Applicazioni per dispositivi mobili

Corso di laurea in Informatica

Apache Cordova

Amleto Di Salle
Università degli studi dell'Aquila
Dipartimento di Ingegneria, Scienze dell'informazione e Matematica
Email: amleto.disalle@univaq.it

Outline

- Introduction
- Architecture
- Install Cordova
- Cordova CLI
- App structure
- Plugins
- Running with emulator and device

References

- https://cordova.apache.org

Introduction

Apache Cordova is an open-source mobile development framework

It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development

Applications execute within wrappers targeted to each platform

They rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc.

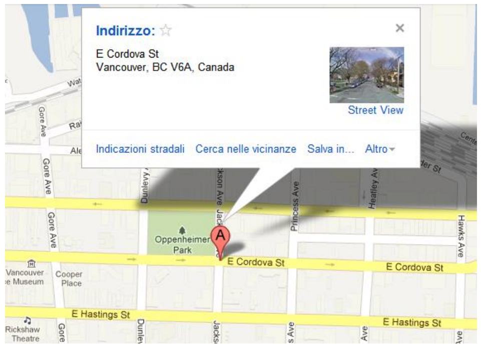
PhoneGap VS Cordova

Adobe/Nitobi donated the PhoneGap codebase to the Apache foundation

- Wider audience and contribut
- Transparent governance
- Better documentation
- Easier contributions for companies: Apache Licensing

PhoneGap is a distribution of Apache Cordova

There was only one problem....
trademark ambiguity → CORDOVA

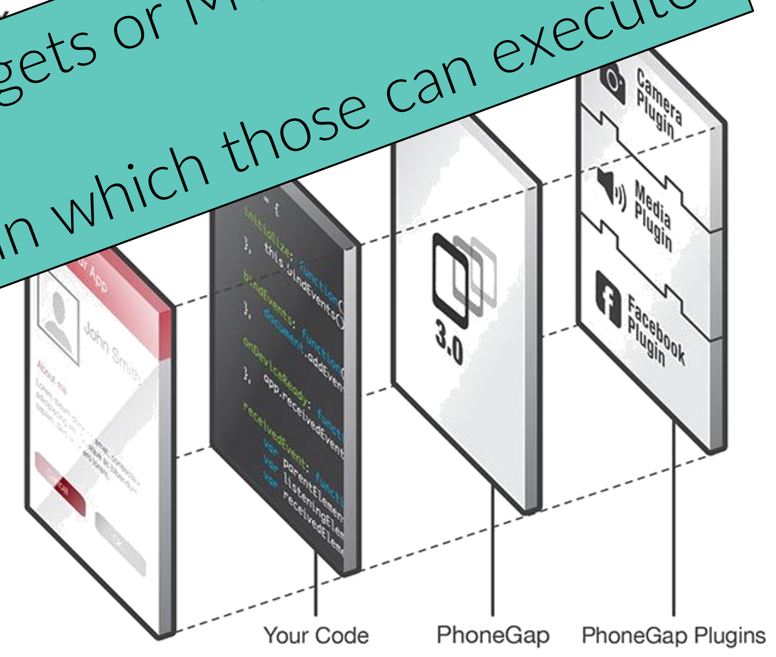


Apache Cordova (1)

You develop your app using the usual thre

Cordova does not provide any UI widgets or MV* frameworks

Cordova provides only the runtime in which those can execute



- OS = `UIWebView`
- Android = `android.webkit.WebView`

http://phonegap.com/blog/2013/06/20/coming-soon-phonegap30/

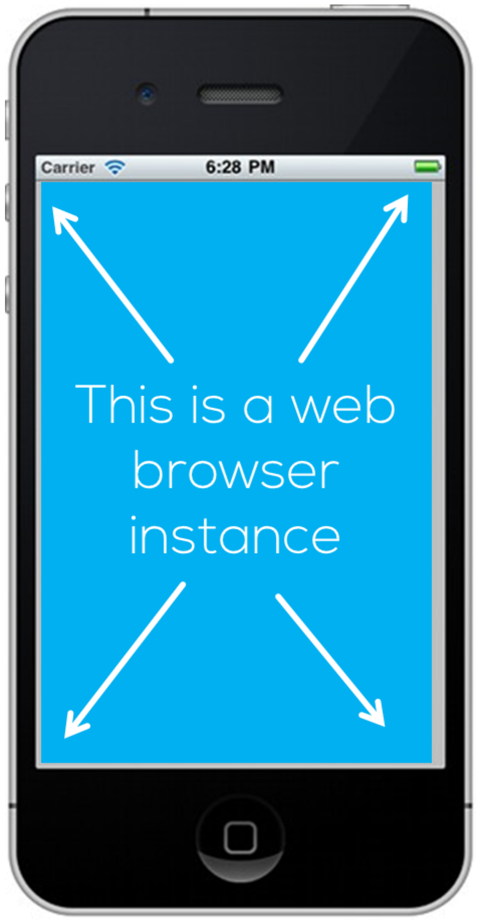
Apache Cordova (2)

