

Formation Mobile Hybride Day 5

Cordova, Ionic, Firebase

Agenda

-
1. Builder sur le téléphone

 2. Statut des Webviews aujourd'hui

 3. Initiation environnements de développement mobile

 4. Initiation tests unitaires front

 5. Separation of concern

Builder sur le téléphone

Environnement de développement Mobile



- **Seulement sous Mac**
 - Xcode + simulateur IOS
 - Debug avec Safari
-



- **Sous Windows/Mac/Linux**
 - Utilisation du SDK Android sans IDE
 - Debug avec chrome sur android>4.4
-

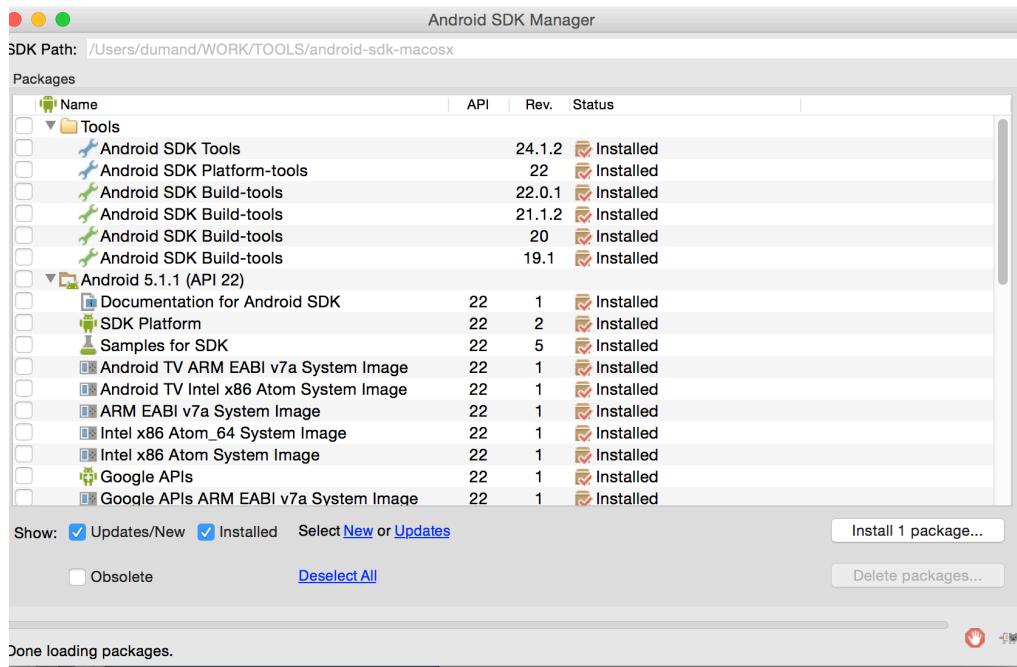


Windows Phone

- **Seulement sous Windows**
- Environnement Visual Studio
- Documentation cordova obsolète



Android: Installation du SDK



3. Configure the `ANDROID_HOME` environment variable based on the location of the Android SDK. Additionally, consider adding `ANDROID_HOME/tools`, and `ANDROID_HOME/platform-tools` to your PATH.

Mac OS X

```
export ANDROID_HOME=<installation location>/android-sdk-macosx
export PATH=${PATH}:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools
```

Linux

```
export ANDROID_HOME=<installation location>/android-sdk-linux
export PATH=${PATH}:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools
```

Windows

```
set ANDROID_HOME=C:\<installation location>\android-sdk-windows
set PATH=%PATH%;%ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools
```

Android: emulateur

Lancement de l'emulateur :

```
→ demo cordova emulate android
Running command: /Users/dumand/Desktop/demo/platforms/android/cordova/run --emulator
ANDROID_HOME=/Users/dumand/WORK/TOOLS/android-sdk-macosx
JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home

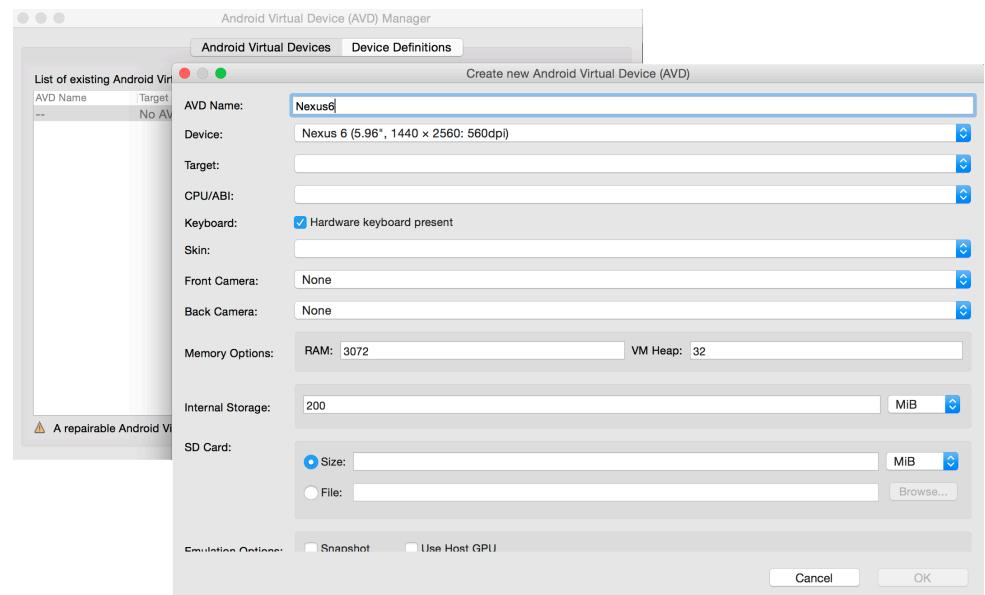
/Users/dumand/Desktop/demo/platforms/android/cordova/node_modules/q/q.js:126
    throw e;
^

ERROR : No emulator images (avds) found.
1. Download desired System Image by running: /Users/dumand/WORK/TOOLS/android-sdk-macosx/tools/android sdk
2. Create an AVD by running: /Users/dumand/WORK/TOOLS/android-sdk-macosx/tools/android avd
HINT: For a faster emulator, use an Intel System Image and install the HAXM device driver

Error: /Users/dumand/Desktop/demo/platforms/android/cordova/run: Command failed with exit code 8
at ChildProcess.whenDone (/usr/local/lib/node_modules/cordova/node_modules/cordova-lib/src/cordova/superspawn.js:131:23)
at ChildProcess.EventEmitter.emit (events.js:98:17)
at maybeClose (child_process.js:735:16)
at Process.ChildProcess._handle.onexit (child_process.js:802:5)
```

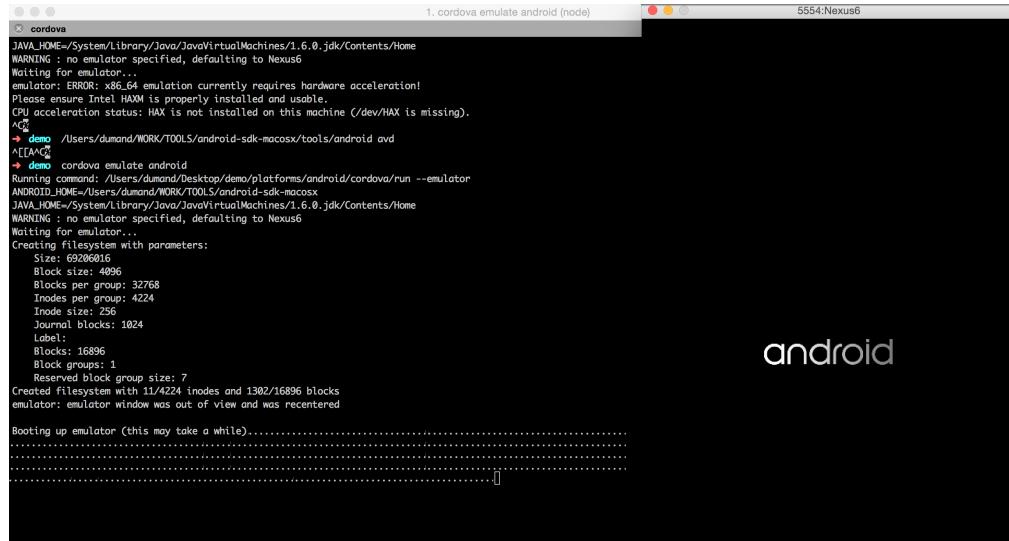
(no emulator found)

