Apache Cordova overview

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/6.3/adding-native-functionality/apache-cordova-overview.html) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

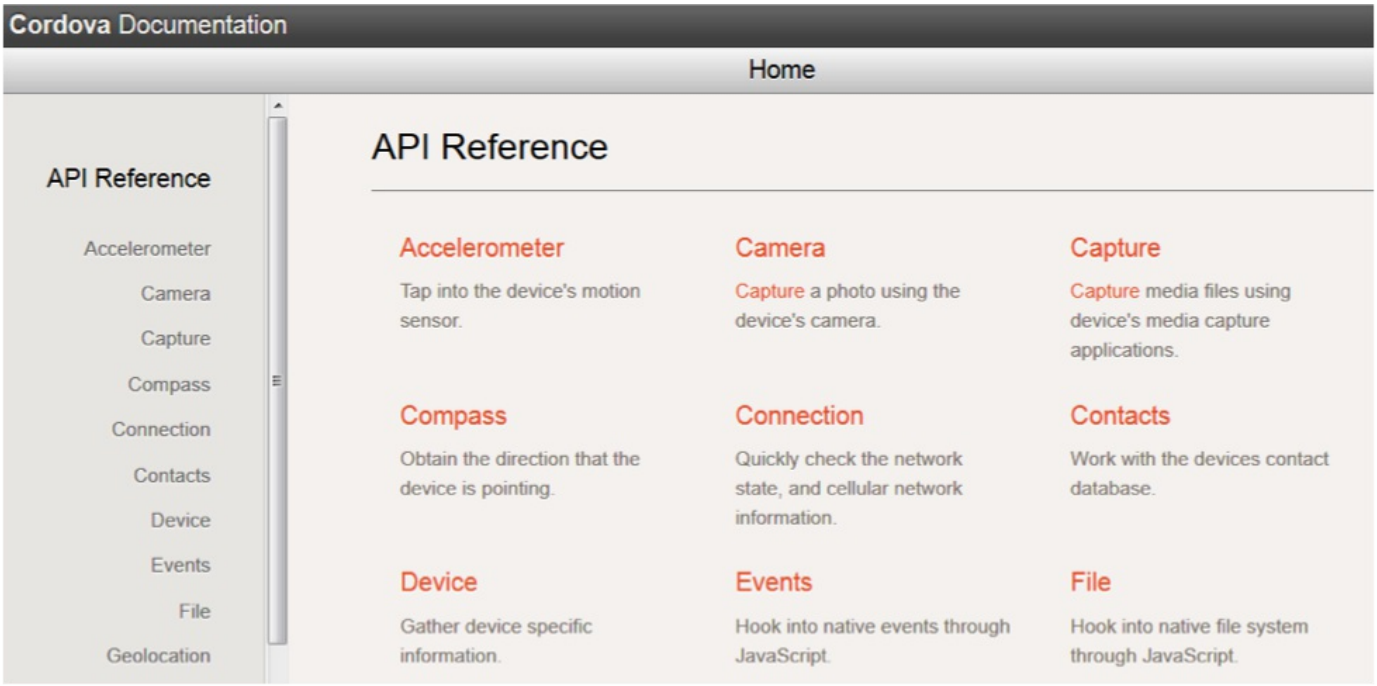
Overview

In this tutorial, an overview about Apache Cordova will be provided: what it is, how it is used in MobileFirst applications and what can be done with it.

