

# Introduction to PhoneGap

## Web development for mobile platforms

Manuel Rego Casasnovas <rego@igalia.com>

**Master on Free Software / August 2012**



# Outline

- 1 About PhoneGap
- 2 Starting with PhoneGap
  - Development environment
  - First PhoneGap application
  - PhoneGap API overview
  - Building PhoneGap
- 3 Example application
  - Example
  - Exercise

# What is PhoneGap?

- Mobile web development framework
- Based on **JavaScript**, **HTML5** and **CSS3**
- Open source: **Apache 2.0 License**
- Supported platforms: iPhone, Android, Windows Phone, BlackBerry, Symbian, webOS and Bada (comming soon)
- Homepage: <http://phonegap.com/>

# History

- Started by **Nitobi Software**
- People's Choice Award at O'Reilly Media's 2009 Web 2.0 Conference
- Adobe acquired Nitobi on October 2011
- PhoneGap was contributed to the **Apache Software Foundation**
  - First it was called Apache Callback
  - Renamed to **Apache Cordova**
  - URL: `http://incubator.apache.org/cordova/`

# Features

- **JavaScript API** to access different parts of the device:
  - Accelerometer
  - Camera
  - Capture
  - Compass
  - Connection
  - Contacts
  - Device
  - Events
  - File
  - Geolocation
  - Media
  - Notifications (alert, sound, vibration)
  - Storage
- **Documentation:** <http://docs.phonegap.com/>

# Advantages

- Multi-platform
- **Well-known technologies:** HTML + JavaScript + CSS
- Vast amount of **JavaScript libraries/frameworks** available:
  - General purpose: jQuery, Prototype, ...
  - Mobile: jQuery Mobile, Enyo, ...
  - ...
- Conclusion: More and more **web** is everywhere

# Eclipse

- Download **Eclipse Classic**

<http://www.eclipse.org/downloads/>

- Uncompress Eclipse

```
# cd /opt/  
# tar -xzvf \  
    /home/user/Downloads/eclipse-SDK-4.2-linux-gtk-x86_64.tar.gz  
# cd /usr/local/bin/  
# ln -s /opt/eclipse/eclipse
```

- Run Eclipse

```
$ eclipse
```

# Android SDK and ADT Plugin

- Download **Android SDK**

(<http://developer.android.com/sdk/index.html>)

```
$ cd ~
```

```
$ tar -xzvf ~/Downloads/android-sdk_r20.0.1-linux.tgz
```

- Download and install **ADT Plugin** in Eclipse

- Add site: Juno -

<http://download.eclipse.org/releases/juno>

- Add site: ADT Plugin -

<https://dl-ssl.google.com/android/eclipse/>

- Install ADT Plugin

- Android SDK Manager
- AVD Manager



# Android project with PhoneGap library I

- Download **PhoneGap** <http://phonegap.com/download>
- Create a new *Android Application Project* in Eclipse
- Create 2 new folders in the root of the project:
  - `libs/`
  - `assets/www/`
- Copy some files from downloaded PhoneGap:
  - Copy `cordova-2.0.0.js` to `assets/www/`
  - Copy `cordova-2.0.0.jar` to `libs/`
  - Copy `xml/` folder to `res/`
- Add `cordova-2.0.0.jar` to project *Build Path*

## Android project with PhoneGap library II

- Modify MainActivity to extend DroidGap

```
package com.igalia.phonegap.example.helloworldphonegap;

import org.apache.cordova.DroidGap;

import android.os.Bundle;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }

}
```

## Configure permissions

- **Modify `AndroidManifest.xml` adding the following lines between `<uses-sdk.../>` and `<application.../>`**

```
<supports-screens
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:resizeable="true"
    android:anyDensity="true" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.
    ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
```