The UI layer is a web browser view

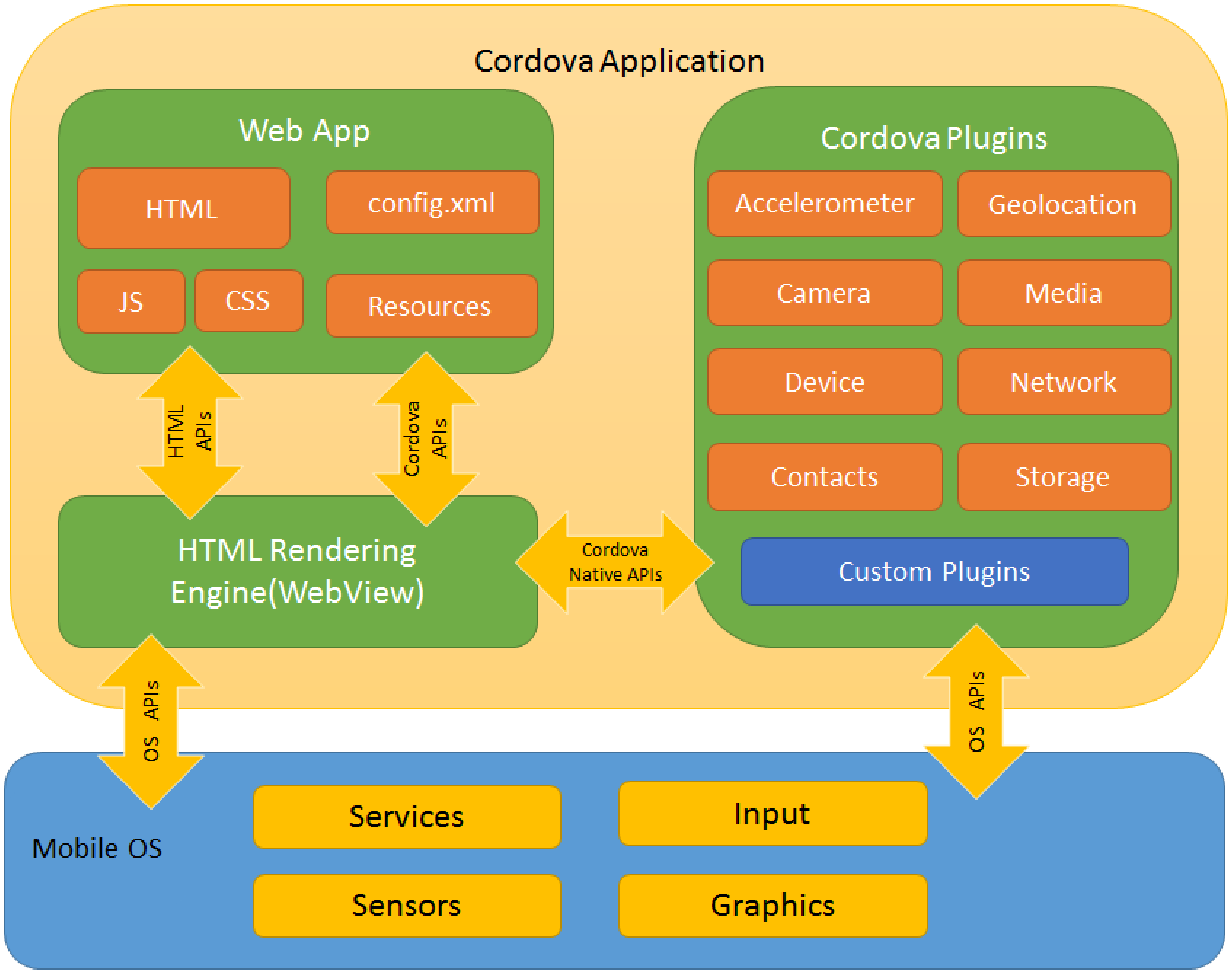
- 100% width
- 100% height

Headless web browser

- No URL bar
- No decorations
- No zooming
- No text selection



Architecture (1)



Architecture (2)

WebView

The Cordova-enabled WebView may provide the application with its entire user interface

On some platforms, it can also be a component within a larger, hybrid application that mixes the WebView with native application components

Architecture (3)

Web App

This is the part where your application code resides

Application is implemented as a web page (index.html) that references CSS, JavaScript, images, media files, or other resources

app executes in a WebView within the native application wrapper, which you distribute to app stores

`config.xml` provides information about the app and specifies parameters affecting how it works, such as whether it responds to orientation shifts

Architecture (4)

Plugins

Plugins are an integral part of the cordova ecosystem

They provide an interface for Cordova and native components to communicate with each other and bindings to standard device APIs