Création d'une machine virtuelle



Android: emulateur

« wait for it ... »



```
cordova
JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
WARNING : no emulator specified, defaulting to Nexus6
Waiting for emulator...
emulator: ERROR: x86_64 emulation currently requires hardware acceleration!
Please ensure Intel HAXM is properly installed and usable.
CPU acceleration status: HAX is not installed on this machine (/dev/HAX is missing).
A[dd] I[nsert] C[onsole]
+ demo /Users/dumand/WORK/TOOLS/android-sdk-macosx/tools/android avd
+ demo
+ demo cordova emulate android
Running command: /Users/dumand/Desktop/demo/platforms/android/cordova/run --emulator
ANDROID_HOME=/Users/dumand/WORK/TOOLS/android-sdk-macosx
JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
WARNING : no emulator specified, defaulting to Nexus6
Waiting for emulator...
Creating filesystem with parameters:
Size: 69206016
Block size: 4096
Blocks per group: 32768
Inodes per group: 4224
Inode size: 256
Journal blocks: 1024
Label:
Blocks: 16896
Block groups: 1
Reserved block group size: 7
Created filesystem with 11/4224 inodes and 1302/16896 blocks
emulator: emulator window was out of view and was recentered
Booting up emulator (this may take a while).....
.....
```

Genymotion, le sauveur !

oo
GENYMOTION

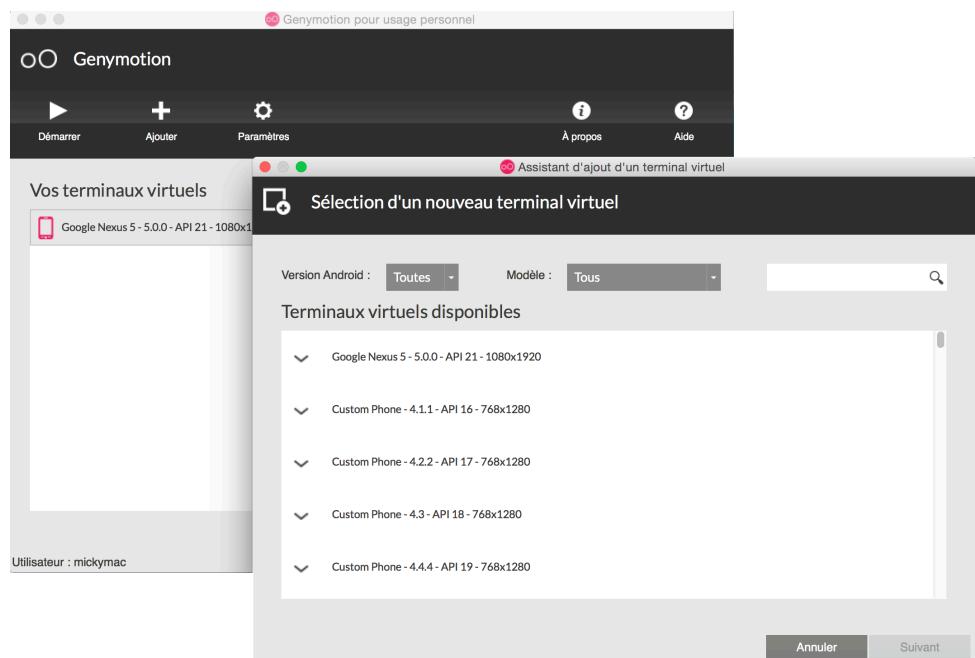
Genymotion

The screenshot shows the official website for Genymotion. At the top, there's a navigation bar with links for "About us", "Dev. zone", "GENYMOTION", "Product", "Store", and "Support". A user profile icon is also present. The main header features a pink background with white clouds. It includes a large monitor icon showing a character with a play button and the text "NO JOB?", followed by the tagline "A faster Android emulator". Below this, a paragraph explains that Genymotion is the next generation of the AndroVM open source project, trusted by 2,500,000 developers. A "Get Genymotion" button is visible. The page then transitions into a comparison table for the "Free" and "Business" editions.

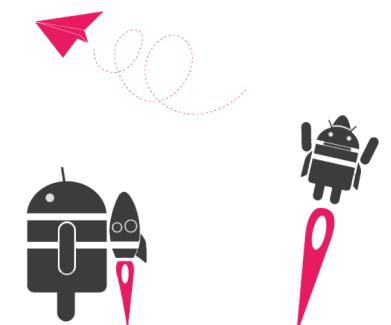
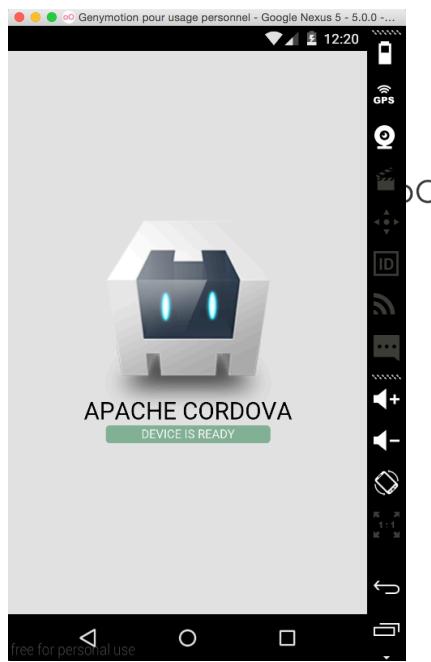
	Free	Business
Android 2.3, 4.1, 4.2, 4.3, 4.4, 5.0, 5.1	✓	✓
GPS	✓	✓
Camera	✓	✓
Multi-touch	✗	✓
Accelerometer	✗	✓
Java API	✗	✓
Screencast	✗	✓
Virtual device migration	✗	✓
Priority support (reply guaranteed within 1 working day)	✗	✓
License restriction	For personal use only	One license per user
	Download	Purchase