# Basic UI

- Create a new file `index.html` inside `assets/www/` with the following content

```
<!DOCTYPE HTML>
<html>
<head>
  <title>PhoneGap Example</title>
  <script type="text/javascript" charset="utf-8" src="cordova-2.0.0.js"></
    script>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

- Run as *Android Application*

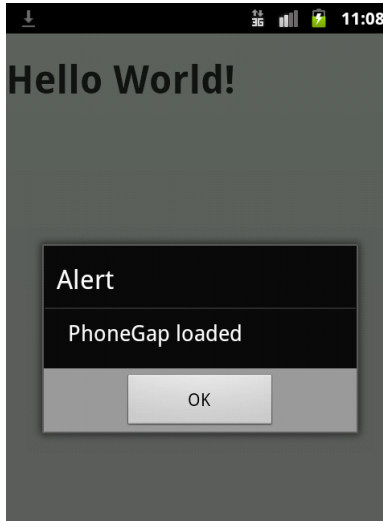
# Check that PhoneGap is ready

- Add JavaScript waiting for PhoneGap to be loaded

```
<script type="text/javascript">
  document.addEventListener("deviceready", onDeviceReady, false);

  function onDeviceReady() {
    alert ("PhoneGap_loaded");
  }
</script>
```

# Screenshot



# Notification I

## ● Alert

```
function showAlert() {  
    navigator.notification.alert(  
        "Testing_notification.alert",  
        afterAlert,  
        "Alert_title",  
        "Button"  
    );  
}  
  
function afterAlert() {  
    // Do something  
}
```

## Notification II

- Confirm

```
function showConfirm() {
    navigator.notification.confirm(
        "Do_you_like_PhoneGap?",
        onConfirm,
        "About_PhoneGap",
        "Yes,No"
    );
}

function onConfirm(buttonIndex) {
    // Do something depending on the button (index starting in 1)
}
```



# Notification III

- Beep

```
navigator.notification.beep(1);
```

- Vibrate

```
navigator.notification.vibrate(2000);
```

# Connection

## ● Internet connection

```
function checkConnection() {
    var networkState = navigator.network.connection.type;

    var states = {};
    states[Connection.UNKNOWN] = "Unknown_connection";
    states[Connection.ETHERNET] = "Ethernet_connection";
    states[Connection.WIFI] = "WiFi_connection";
    states[Connection.CELL_2G] = "Cell_2G_connection";
    states[Connection.CELL_3G] = "Cell_3G_connection";
    states[Connection.CELL_4G] = "Cell_4G_connection";
    states[Connection.NONE] = "No_network_connection";

    alert("Connection_type:_" + states[networkState]);
}
```

# Device

- Device information (name, PhoneGap version, platform, UUID, version)

```
function showDeviceInfo() {  
    alert("name:_" + device.name + "\n" +  
        "cordova:_" + device.cordova + "\n" +  
        "platform:_" + device.platform + "\n" +  
        "uuid:_" + device.uuid + "\n" +  
        "version:_" + device.version);  
}
```

# Camera

- Get picture

```
function takePicture() {
    navigator.camera.getPicture(onSuccess, onFail, {
        quality: 50,
        destinationType: Camera.DestinationType.DATA_URL
    });
}

function onSuccess(imageData) {
    var image = document.getElementById("myImage");
    image.src = "data:image/jpeg;base64," + imageData;
}

function onFail(message) {
    alert("Failed_because:_ " + message);
}
```

## Other

- Accelerometer
- Capture
- Compass
- Contacts
- Events
- File
- Geolocation
- Media
- Storage

Documentation: <http://docs.phonegap.com/>

## Building for other platforms

- Previous example is using Android as target platform
- About **other platforms**:
  - Look for documentation and build it manually
  - Use **PhoneGap Build** (<https://build.phonegap.com/>)
    - **Free for public apps**
    - Platforms: **iOS (only if iOS developer)**, Android, Windows Phone, BlackBerry, webOS and Symbian

# PhoneGap Build

- Special files:
  - *index.html*: App main page
  - *config.xml*: Basic data about your application (name, description, author, icon, permissions, ...) based on the W3C widget spec (<http://www.w3.org/TR/widgets/>)
  - *\*.png*: Icon in PNG format
- More info: <https://build.phonegap.com/docs/>

# Feed Reader

- Goal: **Develop a simple feed reader**
- Technologies:
  - PhoneGap: HTML + JavaScript + CSS
  - jQuery (<http://jquery.com/download/>)
  - jQuery Mobile (<http://jquerymobile.com/download/>)
  - jFeed (jQuery plugin): RSS/Atom feed parser  
(<https://github.com/jfhovinne/jFeed/downloads>)



# Basic page template I

- Include JavaScript and CSS for **jQuery Mobile**