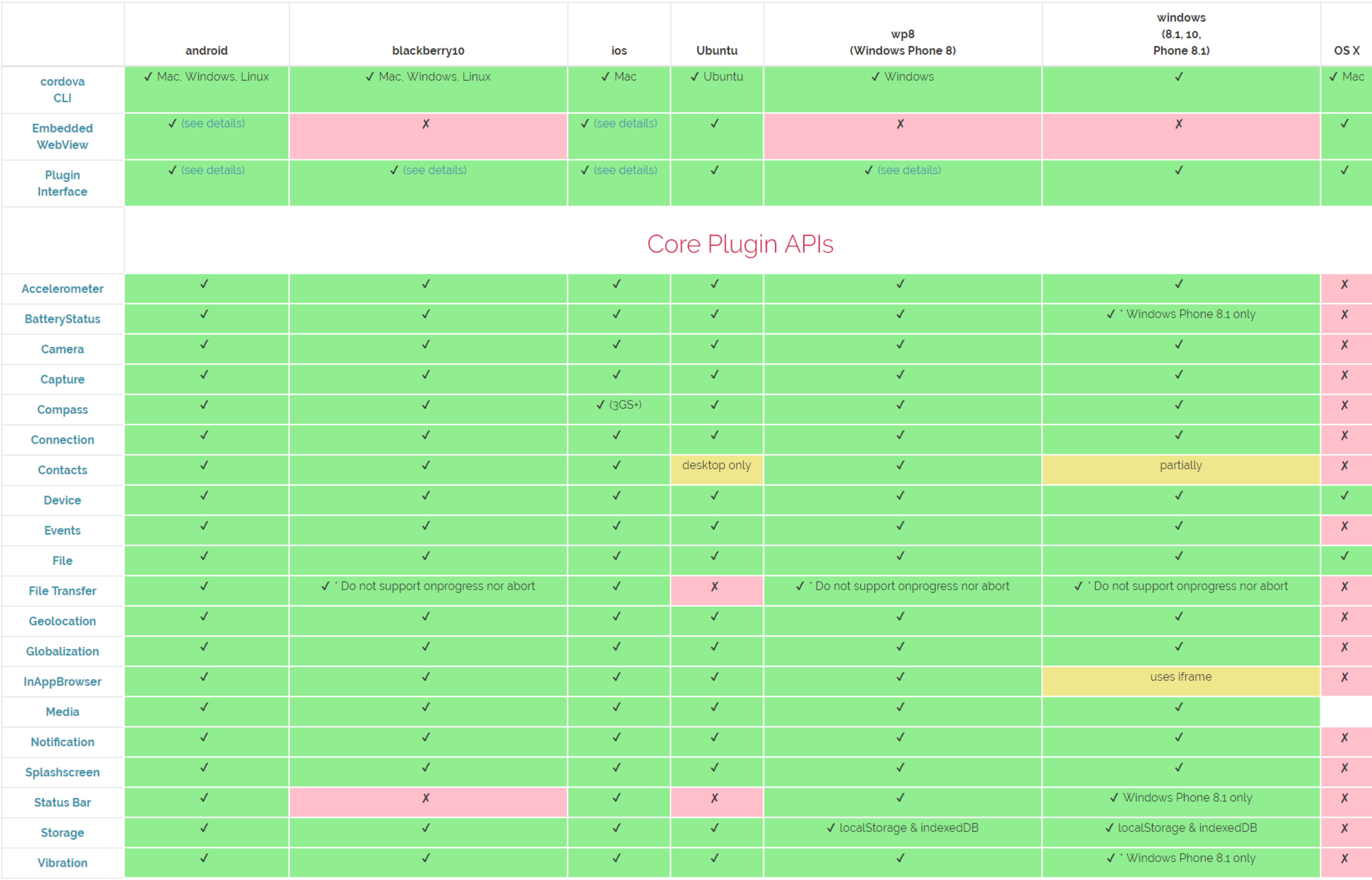
This enables you to invoke native code from JavaScript

Core plugins provide your application to access device capabilities such as battery, camera, contacts, etc.

There are several third-party plugins which provide additional bindings to features not necessarily available on all platforms