Genymotion

Créer un émulateur

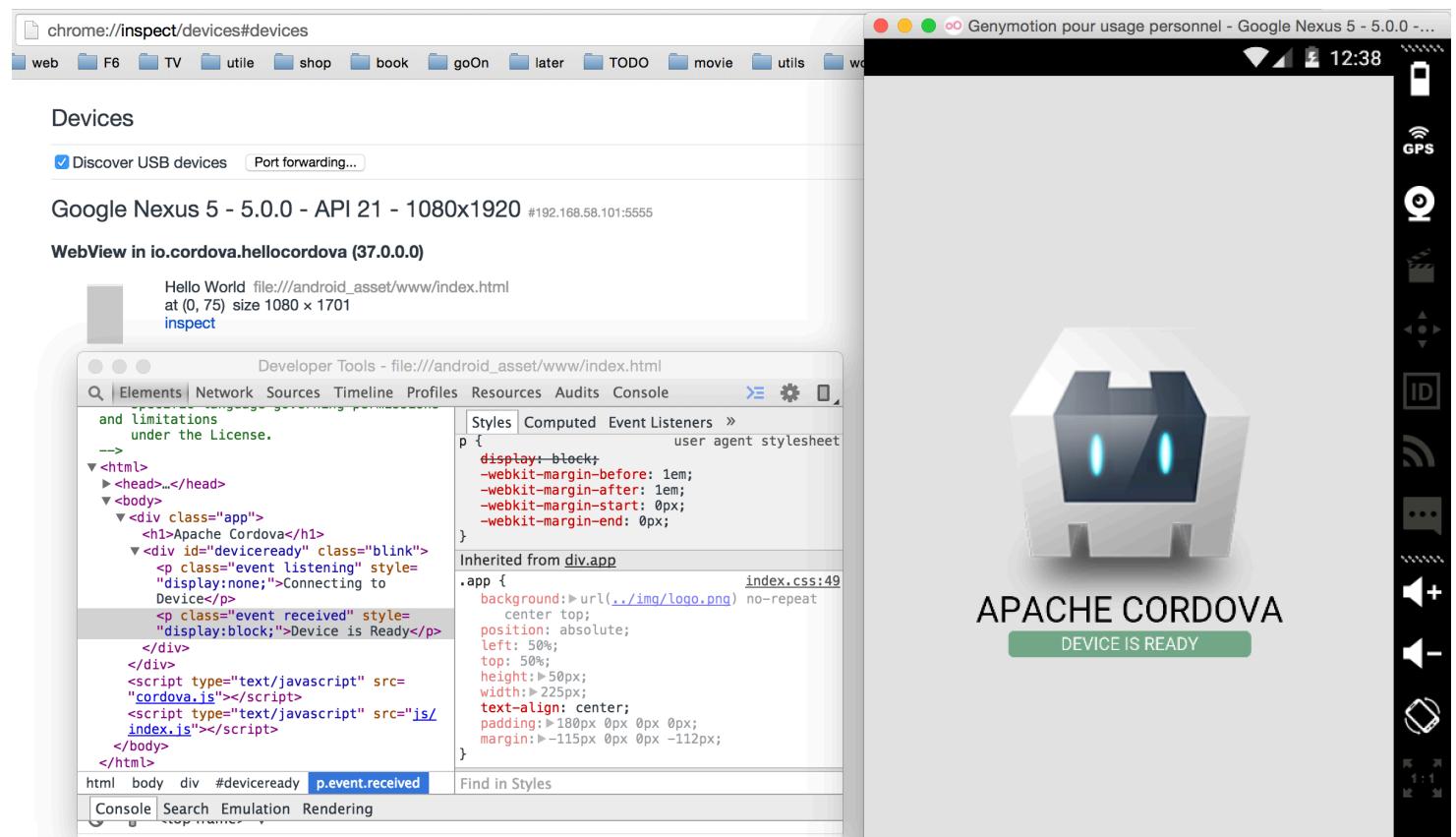


\$ cordova run android



Chrome debug

- App WebView: Android 4.4+ and a WebView configured for debugging.
- **Navigateur: Android 4.0+ and Chrome for Android.**





- **Installation du simulateur**

```
sudo npm install -g ios-sim
```

- **Gestion des devices**

▲ To find out what are the simulator images available you can use to list them

74

```
$ ./platforms/ios/cordova/lib/list-emulator-images
iPhone-4s
iPhone-5
iPhone-5s
iPhone-6-Plus
iPhone-6
iPad-2
iPad-Retina
iPad-Air
Resizable-iPhone
Resizable-iPad
```



Then use one of the simulator names in the --target parameter:

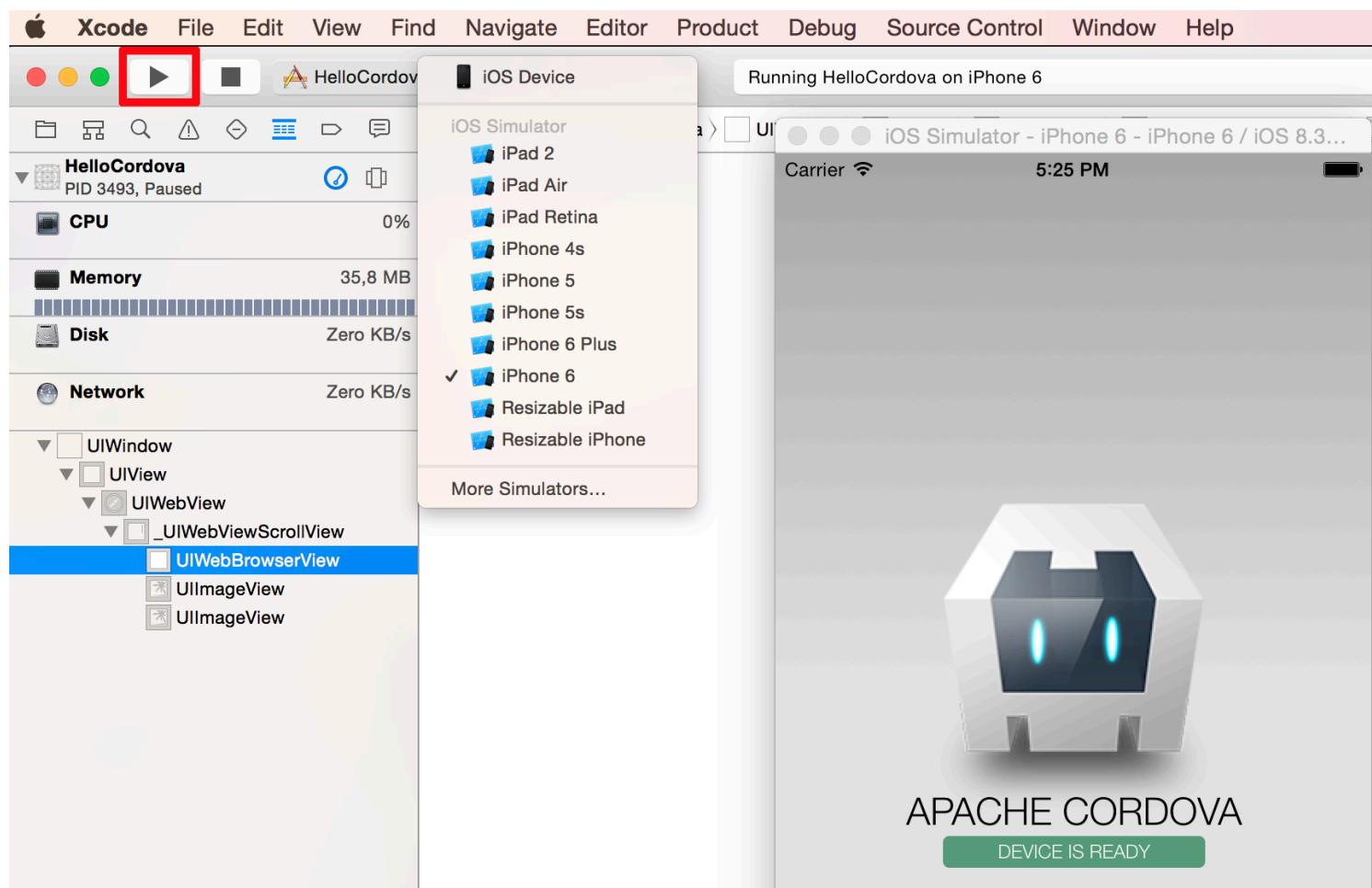
```
cordova emulate ios --target="iPhone-4s"
cordova emulate ios --target="iPad-Air"
cordova emulate ios --target="iPhone-6"
cordova emulate ios --target="iPhone-6-Plus"
```

Take into account that you may need to quit the iOS simulator via menu to switch from 3.5 to 4 inch iPhone.