```
<script type="text/javascript" charset="utf-8"  
  src="jquery-1.7.2.min.js"></script>  
<script type="text/javascript" charset="utf-8"  
  src="jquery.mobile-1.1.1.min.js"></script>  
  
<link rel="stylesheet" type="text/css"  
  href="jquery.mobile-1.1.1.min.css"></link>
```

## Basic page template II

- Page

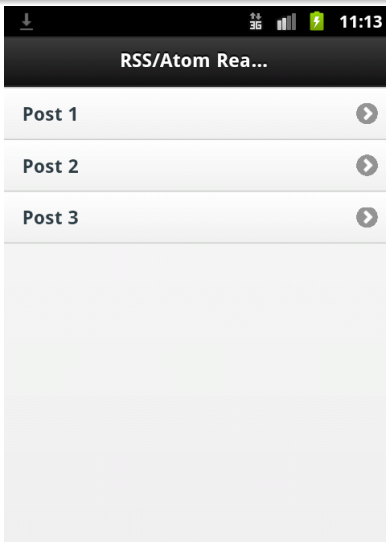
```
<div data-role="page" id="main">  
  
  <div data-role="header">  
    <h1>RSS/Atom Reader</h1>  
  </div>  
  
  <div data-role="content">  
    TODO  
  </div>  
  
</div>
```

# Basic page template III

## List

```
<div data-role="content">  
  <ul data-role="listview">  
    <li><a href="#">Post 1</a></li>  
    <li><a href="#">Post 2</a></li>  
    <li><a href="#">Post 3</a></li>  
  </ul>  
</div>
```

## Basic page template IV



# Read feed I

- Include JavaScript for **jFeed**

```
<script type="text/javascript" charset="utf-8"  
  src="jquery.jfeed.pack.js"></script>
```

## Read feed II

- Change `onDeviceReady()` function (using Planet GNOME as example)

```
function onDeviceReady() {  
    jQuery.getFeed({  
        url: "http://planet.gnome.org/rss20.xml",  
        success: function(feed) {  
            for (var i = 0; i < feed.items.length; i++) {  
                var item = feed.items[i];  
                console.log("Item_" + i + " :_" + item.title);  
            }  
        }  
    });  
}
```

## Read feed III

- Log

```
07-31 11:37:39.110: I/Web Console(1222):  
    Item 0: Richard Hughes: Being an OpenHardware  
    at file:///android_asset/www/index.html:24  
07-31 11:37:39.114: I/Web Console(1222):  
    Item 1: Beth Hadley: GUADEC 2012  
    at file:///android_asset/www/index.html:24  
07-31 11:37:39.125: I/Web Console(1222):  
    Item 2: Eitan Isaacson: GUADEC!  
    at file:///android_asset/www/index.html:24  
07-31 11:37:39.125: I/Web Console(1222):  
    Item 3: Matilda Bernard: GUADEC  
    at file:///android_asset/www/index.html:24
```

# Fill the list with the posts I

- Change HTML

```
<div data-role="content">  
  <ul data-role="listview" id="posts">  
  </ul>  
</div>
```



## Fill the list with the posts II

- Change `onDeviceReady()` function

```
function onDeviceReady(){
    jQuery.getFeed({
        url: "http://planet.gnome.org/rss20.xml",
        success: function(feed) {
            for (var i = 0; i < feed.items.length; i++) {
                var item = feed.items[i];
                appendPost(item);
            }
        }
    });
}