You can also develop your own plugins

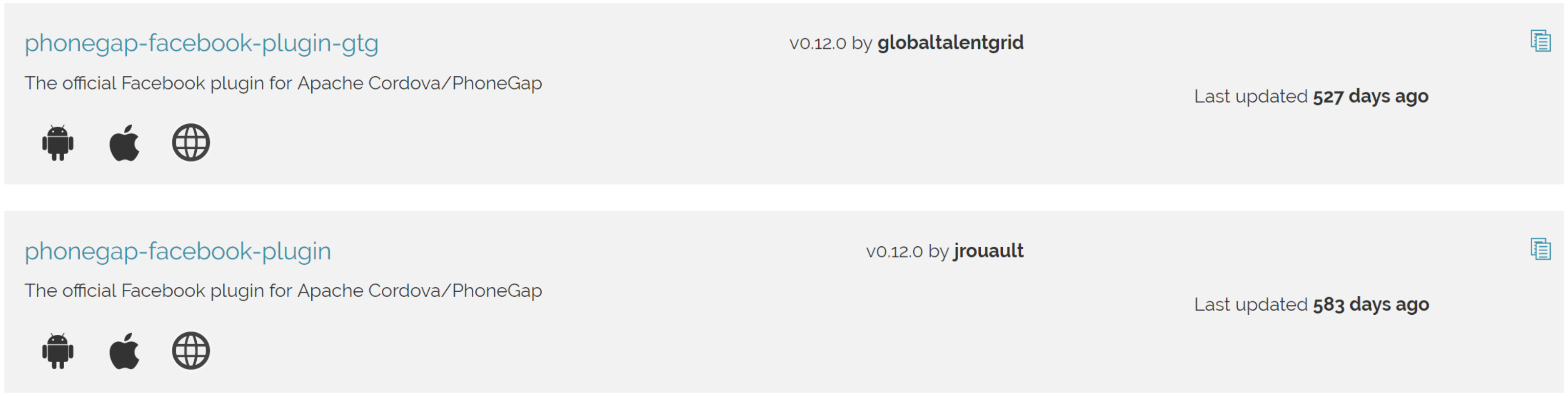
Features coverage



	android	blackberry10	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.1, 10, Phone 8.1)	OS X
cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓	✓ Mac
Embedded WebView	✓ (see details)	✗	✓ (see details)	✓	✗	✗	✓
Plugin Interface	✓ (see details)	✓ (see details)	✓ (see details)	✓	✓ (see details)	✓	✓
	Core Plugin APIs						
Accelerometer	✓	✓	✓	✓	✓	✓	✗
BatteryStatus	✓	✓	✓	✓	✓	✓ * Windows Phone 8.1 only	✗
Camera	✓	✓	✓	✓	✓	✓	✗
Capture	✓	✓	✓	✓	✓	✓	✗
Compass	✓	✓	✓ (3GS+)	✓	✓	✓	✗
Connection	✓	✓	✓	✓	✓	✓	✗
Contacts	✓	✓	✓	desktop only	✓	partially	✗
Device	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✓	✓	✓	✗
File	✓	✓	✓	✓	✓	✓	✓
File Transfer	✓	✓ * Do not support onprogress nor abort	✓	✗	✓ * Do not support onprogress nor abort	✓ * Do not support onprogress nor abort	✗
Geolocation	✓	✓	✓	✓	✓	✓	✗
Globalization	✓	✓	✓	✓	✓	✓	✗
InAppBrowser	✓	✓	✓	✓	✓	uses iframe	✗
Media	✓	✓	✓	✓	✓	✓	
Notification	✓	✓	✓	✓	✓	✓	✗
Splashscreen	✓	✓	✓	✓	✓	✓	✗
Status Bar	✓	✗	✓	✗	✓	✓ Windows Phone 8.1 only	✗
Storage	✓	✓	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✗
Vibration	✓	✓	✓	✓	✓	✓ * Windows Phone 8.1 only	✗

Examples of available plugins

https://cordova.apache.org/plugins/



phonegap-facebook-plugin-gtg v0.12.0 by **globaltalentgrid**

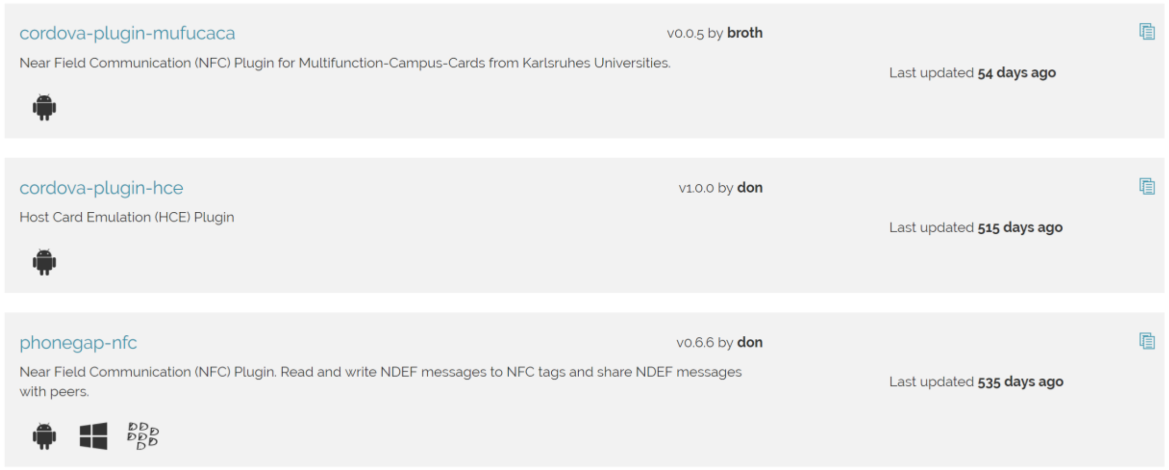
The official Facebook plugin for Apache Cordova/PhoneGap

Last updated **527 days ago**

phonegap-facebook-plugin v0.12.0 by **jrouault**

The official Facebook plugin for Apache Cordova/PhoneGap

Last updated **583 days ago**



cordova-plugin-mufucaca v0.0.5 by **broth**

Near Field Communication (NFC) Plugin for Multifunction-Campus-Cards from Karlsruhes Universities.