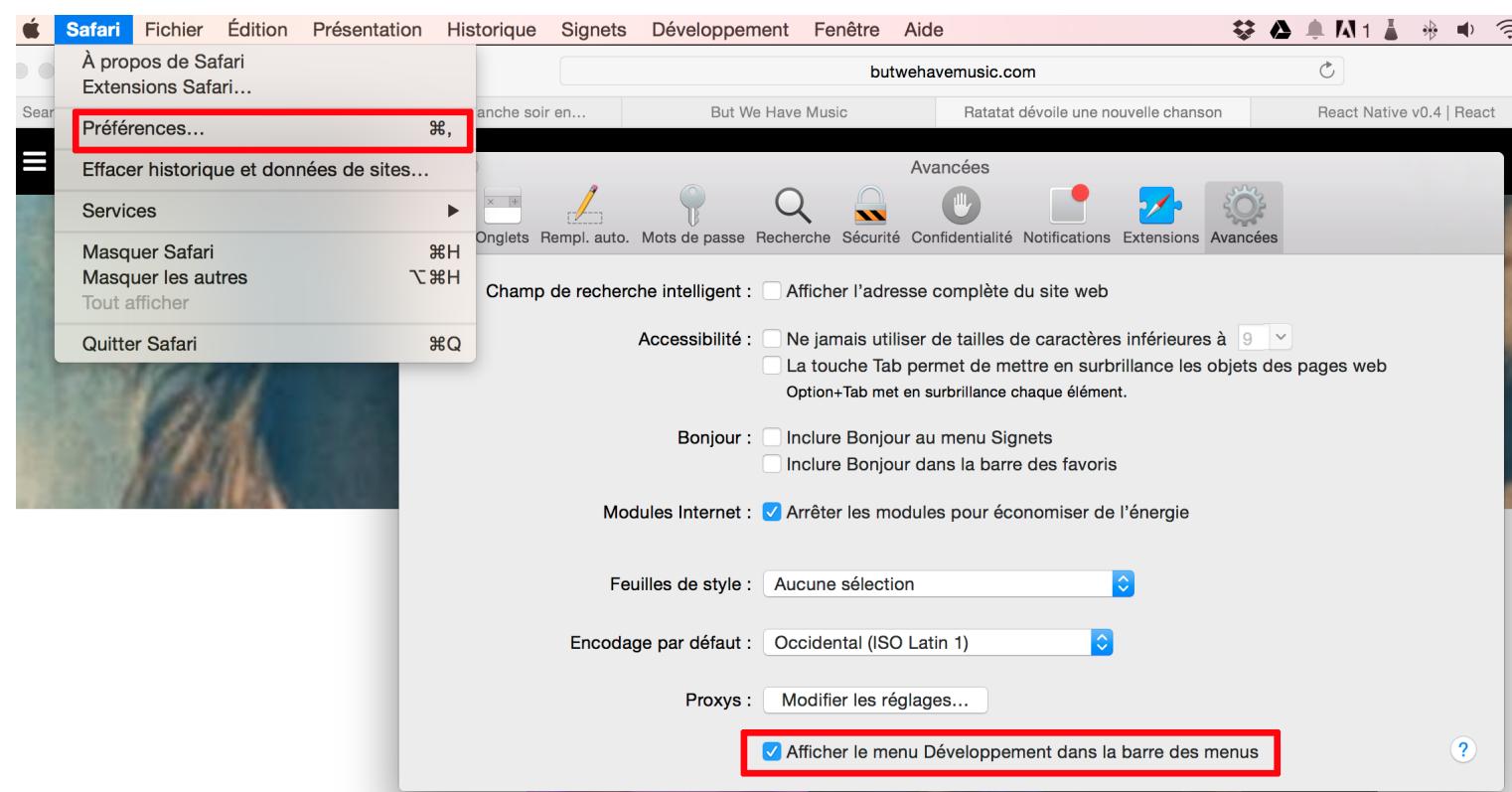
dynamic list is available in platforms/ios/cordova/lib/list-emulator-images

IOS : lancer le simulateur depuis Xcode

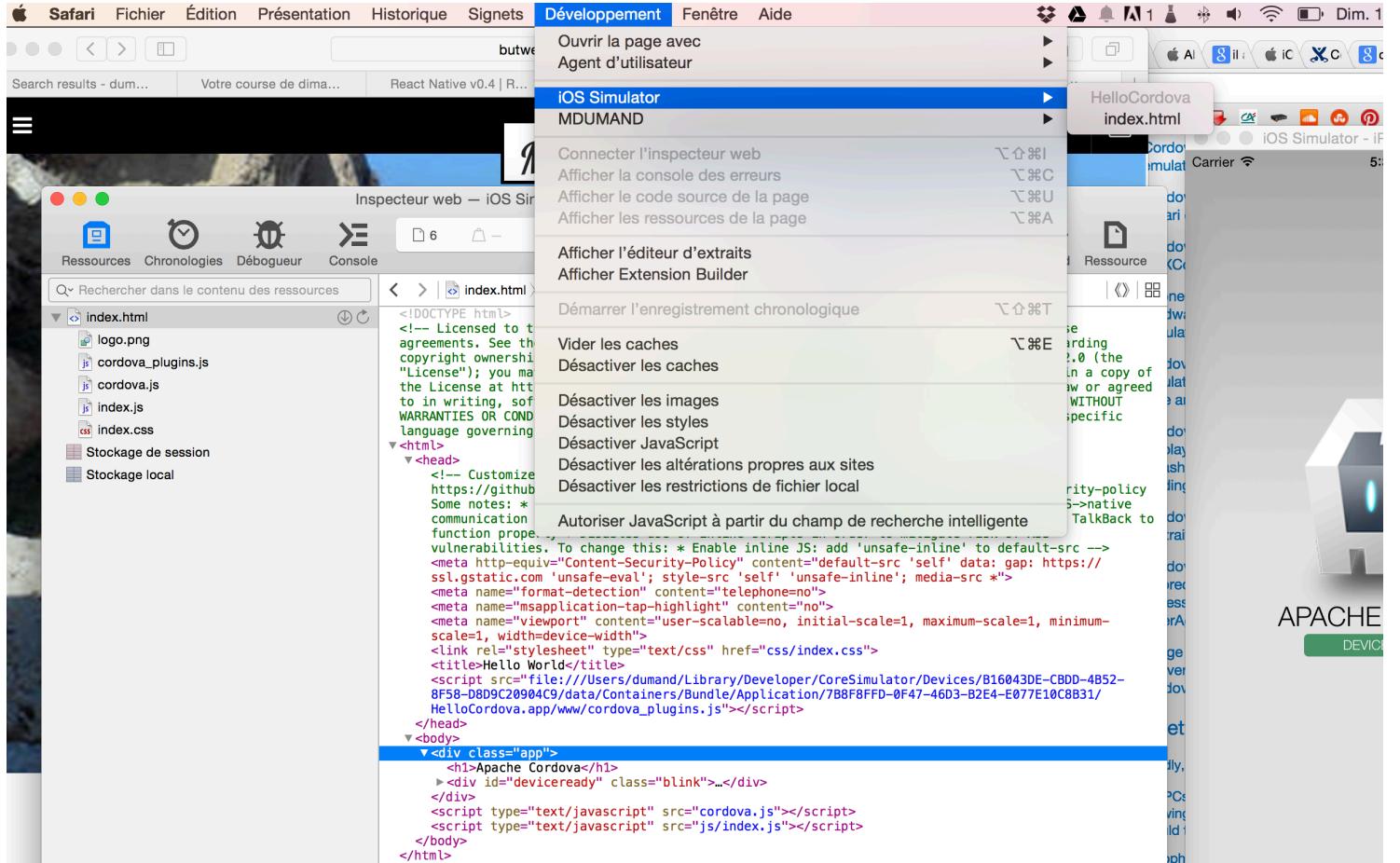
Ouvrir platforms/ios/HelloCordova.xcodeproj