function appendPost(item) {
    $("#posts").append("<li>" + item.title + "</li>");
}
```

## Fill the list with the posts III



The screenshot shows a mobile application interface. At the top, there is a status bar with a download icon, 3G signal strength, battery level, and the time 12:15. Below the status bar is a dark header with the text "RSS/Atom Rea...". The main content area is a list of text entries, each representing a post from an RSS/Atom feed. The entries are:

- Richard Hughes: Being an OpenHardware Vendor
- Beth Hadley: GUADEC 2012
- Eitan Isaacson: GUADEC!
- Matilda Bernard: GUADEC
- Andre Klapper: A Monday at GUADEC 2012
- John Palmieri: GUADEC A Coruña Restaurant Suggestion
- Guillaume Desmottes: Empathy BoF at GUADEC
- Fabien Parent: GSoC - Status Report
- Udesh Liyanaarachchi: GUADEC 2012 and my Lightening Talk
- Frédéric Crozat: My hackweek8 project: dracut
- Stefano Candori: GUADEC 2012 talk
- Danielle Madeley: thanks for the sweet thoughts
- Stefano Candori: GUADEC 2012
- Giovanni Campagna: Halfway through

# Fix list rendering I

- Refresh list using `listview("refresh")`

```
function onDeviceReady() {  
    jQuery.getFeed({  
        url: "http://planet.gnome.org/rss20.xml",  
        success: function(feed) {  
            for (var i = 0; i < feed.items.length; i++) {  
                var item = feed.items[i];  
                appendPost(item);  
            }  
            $("#posts").listview("refresh");  
        }  
    });  
}
```

## Fix list rendering II

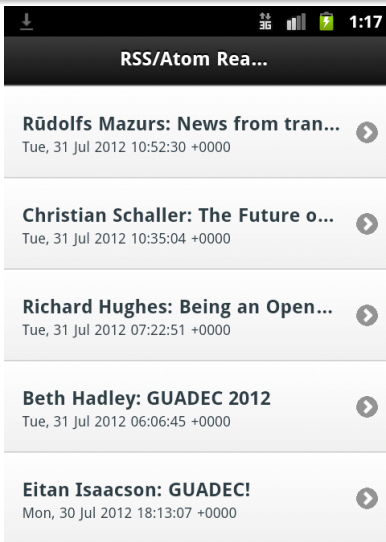


# Improving list of posts I

- **Modify** `appendPost(item)` function

```
function appendPost(item) {  
    var link = $("    $("<h3_>").append(item.title).append(link);  
    $("<p_>").append(item.updated).append(link);  
  
    var li = $("<li_>").append(link);  
  
    $("#posts").append(li);  
}
```

## Improving list of posts II



## Add option to configure feed

- Tasks **TODO**:
  - Create a new *Options* page with the form
  - Process the form with jQuery and save the URL using PhoneGap API for Storage
  - Reload the posts list with the new URL
  - Add feed title
  - Manage possible errors in the URL and request a new one again
  - Add warning if there is not Internet connection

# New page for options I

- Add link to *Options* page

```
<div data-role="header">  
  <h1>RSS/Atom Reader</h1>  
  <a href="#options" class="ui-btn-right"  
    data-icon="gear" data-iconpos="notext">Options</a>  
</div>
```

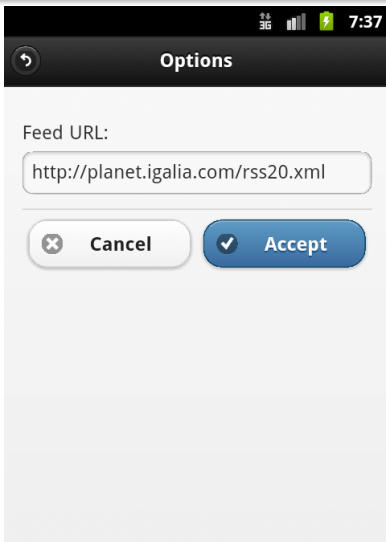


## New page for options II

- Add new *Options* page

```
<div data-role="page" id="options">
  <div data-role="header">
    <a data-rel="back" data-role="button"
      data-icon="back" data-iconpos="notext">Back</a>
    <h1>Options</h1>
  </div>
  <div data-role="content">
    <div data-role="fieldcontain">
      <label for="url">Feed URL:</label>
      <input id="url" type="text"></input>
    </div>
    <fieldset class="ui-grid-a">
      <div class="ui-block-a">
        <a href="#main" data-role="button"
          data-icon="delete">Cancel</a>
      </div>
      <div class="ui-block-b">
        <a href="#main" data-role="button"
          data-icon="check" data-theme="b"
          onClick="saveFeedURL();" >Accept</a>
      </div>
    </fieldset>
  </div>
</div>
```

## New page for options III



# Save feed URL

- Save feed URL using Storage API