Last updated **54 days ago**

cordova-plugin-hce v1.0.0 by **don**

Host Card Emulation (HCE) Plugin

Last updated **515 days ago**

phonegap-nfc v0.6.6 by **don**

Near Field Communication (NFC) Plugin. Read and write NDEF messages to NFC tags and share NDEF messages with peers.

Last updated **535 days ago**

Development Paths

Cordova provides you two basic workflows to create a mobile app

Cross-platform (CLI) workflow

Use this workflow if you want your app to run on as many different mobile operating systems as possible

This workflow centers around the cordova CLI

Platform-centered workflow

Use this workflow if you want to focus on building an app for a single platform and need to be able to modify it at a lower level

Use this approach if you want your app to mix custom native components with web-based Cordova components

Cordova CLI

CLI = Command-Line Interface

The main tool to use for the cross-platform workflow

It allows you to

- create new projects
- add platforms
- build a project w.r.t. different platforms
- emulate a project on platform-specific emulators
- run a project on device
- include specific plugins into a project

Install Cordova

- Download and install Node.js
- (Optional) Download and install a git client, if you don't already have one. CLI uses it to download assets when they are referenced using a url to a git repo
- Install the cordova module using npm utility of Node.js

On OS X and Linux

```
 $ sudo npm install -g cordova
```

On Windows

```
C:\>npm install -g cordova
```

`-g` tells npm to install cordova globally (otherwise it will be installed in the node_modules subdirectory of the current working directory)

Node JS

- A platform built on Chrome’s JavaScript runtime for easily building fast, scalable network applications

- It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices

- It uses V8, the virtual machine that powers Google Chrome, for server-side programming

- V8 gives Node a huge boost in performance because it cuts out the middleman, preferring straight compilation into native machine code over executing bytecode or using an interpreter

https://nodejs.org/en/

Node Package Manager (NPM)

- It is the package manager for JavaScript and the world's largest software registry
- https://www.npmjs.com/

Create the app

Syntax

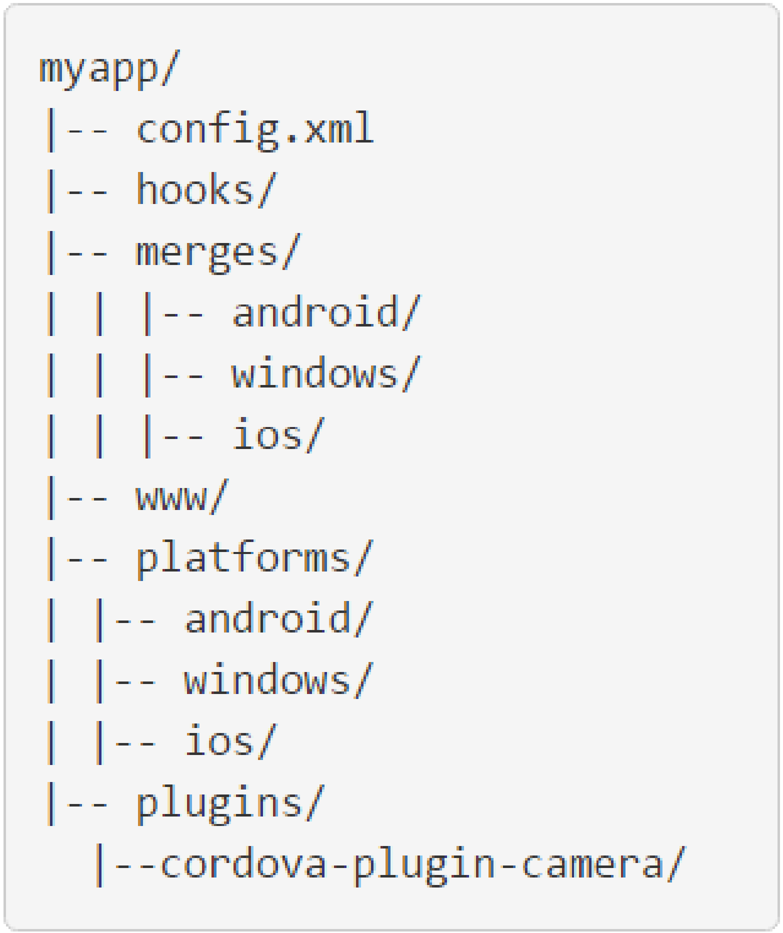
```
$ cordova create path [id [name [config]]] [options]
```

- `path`: Directory which should not already exist. Cordova will create this directory
- `id` (default `io.cordova.hellocordova`) Reverse domain-style identifier that maps to id attribute of widget element in `config.xml` (select an appropriate value)
- `name` (default: HelloCordova): Application's display title that maps name element in `config.xml` (select an appropriate value)
- config JSON string whose key/values will be included in `<path>/.cordova/config.json`

Example

```
$ cordova create hello com.example.hello HelloWorld
```

Project Structure (1)



```
myapp/
|-- config.xml
|-- hooks/
|-- merges/
| | |-- android/
| | |-- windows/
| | |-- ios/
|-- www/
|-- platforms/
| |-- android/
| |-- windows/
| |-- ios/
|-- plugins/
  |--cordova-plugin-camera/
```