IOS : Afficher le menu développement Safari

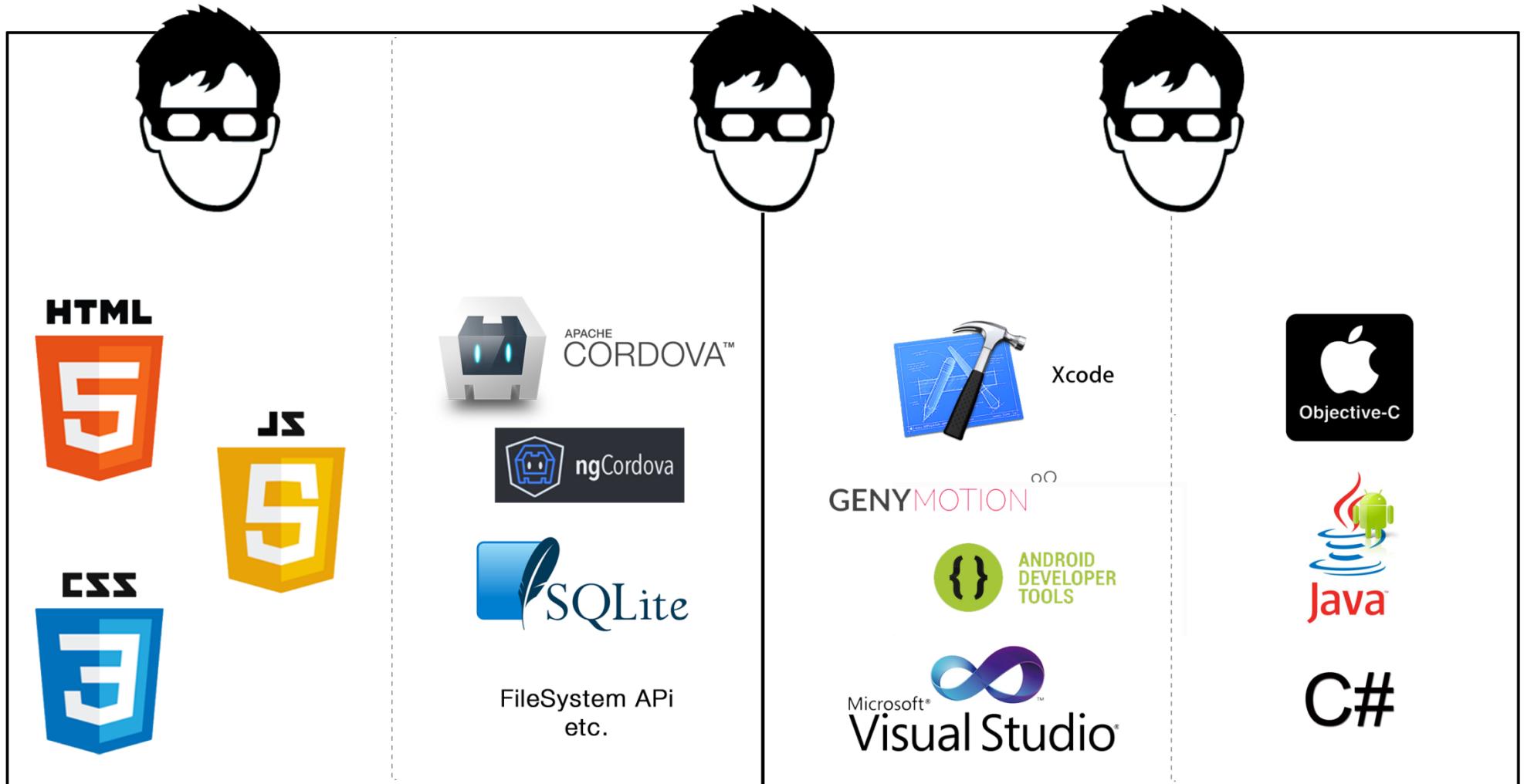


IOS : Inspecter une WebView dans Safari



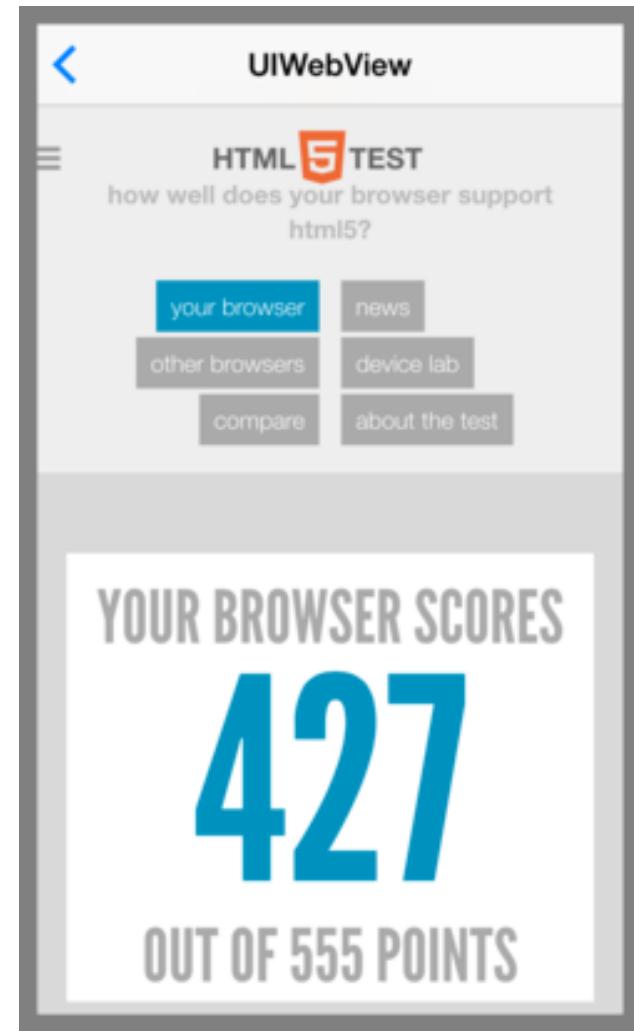
Une chose est sûre !