What is Apache Cordova

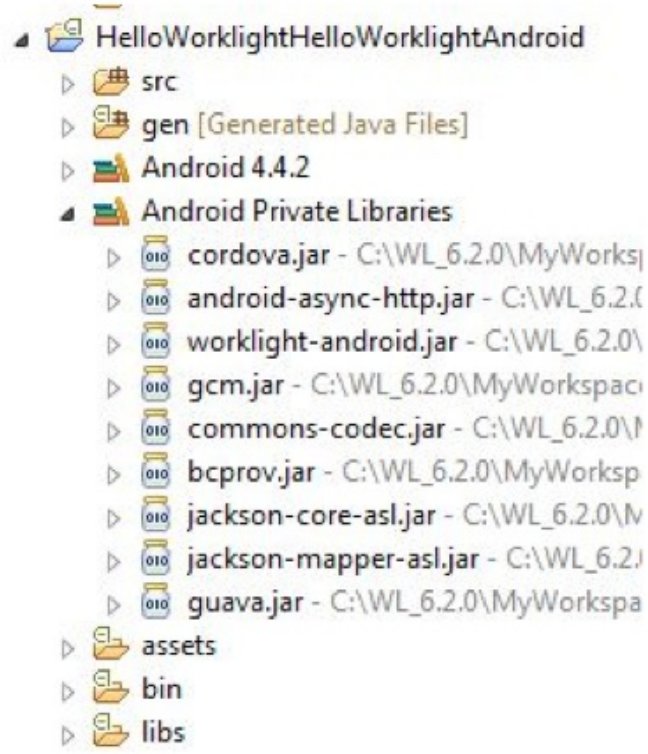
Apache Cordova is an open source framework for mobile development.
With the JavaScript API provided by Cordova, developers can access native mobile device features, and even execute native code by using JavaScript.

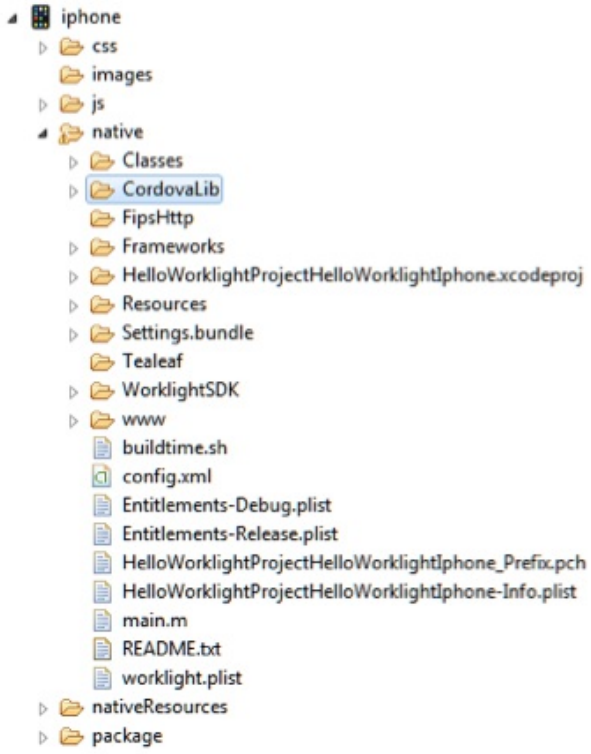
The Cordova framework is integrated into all Mobile environments in IBM MobileFirst Platform Foundation (Android, BlackBerry 10, iOS, Windows Phone 8 and Windows 8).



Apache Cordova in MobileFirst Projects

The Cordova framework is automatically added to MobileFirst projects, for example in iOS and Android:

