Developing Native JavaScript Mobile Apps Using Apache Cordova

Hazem Saleh
About Me

• Ten years of experience in Java enterprise, portal, mobile solutions.
• Apache Committer and an open source fan.
• Author of three books.
• DeveloperWorks Contributing author.
• Technical Speaker.
• Advisory Software Engineer in IBM.
Agenda

Apache Cordova Introduction

Configurations

Cordova Command Line

Cordova APIs Overview

jQuery Mobile Integration

Memo Application Demo
Apache Cordova Introduction
Apache Cordova Introduction

Device native functions examples:
Apache Cordova Introduction

## Hybrid vs Native Application

<table>
<thead>
<tr>
<th></th>
<th>Hybrid Application (Cordova)</th>
<th>Native Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be uploaded to App Store</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Technology</td>
<td>HTML + CSS + JavaScript</td>
<td>Native platform Programming Language</td>
</tr>
<tr>
<td>Cross-mobiles support</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Development Speed</td>
<td>Faster</td>
<td>Slower</td>
</tr>
<tr>
<td>Uses Device Native Features</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Cordova is supported on the following platforms:
Apache Cordova Introduction

Challenges of the current mobile development:

- Many platforms and devices.
- Different skills needed.
  - For Android: Java skills needed.
  - For iOS: Objective C skills needed.
  - For Windows: .Net skills needed.
- Different problem types.
- Huge Development and Testing Effort to have a single application on these platforms.
Who can use Cordova?

If you are a web developer and wants to develop a mobile application that can be deployed on the different app store portals.

If you are a mobile native developer and wants to develop a single application on the different mobile platforms without having to re- implement the application code on every platform.
Agenda

- Apache Cordova Introduction
- Configurations
- Cordova Command Line
- Cordova APIs Overview
- jQuery Mobile Integration
- Memo Application Demo
Configuration

Prerequisites:
- Node JS.
- Target SDK.

From command line:
- `> sudo npm install -g cordova`

To know the installed version of Cordova:
- `> cordova -v`
Agenda

- Apache Cordova Introduction
- Configurations
- Cordova Command Line
- Cordova APIs Overview
- jQuery Mobile Integration
- Memo Application Demo
Cordova Command Line

To create an application:

```
> cordova create <<app_dir>> <<project_id>> <<app_title>>
```

To add a platform (from the app folder):

```
> cordova platform add <<platform_name>>
```

To build Cordova project:

```
> cordova build
```

To deploy the app on emulator:

```
> cordova emulate <<platform_name>>
```
Hello World Demo
Agenda

- Apache Cordova Introduction
- Configurations
- Cordova Command Line
- Cordova APIs Overview
- jQuery Mobile Integration
- Memo Application Demo
Cordova APIs Overview

Native device functions are represented as plugins that can be added and removed using the command line.

Adding camera plugin example:

```bash
> cordova plugin add https://git-wip-us-apache.org/repos/asf/cordova-plugin-camera.git
```

Removing Camera plugin example:

```bash
> cordova plugin rm org.apache.cordova.core.camera
```
Device

An object that holds information about the device hardware and software.

Device information is mainly about:

- Device name.
- Device Platform.
- Device Platform version.
- Device model.

“deviceready” event is an indicator that Cordova finishes loading and Cordova APIs are ready to be called.
Camera

An object that provides an access to the default camera application.

```javascript
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);
}
```
Cordova APIs Overview

Capture

An object that provides an access to audio, image, and video capture capabilities.

```javascript
var captureSuccess = function(mediaFiles) {
    // Do something with the captured Audio media files
};

var captureError = function(error) {
    navigator.notification.alert('Error code: ' + error.code, null, 'Capture Error');
};

navigator.device.capture.captureAudio(captureSuccess, captureError, {limit:2});
```
Media

An object that allows playing back audio files on the device.

```javascript
var my_media = new Media("someFile.mp3", onSuccess, onError);

my_media.play();
function onSuccess() {
    console.log("playAudio():Audio Success");
}

function onError(error) {
    alert('code: ' + error.code + 
            'message: ' + error.message + 
    '');
}
```
Cordova APIs Overview

Notification

An object that displays visual, audible, and tactile notification.

```javascript
// Show a native looking alert
navigator.notification.alert(
    'Cordova is great!',  // message
    'Cordova',            // title
    'Ok'                  // buttonName
);

// Beep four times
navigator.notification.beep(4);

// Vibrate for 3 seconds
navigator.notification.vibrate(3000);
```
Cordova APIs Overview

Storage

Provides an access to the W3C Web Storage interface:

- Local Storage (window.localStorage).
- Session Storage (window.sessionStorage).

```javascript
window.localStorage.setItem("key", "value");
var value = window.localStorage.getItem("key");
window.localStorage.removeItem("key");
window.localStorage.clear();
```
Cordova APIs Overview

Storage

Provides an access to the device Web SQL Database (Full featured database).

```javascript
function populateDB(tx) {
  tx.executeSql('DROP TABLE IF EXISTS DEMO');
  tx.executeSql('CREATE TABLE IF NOT EXISTS DEMO (id unique, data)');
  tx.executeSql('INSERT INTO DEMO (id, data) VALUES (1, "First row")');
  tx.executeSql('INSERT INTO DEMO (id, data) VALUES (2, "Second row")');
}
function errorCB(err) {
  alert("Error processing SQL: " + err.code);
}
function successCB() {
  alert("success!");
}
var db = window.openDatabase("Demos", "1.0", "Cordova Demo", 200000);
db.transaction(populateDB, errorCB, successCB);
```
Cordova APIs Overview

Geolocation

Provides an access to location data (based on GPS sensor or inferred from Network signal).

```javascript
var onSuccess = function(position) {
    alert('Latitude: ' + position.coords.latitude + '
         Longitude: ' + position.coords.longitude + '
');
};

function onError(error) {
    alert('code: '    + error.code    + '
         message: ' + error.message + '
');
}

navigator.geolocation.getCurrentPosition(onSuccess, onError);
```
Cordova APIs Overview

More Objects:

- Accelerometer (Capture device motion)
- Compass (Get the device direction)
- Connection (Get the device connection)
- Contacts (Access to device contacts database)
- File (Access to device File system based on W3C File API)
- Globalization (Access to user locale information)
Agenda

Apache Cordova Introduction

Configurations

Cordova Command Line

Cordova APIs Overview

jQuery Mobile Integration

Memo Application Demo
jQuery Mobile Integration

jQuery Mobile is one of the most popular User Interface framework for building Mobile Web applications.

jQuery Mobile uses HTML5 + CSS3 for layout pages with minimal scripting.

It is compatible with most of the mobile and tablet browsers.
Cordova does not restrict using any specific JavaScript library but using a JavaScript library will save you a lot of time creating your own widgets from scratch.

jQuery Mobile is used in the demo application with Cordova to create the Memo application.
In order to boost the performance of jQuery mobile with Cordova, it is recommended to disable transition effects as follows (jQuery mobile 1.4):

$.mobile.defaultPageTransition = 'none';
$.mobile.defaultDialogTransition = 'none';
Agenda

Apache Cordova Introduction

Configurations

Cordova Command Line

Cordova APIs Overview

jQuery Mobile Integration

Memo Application Demo
Memo Application

GitHub project:
http://github.com/hazems/memo

Running Android project:
Questions ???