Développeur web != développeur hybride != développeur natif



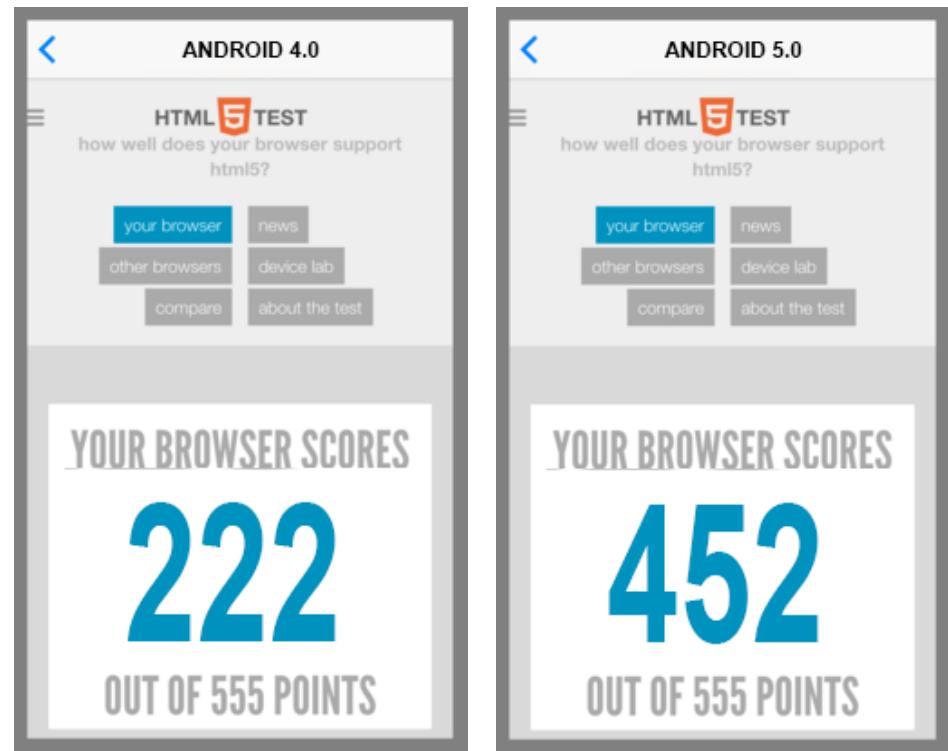
Statut des WebViews aujourd'hui

Sur IOS:
Bon support HTML5 et
bonne performance



The image shows a screenshot of a UIWebView interface. At the top, it displays the 'UIWebView' title and a back arrow icon. Below this, there is a header with the 'HTML5 TEST' logo and the text 'how well does your browser support html5?'. A navigation menu below the header includes buttons for 'your browser' (highlighted in blue), 'news', 'other browsers', 'device lab', 'compare', and 'about the test'. The main content area is titled 'YOUR BROWSER SCORES' and features a large blue '427' score. Below the score, it says 'OUT OF 555 POINTS'.

Sur Android: Les premières versions de WebView sont lentes !



Plus pour longtemps.

Android 3.2.1	534.13
Android 4.0.1	534.30
Android 4.0.2	534.30
Android 4.0.3	534.30
Android 4.0.4	534.30
Android 4.1.1	534.30
Android 4.1.2	534.30
Android 4.2	534.30
Android 4.2.1	534.30
Android 4.2.2	534.30
Android 4.3	534.30
Android 4.4.x	537.36
Android 5.0.x	537.36

En voilà des **bonnes nouvelles**.

Intégration du moteur de Chrome pour la version 4.4.x !

Mise à jour automatique et indépendante de l'OS à partir de la version 5.0.x !!!

L'hybride est sans aucun doute, une solution d'avenir !

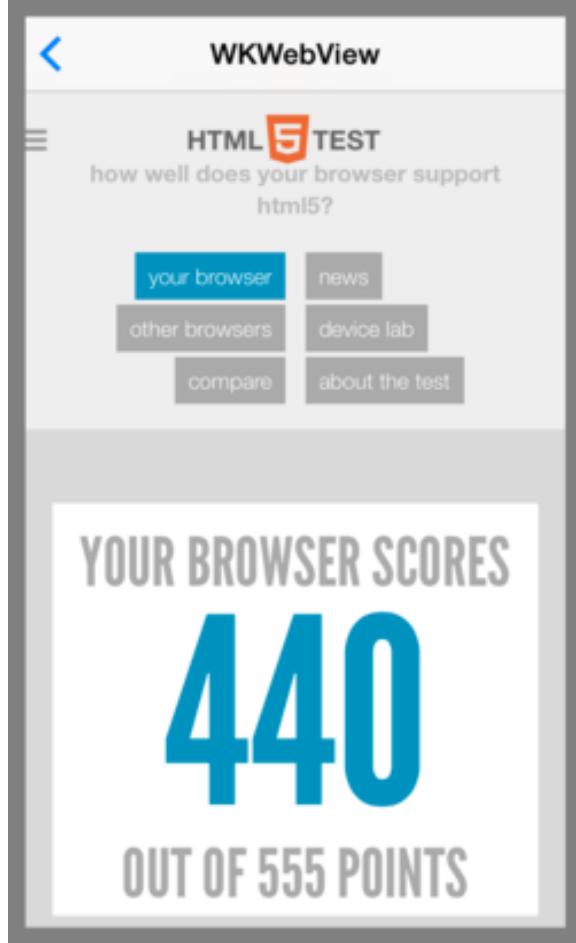
**Et aujourd’hui,
des solutions ?**

Nouvelle WebView iOS8: WKWebView

De meilleures performances,
activable via un plugin !



 apache / cordova-plugins / wkwebview-engine /
mirrored from git://git.apache.org/cordova-plugins.git



A screenshot of a mobile application window titled "WKWebView". The top navigation bar shows a back arrow and the title "WKWebView". Below the title is a section for "HTML5 TEST" with the subtext "how well does your browser support html5?". There are several buttons: "your browser" (highlighted in blue), "other browsers", "compare", "news", "device lab", and "about the test". The main content area displays "YOUR BROWSER SCORES" in large text, followed by a large blue number "440", and "OUT OF 555 POINTS" below it.

Crosswalk pour Android

Embarquez une WebView « Moderne » dans vos applications



No developer wants old, outdated devices to prevent them from using awesome new APIs.

With the platform variations in the market you are forced to assume that every Android WebView works differently. There are differences in:

Chromium/Blink

- Meilleure compatibilité Css/JavaScript



Une meilleure compatibilité HTML5 et une amélioration des performances

Ça change le poids de mon application ?

- Application installé = ~58Mb
- En téléchargement = ~20Mb

Partager crosswalk entre plusieurs applications ?

Can one Crosswalk installation be shared between multiple applications?

Bundling the runtime with the application (aka "embedded mode") is the simplest approach for distribution purposes. But Crosswalk applications *can* share a single Crosswalk runtime library (in "shared mode"); and a package which enables shared mode is part of the Crosswalk for Android distribution. However, you would have to distribute this shared runtime package yourself.

Testons sur Galaxy SIII avec HTML5Test

- Avec Crosswalk : 494 // SunSpider : 1670,5ms
- Sans Crosswalk : 326 // SunSpider : 1732,7ms

Initiation aux environnements de développement

Node.js en bref

Du javascript côté Serveur

Création d'applications backend complètes en js

Un vaste panel de plateformes

- OS X
- Microsoft Windows
- Linux
- FreeBSD
- IBM i.

Construit à partir du moteur Js de Chrome

V8 Javascript Engine (

<https://code.google.com/p/v8/>)

Node.js en quelques lignes... de code

Créer un serveur web est aussi simple que les quelques lignes de code suivantes :

```
/* app.js */
var http = require('http');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World\n');
}).listen(1337, '127.0.0.1');

console.log('Serveur accessible via http://127.0.0.1:1337/');
```

Et pour le lancer, exécutez :

```
$ node app.js
Serveur accessible via http://127.0.0.1:1337/
```

Node Package Manager

Node modules

- Des modules publiés par la communauté
- Environ 115k modules
- Une moyenne de 700M de téléchargements le mois dernier (Novembre 2014)

Et pour le front-end ?

- Compactage de code
- Analyseurs statiques
- Générateurs (Yeoman)
- Etc.



Node.js & NPM, comment ça marche ? (1/4)

Prérequis

- NodeJS

Installation de module

```
$ npm install gulp
```

Aussi simple que ça !

Le résultat

- Création d'un répertoire « node_modules »
- Téléchargement des fichiers dans un sous-répertoire « gulp »
- Situé à l'emplacement d'exécution de la commande

Node.js & NPM, « package.json » (2/4)

Le fichier « package.json »

- Configuration de votre projet
- Références des modules

Son intérêt

- Ne pas commiter le répertoire « node_modules »
- Maintenabilité de vos dépendances
- Travail en équipe

La création du fichier

```
$ npm init
```

Exécution du « wizard » de configuration de votre projet

- Nom
- Version
- Mots-clés
- Url repo git
- ...