Version control

It is recommended not to check in `platforms/` and `plugins/` directories into version control as they are considered a build artifact
You should save the platform/plugin spec in the config.xml and they will be downloaded when on the machine when cordova prepare is invoked

Project Structure (2)

`config.xml`

Configures your application and allows you to customize the behavior of your project

http://cordova.apache.org/docs/en/latest/config_ref/index.html

`www/`

Contains the project's web artifacts, such as .html, .css and .js files

Most of your code and assets will go here

They will be copied on a cordova prepare to each platform's www directory (for example in platforms/ios/www or platforms/android/assets/www)

Project Structure (3)

`platforms/`

Contains all the source code and build scripts for the platforms

WARNING

you should not edit any files in the /platforms/ directory unless you know what you are doing, or if documentation specifies otherwise

The files in this directory are routinely overwritten when preparing applications for building, or when plugins are re-installed

`plugins/`

Any added plugins will be extracted or copied into this directory

`hooks/`

It may contains scripts used to customize cordova-cli commands

Any scripts you add to these directories will be executed before and after the commands corresponding to the directory name

Useful for integrating your own build systems or integrating with version control systems

Project Structure (4)

`merges/`

Platform-specific web assets (HTML, CSS and JavaScript files) are contained within appropriate subfolders in this directory

These are deployed during a prepare to the appropriate native directory

Files placed under `merges/` will override matching files in the `www/` folder for the relevant platform

```
merges/
|-- ios/
| -- app.js
|-- android/
| -- android.js
www/
-- app.js
```

After building
Android application will contain both `app.js` and `android.js`
iOS application will only contain an `app.js`, and it will be the one from `merges/ios/app.js`, overriding the "common" `app.js` located inside `www/`

Add platforms

```
$ cordova platform add PLATFORM_NAME
```

`PLATFORM_NAME`

the name of the platform (e.g., ios, android, wp8)

With this command you add a target platform of your project

The platform will appear as subfolder of platforms containing the platform-specific project mirroring the `www` folder

Check and remove platform

To check your current set of platforms

```
$ cordova platform ls
```

If you want to remove a specific platform

```
$ cordova platform remove ios
```

Add Plugins (1)

A plugin exposes a Javascript API for native SDK functionality

Plugins are typically hosted on npm and you can search for them on the plugin search page

Some key APIs are provided by the Apache Cordova open source project and these are referred to as Core Plugin APIs

You can also use the CLI to launch the search page

```
$ cordova plugin search camera
```

Add Plugins (2)

To add and save the camera plugin to `config.xml` and `package.json`, we will specify the `npm` package name for the camera plugin

```
$ cordova plugin add cordova-plugin-camera
```

Use `plugin ls` (or `plugin list`, or `plugin` by itself) to view currently installed plugin

```
$ cordova plugin ls
```

Install pre-requisites for building

To build and run apps, you need to install SDKs for each platform you wish to target

Alternatively, if you are using browser for development you can use `browser` platform which does not require any platform SDKs

To check if you satisfy requirements for building the platform

```
$ cordova requirements
```

Install Android SDK (1)

Java Development Kit (JDK)

Install Java Development Kit (JDK) 8 or later

Android SDK

Install Android Studio:
`https://developer.android.com/studio/index.html`

Adding SDK Packages

You must install the packages for whatever API level you wish to target

Recommendation: install highest SDK version that your version of cordova-android supports

Open the Android SDK Manager (run `sdkmanager` from terminal) and make sure the following are installed

Android Platform SDK for your targeted version of Android

Android SDK build-tools version 19.1.0 or higher

Android Support Repository (found under "Extras")

Install Android SDK (2)

Cordova's CLI tools require some environment variables to be set in order to function correctly

The CLI will attempt to set these variables for you, but in certain cases you may need to set them manually

- Set the `JAVA_HOME` environment variable to the location of your JDK installation
- Set the `ANDROID_HOME` environment variable to the location of your Android SDK installation
- It is also recommended that you add the Android SDK's `tools`, `tools/bin`, and `platform-tools` directories to your PATH

Build the App

By default, cordova `create` script generates a skeletal web-based application (start page `www/index.html`)

Any initialization should be specified as part of the `deviceready` event handler defined in `www/js/index.js`

Run the following command to build the project for all platforms

```
$ cordova build [platform]
```

`platform` for a specific platform (i.e., `android`)

Android uses Gradle (www.gradle.org) for building the app

```
PATH=$ANDROID_STUDIO_HOME\gradle\gradle-3.2\bin
```

`$ANDROID_STUDIO_HOME` Android Studio home directory

Run in the emulator (1)

SDKs for mobile platforms often come bundled with emulators that execute a device image

you can launch the app from the home screen and see how it interacts with many platform features