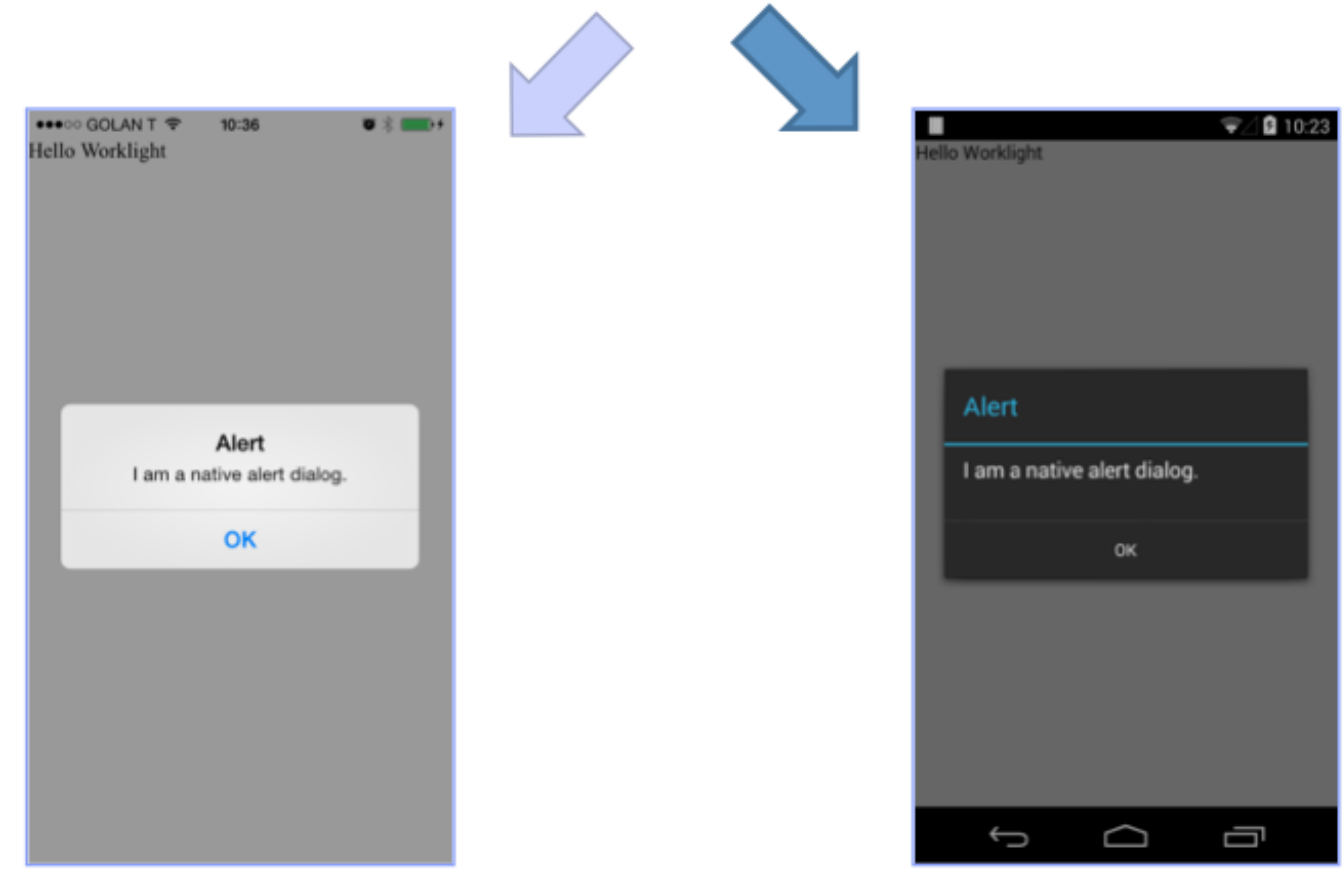




Usage samples

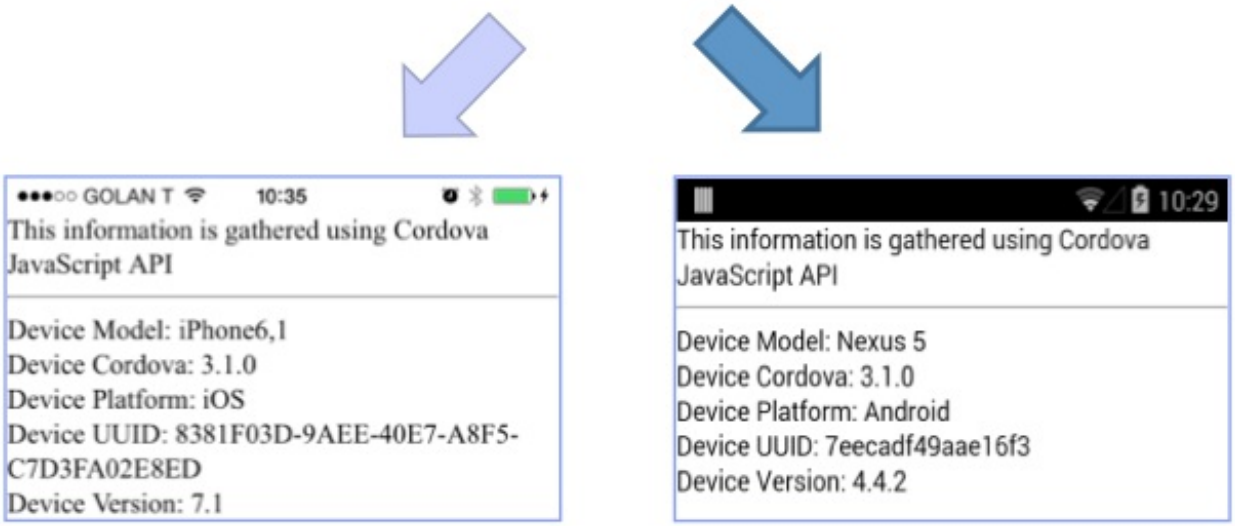
Calling a native alert notification from JavaScript on iOS and Android devices:

```
navigator.notification.alert("I am a native alert dialog.");
```