Node.js & NPM, « package.json » (2/4)

Avec l'option --save-dev

```
$ npm install --save-dev grunt
```

Résultat

Ajout du module dans la section
« devDependencies » de package.json

```
{
  ...
  "devDependencies": {
    "grunt": "^0.4.5"
  }
}
```

Avec l'option --save

Ajout du module dans la section
« dependencies » de package.json

A noter

« dependencies » (runtime)
« devDependencies » uniquement pour le
développement

Node.js & NPM, « package.json » (2/4)

Désinstallation d'un paquet

```
$ npm uninstall grunt
```

- Suppression du sous-répertoire « node_modules/grunt »

Désinstallation de la dépendance

```
$ npm uninstall --save-dev grunt
```

- Suppression du sous-répertoire « node_modules/grunt »
- Suppression de la référence dans le « package.json »

La variante --save fonctionne également

Node.js & NPM, Installations Globales (3/4)

Installation Globale

```
$ npm install grunt -g
```

- Ajout de la dépendance dans le répertoire Node
- Accessible pour tout les projets

Attention tout de même !

- Dépendance non disponible dans « package.json »
- Possibilité de conflits sur les versions

Restauration de paquets

```
$ npm install
```

- Installation des dépendances
- « devDependencies » et « dependencies »

Node.js & NPM, Quelques commandes (3/4)

npm ls

Liste des dépendances du projet

npm ls --depth=(0,1,2,3)

npm ls -g --depth=(0,1,2,3)

Récupération de dépendance déjà existante sur le projet

npm search

Découverte des modules en relation avec la recherche

Node.js & NPM, Le Versioning (4/4)

Versioning

Npm utilise la norme « Semantic Versioning »

Voici quelques exemples :

1.3.5:

Utiliser cette version uniquement

~1.3.5 ou 1.3.x

Ne mettre à jour que les patchs (ou 1.3.5 minimum et 1.4.0 maximum)

^1.3.5

Ne mettre à jour que les versions mineures (ou 1.3.5 minimum et 2.0.0 maximum)

latest ou *:

Toujours mettre à jour le module

Bower.

Bower, comment ça marche ? (1/3)

Présentation

- NPM pour le Front-End
- Des librairies et frameworks (jQuery, Angular, Bootstrap etc.)

Installation

```
$ npm install -g bower
```

Fichier de configuration « bower.json »

```
$ bower init
```

Même concept que « package.json »

Bower, Installer des paquets (2/3)

Installation de paquet

```
$ bower install --save jquery
```

- Gestion identique que « package.json »
- « devDependencies » et « dependencies »
- Création d'un répertoire « bower_components »

Variantes

```
$ bower install jquery#1.9.1
```

Spécification de version

```
$ bower install git:/github.com/user/package.git
```

Depuis un dépôt git

```
$ bower install http://example.com/script.js
```

Et même depuis un fichier hébergé sur un serveur !

Bower, Configuration (3/3)

Configuration

Un fichier « `.bowerrc` »

Placé à la racine de votre projet

Quelques exemples

directory

Spécifier le chemin du « `bower_components` »

timeout

Limite de temps pour l'installation d'un paquet

registry

Spécifier un serveur différent du dépôt Bower initial

(Développements de librairies internes)

etc.

Grunt.

Grunt, comment ça marche ? (1/3)

Présentation

- Un « task runner » sur Node
- Compactage de ressources
- Tests unitaires
- Révisions de fichiers
- Etc.

Une API complète à votre disposition

file

- Interactions avec le « file system »
- Écriture
- Lecture

template

- Dynamiser vos tâches

Les modules

- Officiels grunt-contrib-
- Non-officiels grunt-

Grunt, Gruntfile.js (2/3)

Le fichier Gruntfile.js

```
module.exports = function(grunt) {  
    // Do grunt-related things in here  
};
```

- Configuration de vos tâches

Composition d'une tâche

```
grunt.registerTask('formation',  
    'Task description',  
    function () {  
        var config = grunt.config.requires('formation');  
  
        console.log(config);  
    });
```

- Un nom
- Une description
- Une fonction

Sa configuration

```
grunt.config.set('formation', {  
    toto: true  
});
```

Son chargement

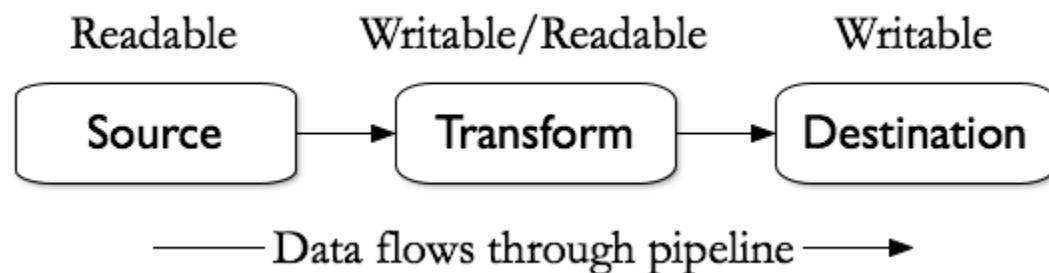
```
grunt.loadNpmTasks('grunt-contrib-connect');
```

Gulp.

Gulp, Le réel avantage (1/3)

Utilisation des « nodes streams »

- Des tunnels Unix
- Lire des données et les rediriger
- Gulp : chaîne de « streams »
- Transport jusqu'à la fin de la tâche



Gulp, comment ça marche ? (2/3)

Présentation

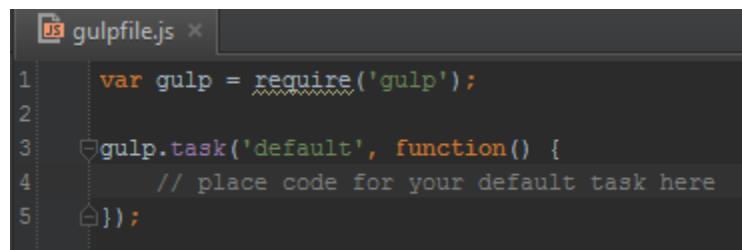
- Même objectif que Grunt : Task Runner
- Préférer le code à la configuration
- Utilisation des « Node Streams »

Installation

```
$ npm install -g gulp
```

« gulpfile.js »

- Regroupement des tâches
- Import des modules



A screenshot of a code editor showing a file named 'gulpfile.js'. The code in the editor is:

```
1 var gulp = require('gulp');
2
3 gulp.task('default', function() {
4     // place code for your default task here
5 });

```

Gulp, Modules relatifs (3/3)

Quelques exemples :

- gulp-concat : concaténation de javascript
- gulp-uglify : compacteur de javascript
- gulp-cssmin : compacteur de css
- gulp-jshint : le célèbre outil de qualité pour le javascript inclus dans une tâche Gulp
- gulp-sass : appel du préprocesseur sass

La création d'une tâche

```
1  var gulp = require('gulp'),
2      concat = require('gulp-concat'),
3      uglify = require('gulp-uglify');
4
5  gulp.task('default', function() {
6      // place code for your default task here
7  });
8
9  gulp.task('scripts', function() {
10     return gulp.src('app/scripts/**/*.{js}')
11         .pipe(concat('app.min.js'))
12         .pipe(uglify())
13         .pipe(gulp.dest('dist/'));
14 })
```

Les modules

- Pas d'exigences de nommage
- Possibilité d'utiliser n'importe quel paquet node

Yeoman.