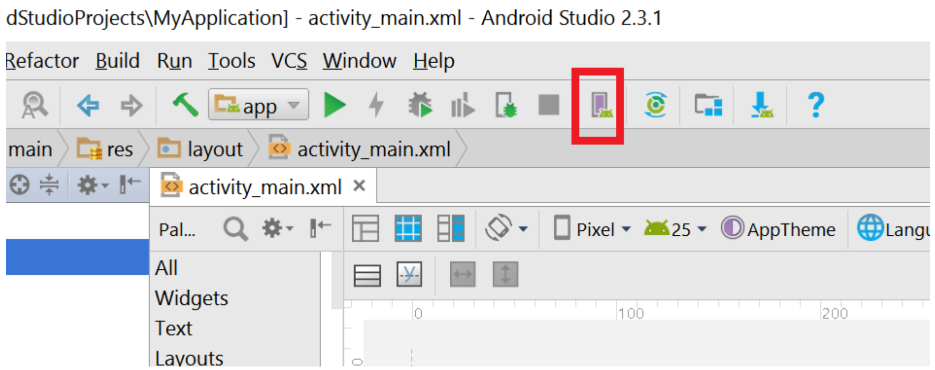
The following rebuilds the app and view it within a specific platform's emulator

```
$ cordova emulate android
```

Setting up an Emulator (1)

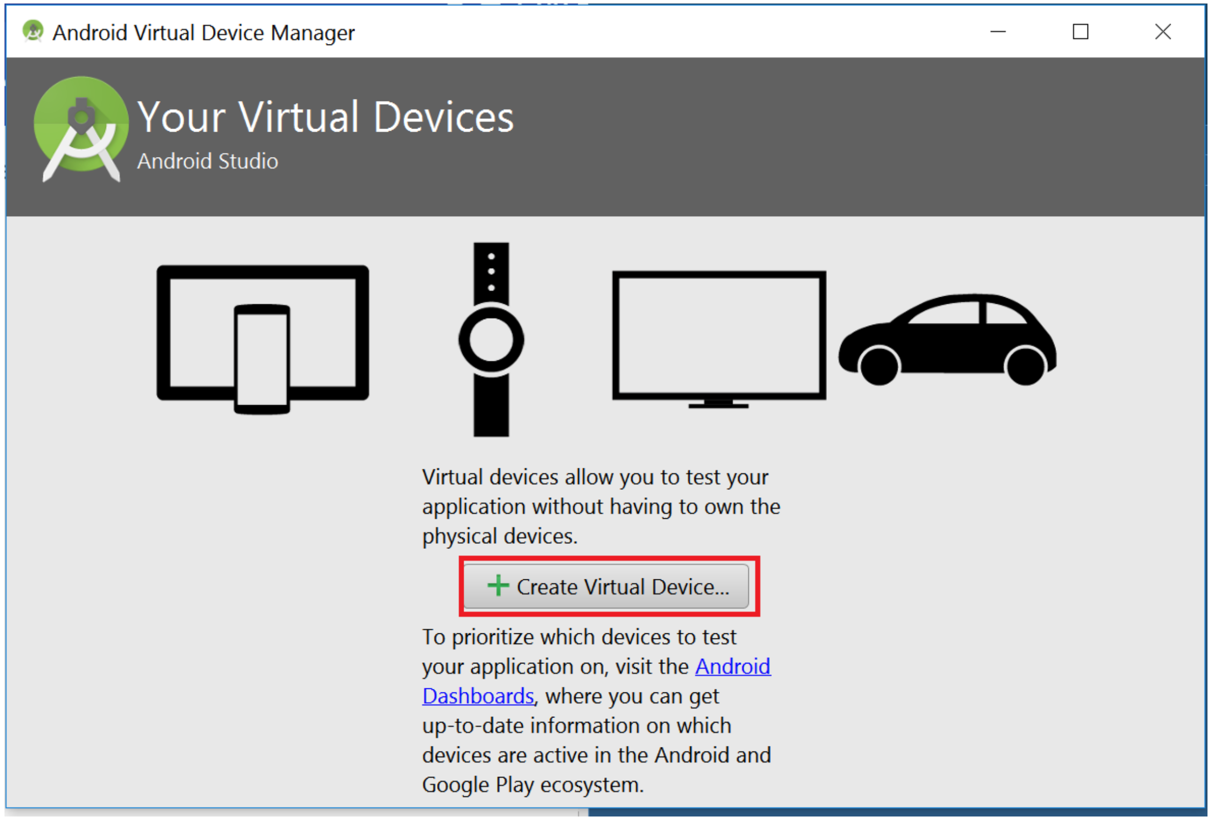
If you wish to run your Cordova app on an Android emulator, you will first need to create an Android Virtual Device (AVD)

1. Start Android Studio
2. Create and empty Android app
3. Click on


dStudioProjects\MyApplication] - activity_main.xml - Android Studio 2.3.1
Refactor Build Run Tools VCS Window Help
app
main
res
layout
activity_main.xml
activity_main.xml
Pal...
Pixel
25
AppTheme
Langu
All
Widgets
Text
Layouts
0
100
200

Setting up an Emulator (2)

- Click on «Create Virtual Device…»
- Follow the next steps



Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

Run on the device

Alternately, you can plug the handset into your computer and test the app directly