```
function saveFeedURL() {  
    var url = $("#url").val();  
    window.localStorage.setItem("feedUrl", url);  
}
```

- Read the stored feed URL

```
function onDeviceReady() {  
    var url = window.localStorage.getItem("feedUrl");  
    $("#url").val(url);  
  
    jQuery.getFeed({  
        url: url,  
        success: function(feed) {  
            [...]  
        }  
    });  
}
```

## Reload posts lists I

- Move to a separate method the code inside `onDeviceReady()`

```
function onDeviceReady() {
    refreshPosts();
}

function refreshPosts() {
    var url = window.localStorage.getItem("feedUrl");
    $("#url").val(url);

    $("#posts").empty();
    jQuery.getFeed({
        url: url,
        success: function(feed) {
            for (var i = 0; i < feed.items.length; i++) {
                var item = feed.items[i];
                console.log("Item_" + i + ":_ " + item.title);
                appendPost(item);
            }
            $("#posts").listview("refresh");
        }
    });
}
```

## Reload posts lists II

- Call *refreshPosts()* from *saveFeedURL()*

```
function saveFeedURL() {  
    var url = $("#url").val();  
    window.localStorage.setItem("feedUrl", url);  
  
    refreshPosts();  
}
```

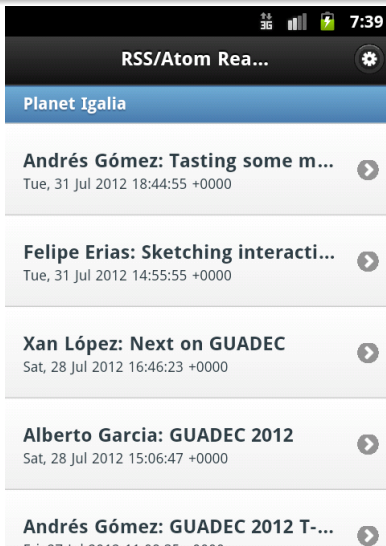
## Add feed title I

- Modify success function inside *refreshPosts()*

```
success: function(feed) {
    $("#posts").append($("<li_data-role='list-divider'>")
        .append(feed.title));

    for (var i = 0; i < feed.items.length; i++) {
        var item = feed.items[i];
        console.log("Item_" + i + ":" + item.title);
        appendPost(item);
    }
    $("#posts").listview("refresh");
}
```

# Add feed title II



# Manage errors reading feed I

- Add error function inside *refreshPosts()*

```
jQuery.getFeed({
  url: url,
  success: function(feed) {
    [...]
  },
  error: function(error) {
    var msg = "URL:_" + url + "\n" +
      error.status + ":_ " + error.statusText;
    navigator.notification.alert(
      msg,
      goToOptions,
      "Problems_reading_feed",
      "Ok"
    );
  }
});
```



## Manage errors reading feed II

- Add new function *goToOptions()*

```
function goToOptions() {  
    $.mobile.changePage("#options");  
}
```

# Manage errors reading feed III



# Check Internet connection I

- Modify *onDeviceReady()*

```
function onDeviceReady() {  
    if (isOnline()) {  
        refreshPosts();  
    } else {  
        navigator.notification.alert(  
            "Sorry_but_it_needs_Internet_to_download_the_posts",  
            null,  
            "Off-line",  
            "Ok"  
        );  
    }  
}
```

- Create new function *isOnline()*

```
function isOnline() {  
    var networkState = navigator.network.connection.type;  
    return (networkState != Connection.NONE);  
}
```

## Check Internet connection II



# References



## PhoneGap

<http://phonegap.com/>



## Apache Cordova API Documentation

<http://docs.phonegap.com/>



## PhoneGap Wiki

<http://wiki.phonegap.com/>



## PhoneGap Build Documentation

<https://build.phonegap.com/docs/>



## PhoneGap - Wikipedia, the free encyclopedia

<http://en.wikipedia.org/wiki/PhoneGap>