Yeoman, comment ça marche ? (1/3)

Présentation

- Un générateur front-end
- Équipe dédiée
- Des infrastructures de projets

Quel intérêt ?

- Accélérer la mise en place d'un nouveau projet.
- Un regroupement de bonnes pratiques
- Des outils actuels et maintenus
- Une bonne façon de progresser

Installation

```
$ npm install -g yo
```

Les générateurs

- Un moteur de recherche :
<http://yeoman.io/generators/>
- Label officiel
- 1374 générateurs Aujourd'hui (30/01/2015) !

Yeoman, Installer un générateur (2/3)

Installation d'un générateur

```
$ npm install -g [generator-name]
```

Un module node à installer en global

Le générateur AngularJS

```
$ npm install -g generator-angular
```

Création d'un projet

```
$ yo angular [app-name]
```

- Création de l'arborescence
- Dépendances Node et Bower
- Gruntfile ou Gulpfile

Yeoman, Installer un générateur (3/3)

Les commandes relatives

- Spécifiques au générateur
- Personnalisable à volonté

Construction des composants Angular

```
rioscampo@BRIOSCAMPO /c/www/formation-ing
$ yo angular:controller home
  create app\scripts\controllers\home.js
  create test\spec\controllers\home.js

rioscampo@BRIOSCAMPO /c/www/formation-ing
$ cat app/scripts/controllers/home.js
'use strict';

/**
 * @ngdoc function
 * @name formationIngeApp.controller:HomeCtrl
 * @description
 * # HomeCtrl
 * Controller of the formationIngeApp
 */
angular.module('formationIngeApp')
  .controller('HomeCtrl', function ($scope) {
    $scope.awesomeThings = [
      'HTML5 Boilerplate',
      'AngularJS',
      'Karma'
    ];
  });

rioscampo@BRIOSCAMPO /c/www/formation-ing
$ cat app/index.html
<script src="scripts/controllers/home.js"></script>
```

Initiation aux tests unitaires 'front'

Developer Guide

- [Introduction](#)
- [Conceptual Overview](#)
- [Data Binding](#)
- [Controllers](#)
- [Services](#)
- [Scopes](#)
- [Dependency Injection](#)
- [Templates](#)
- [Expressions](#)
- [Filters](#)
- [Forms](#)
- [Directives](#)
- [Animations](#)
- [Modules](#)
- [HTML Compiler](#)
- [Providers](#)
- [Bootstrap](#)
- [Unit Testing](#)
- [E2E Testing](#)
- [Using \\$location](#)
- [Working With CSS](#)
- [i18n and I10n](#)
- [Security](#)
- [Accessibility](#)
- [Internet Explorer Compatibility](#)
- [Running in Production](#)
- [Migrating from Previous Versions](#)

With great power comes great responsibility

Angular is written with testability in mind, but it still requires that you do the right thing. We tried to make the right thing easy, but if you ignore these guidelines you may end up with an untestable application.

Dependency Injection

Angular comes with [dependency injection](#) built-in, which makes testing components much easier, because you can pass in a component's dependencies and stub or mock them as you wish.

Components that have their dependencies injected allow them to be easily mocked on a test by test basis, without having to mess with any global variables that could inadvertently affect another test.

Additional tools for testing Angular applications

For testing Angular applications there are certain tools that you should use that will make testing much easier to set up and run.

Karma

[Karma](#) is a JavaScript command line tool that can be used to spawn a web server which loads your application's source code and executes your tests. You can configure Karma to run against a number of browsers, which is useful for being confident that your application works on all browsers you need to support. Karma is executed on the command line and will display the results of your tests on the command line once they have run in the browser.

Installation



Karma runs on [Node.js](#) and is available as an [NPM](#) package.

Installing Node.js or iojs

There are [node.js](#) or [iojs](#) for both Mac and Windows. On Linux, we recommend using [NVM](#).

Note: Karma works on the three latest stable versions of node. That is **0.8.x**, **0.10.x** and **0.12.x** at this point. Also works on iojs 2.x.x.

Installing Karma and plugins

The recommended approach is to install Karma (and all the plugins your project needs) locally in the project's directory.

```
# Install Karma:  
$ npm install karma --save-dev  
  
# Install plugins that your project needs:  
$ npm install karma-jasmine karma-chrome-launcher --save-dev
```

This will install `karma` , `karma-jasmine` and `karma-chrome-launcher` packages into `node_modules` in your current working directory and also save these as `devDependencies` in `package.json` , so that any other developer working on the project will only have to do `npm install` in order to get all these dependencies installed.

Exemple

untitled

```
1 "use strict";
2
3
4 describe("Factory : locale", function () {
5   var localeFactory, $i18next, prefsFactory;
6
7   beforeEach(module("casaApp"));
8
9   beforeEach(inject(function (_localeFactory_, _$i18next_, _prefsFactory_) {
10    localeFactory = _localeFactory_;
11    $i18next = _$i18next_;
12    prefsFactory = _prefsFactory_;
13    prefsFactory.createCache();
14 }));
15
16  afterEach(inject(function () {
17    localeFactory.setLocale('en');
18  }));
19
20  it("should switch the i18next locale", function () {
21    localeFactory.setLocale('fr');
22    expect($i18next.options.lng).toBe('fr');
23  });
24
25  it("should switch the moment locale", function () {
26    localeFactory.setLocale('fr');
27    expect(moment.locale()).toBe('fr');
28  });
29
30  it("should return locales weights", function () {
31    expect(localeFactory.getLocaleIndex('en')).toBe(1);
32    expect(localeFactory.getLocaleIndex('fr')).toBe(0);
33  });
34
35 });
36
```

v1.4.2-build.4090 (snapshot)

Developer Guide

Introduction

Conceptual Overview

Data Binding

Controllers

Services

Scopes

Dependency Injection

Templates

Expressions

Filters

Forms

Directives

Animations

Modules

HTML Compiler

Providers

Bootstrap

Unit Testing

E2E Testing

Using \$location

Working With CSS

i18n and i10n

Security

Accessibility

Internet Explorer Compatibility

Running in Production

Separation of Concerns

Unit testing, as the name implies, is about testing individual units of code. Unit tests try to answer questions such as "Did I think about the logic correctly?" or "Does the sort function order the list in the right order?"

In order to answer such a question it is very important that we can isolate the unit of code under test. That is because when we are testing the sort function we don't want to be forced into creating related pieces such as the DOM elements, or making any XHR calls to fetch the data to sort.

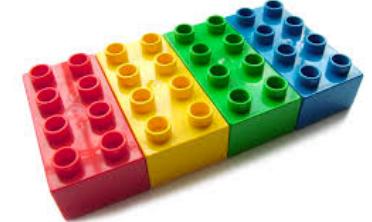
While this may seem obvious it can be very difficult to call an individual function on a typical project. The reason is that the developers often mix concerns resulting in a piece of code which does everything. It makes an XHR request, it sorts the response data and then it manipulates the DOM.

With Angular we try to make it easy for you to do the right thing, and so we provide dependency injection for your XHR requests, which can be mocked, and we provide abstractions which allow you to test your model without having to resort to manipulating the DOM. The test can then assert that the data has been sorted without having to create or look at the state of the DOM or wait for any XHR requests to return data. The individual sort function can be tested in isolation.

With great power comes great responsibility

Angular is written with testability in mind, but it still requires that you do the right thing. We tried to make the right thing easy, but if you ignore these guidelines you may end up with an untestable application.

Separation of concerns



Coupling

The degree of dependency between two modules



Maintainability

A measure of how easy it is to maintain the system.



Reusability

A measure of how easy it is to reuse a module in a different system.



Extensibility

A measure of how easily the system can be extended with new functionality.

