .addEventListener('updateready', function() {
   status.innerText = "Update ready - click to
update";
});
```

Un cache plus avancé : Service Worker

Objectif : Répondre aux nombreuses limites de App Cache

- Concept général : un proxy permettant de controller les requêtes http depuis venant des pages web.
- C’est un JavaScript Worker : pas d’accès au DOM, communication via passage de messages, mais accès à l’API IndexedDB
- Utilisation des promesses
- En cours de développement :
  - synchronisation en tache de fond, notifications…

HTML 5

Responsive Web + Device API + App Cache
permet du faire une grande partie du chemin.
À combiner avec du stockage local et des Service Workers.

Mais :

- pas de présence sur les stores
- pas d’accès à toutes les APIs
- problèmes possibles de performances

Les alternatives techniques pour le développement mobile

- Reponsive Web
  - HTML 5
- Natif mobile
  - iOS, Android, Windows, Tizen, Firefox OS…
- Cross-platform mobile
  - Xamarin Studio (C#), Adobe Air (Flex/Actionscript), Titanium (XML/Javascript)
- **Web embarqué**
  - **Cordova**

Cordova : détails pratiques

- Principe :
  - écrire une application Web et la faire compiler pour plusieurs OS mobiles
- OS pris en charge :
  - Android, iOS, BlackBerry, Windows Phone, Tizen, Firefox OS
- Permet d'accéder à certaines API natives
  - API JS spécifiques (pas toujours conformes aux specs du W3C)
  - Les SDKs sont nécessaires pour accéder aux API (donc les licences associées)
- Exécution en plein écran dans le navigateur natif
  - le coeur est toujours une WebView.
- PhoneGap (http://phonegap.com) est la surcouche de Cordova la plus populaire. Racheté en 2011 par Adobe.

Cordova : détails pratiques

- Mode CLI (pas d'IDE)
  - Compilation, émulation
  - Utilisation de npm pour gérer les plugins (console améliorée, accès aux capteurs…)
- Possibilité de faire tourner l'application
  - Sur un appareil connecté en USB (nécessite les drivers)
  - Sur un émulateur (Android Virtual Device) à créer avec le AVD manager

Run Chrome Apps on Mobile Using Apache Cordova

The toolchain for running Chrome Apps on mobile is in early developer preview. Feel free to give us your feedback using the Github issue tracker, our Chrome Apps developer forum, on Stack Overflow, or our G+ Developers page.

Overview

You can run your **Chrome Apps** on Android and iOS via a **toolchain** based on **Apache Cordova**, an open source mobile development framework for building mobile apps with native capabilities using HTML, CSS and JavaScript.

Apache Cordova wraps your application's web code with a native application shell and allows you to distribute your hybrid web app via Google Play and/or the Apple App Store. To use Apache Cordova with an existing Chrome App, you use the `cca` (**c** ordova **c** hrome **a** pp) command-line tool.