Getting device hardware and software information:

```
var element = document.getElementById('deviceProperties');
element.innerHTML = 'Device Model: '    + device.model    + '<br />' +
    'Device Cordova: '  + device.cordova  + '<br />' +
    'Device Platform: ' + device.platform + '<br />' +
    'Device UUID: '     + device.uuid     + '<br />' +
    'Device Version: '  + device.version  + '<br />';
```



For detailed documentation and samples, see:

- Apache Cordova website - http://cordova.apache.org/ (http://cordova.apache.org/)
- Apache Cordova Wiki - http://wiki.apache.org/cordova/ (http://wiki.apache.org/cordova/)

Using native pages in hybrid applications

A lot can be done with web technologies only. However, there are still cases where web technologies are not enough. In such cases Apache Cordova provides greater flexibility, allowing JavaScript access to native phone features. For example:

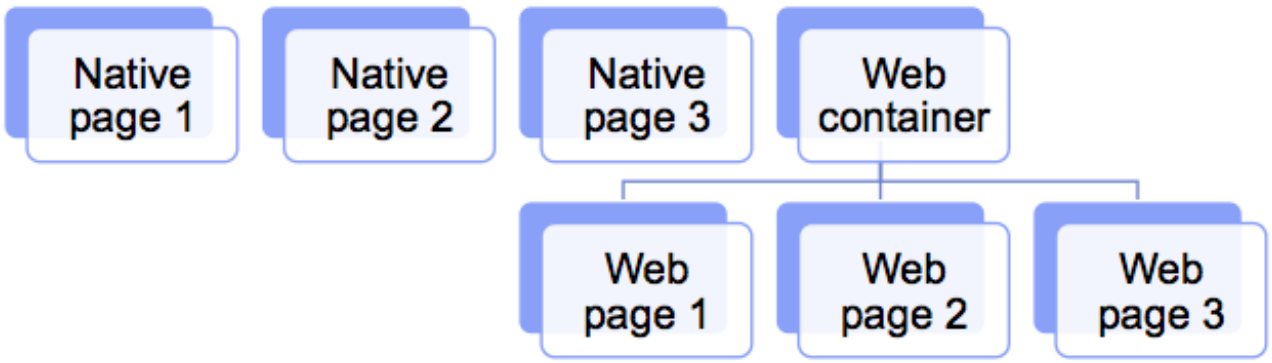
- Augmented reality
- Barcode scanner
- Opening a native contacts page

IBM MobileFirst Platform Foundation provides ways to augment your web application with native pages.
By using hybrid coding, the following actions can be performed:

- Navigate freely between web and native pages
- Share data between pages
- Share a single server session
- Implement your own functions by creating Apache Cordova plug-ins

The ability to add native pages to a web application is readily available for the iOS, Android and Windows Phone 8 platforms.
A basic scenario then is:
A developer writes one or more native pages.
A developer writes one or more web pages.

With the IBM MobileFirst Platform Foundation web API, it is possible to navigate between native and web pages, and data can be transferred back and forth.
Native API is provided to be able to communicate with the MobileFirst Server from the native page.
Cookies are shared between web and native, providing seamless client-server integration.



JavaScript API to activate native page:

```
WL.NativePage.show(className, callback, data);
```

The web page invokes the native class `className`, with the parameter `data`, and calls the callback function on return. Example:

```
WL.NativePage.show("com.demo.NativePage", function(data) {demoApplication.returnFromNative(data)
;},{inputData: "Hello"});
```

The following tutorials explain how to integrate native pages in hybrid applications.

Extending hybrid functionality with Cordova plug-ins

In some cases, a native page is not required, but native actions are still required to be performed using native code.

Examples might be:

- Retrieving device data that is not available on a JavaScript level, for example, certificates from a Java KeyStore
- Using a native third-party library that is embedded in the application
- Performing complex calculation or encryption that takes too much time in JavaScript

The solution for such UI-less native functionality is to implement a Cordova plug-in.
The following tutorials explain how to add native functionality to hybrid applications.