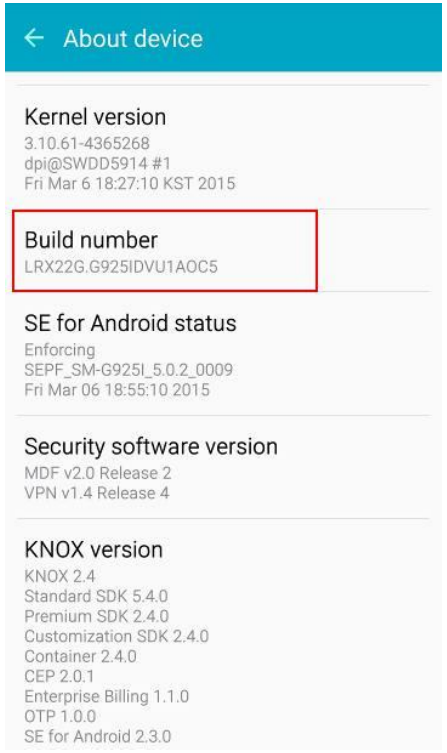
```
$ cordova run android
```

Before running this command, you need to set up the device for testing, following procedures that vary for each platform

Setting up the device

For Samsung Galaxy S6

- Step 1: Go to **Settings—About device—Info Software**
- Step 2: Tap **Build number** 7 times
- Step 3: Go to Settings – **Developer options**
- **Step 4:** Enable **USB Debugging**



About device

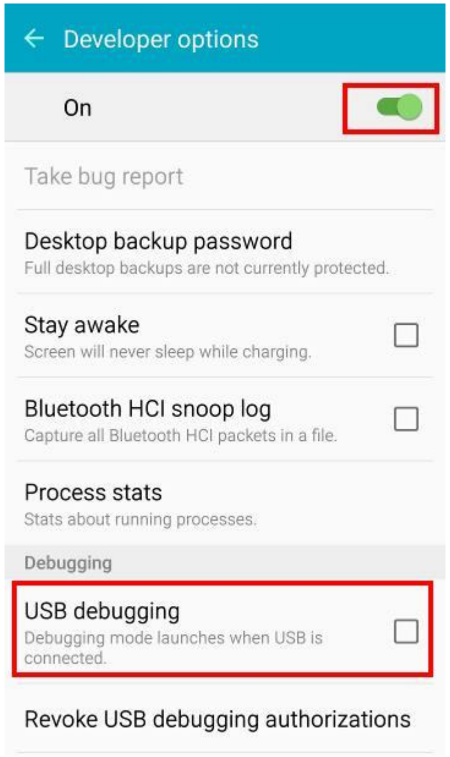
Kernel version
3.10.61-4365268
dpi@SWDD5914 #1
Fri Mar 6 18:27:10 KST 2015

Build number
LRX22G.G925IDVU1AOC5

SE for Android status
Enforcing
SEPF_SM-G925I_5.0.2_0009
Fri Mar 06 18:55:10 2015

Security software version
MDF v2.0 Release 2
VPN v1.4 Release 4

KNOX version
KNOX 2.4
Standard SDK 5.4.0
Premium SDK 2.4.0
Customization SDK 2.4.0
Container 2.4.0
CEP 2.0.1
Enterprise Billing 1.1.0
OTP 1.0.0
SE for Android 2.3.0



Developer options

On

Take bug report

Desktop backup password
Full desktop backups are not currently protected.

Stay awake
Screen will never sleep while charging.

Bluetooth HCI snoop log
Capture all Bluetooth HCI packets in a file.

Process stats
Stats about running processes.

Debugging

USB debugging
Debugging mode launches when USB is connected.

Revoke USB debugging authorizations

Platform custom code

Sometimes it may happen to need different JavaScript code, CSS stylesheets or generic assets for a specific platform

Each platform-specific subdirectory within `merges` mirrors the directory structure of the www source tree, allowing you to override or add files as needed

Example
Edit www/index.html

```
<link rel="stylesheet" type="text/css"
href="css/overrides.css" />
```

Optionally create an empty `www/css/overrides.css` (prevents missing-file error for all non-Android builds)

Create `merges/android`, then add `overrides.css`

```
body { font-size:14px; }
```

When you rebuild the project, the Android version features the custom font size, while others remain unchanged
You can also use merges to add files not present in the original www directory

Updating Cordova and Your Project

You can always update cordova to the latest version

```
$ sudo npm update -g cordova
```

Use this syntax to install a specific version

```
$ sudo npm install -g cordova@3.1.0-0.2.0
```

Run `cordova -v` to see which version is currently running

To find the latest released cordova version

```
$ npm info cordova version
```

To update platform that you're targeting

```
$ cordova platform update android --save
$ cordova platform update ios --save
```

Other useful commands

Syntax

```
$ cordova clean [<platform> [...]]
```

Cleans the build artifacts for the specified platform, or all platforms by running platform-specific build cleanup

Example

```
$ cordova clean android
```

Syntax

```
$ cordova serve [port]
```

Run a local web server for `www/` assets using specified port or default of 8000

Access projects at: `http://HOST_IP:PORT/PLATFORM/www`

https://cordova.apache.org/docs/en/latest/reference/cordova-cli/index.html