Apache Cordova overview

Overview

This tutorial provides an overview of what you can do with Apache Cordova in your MobileFirst applications.

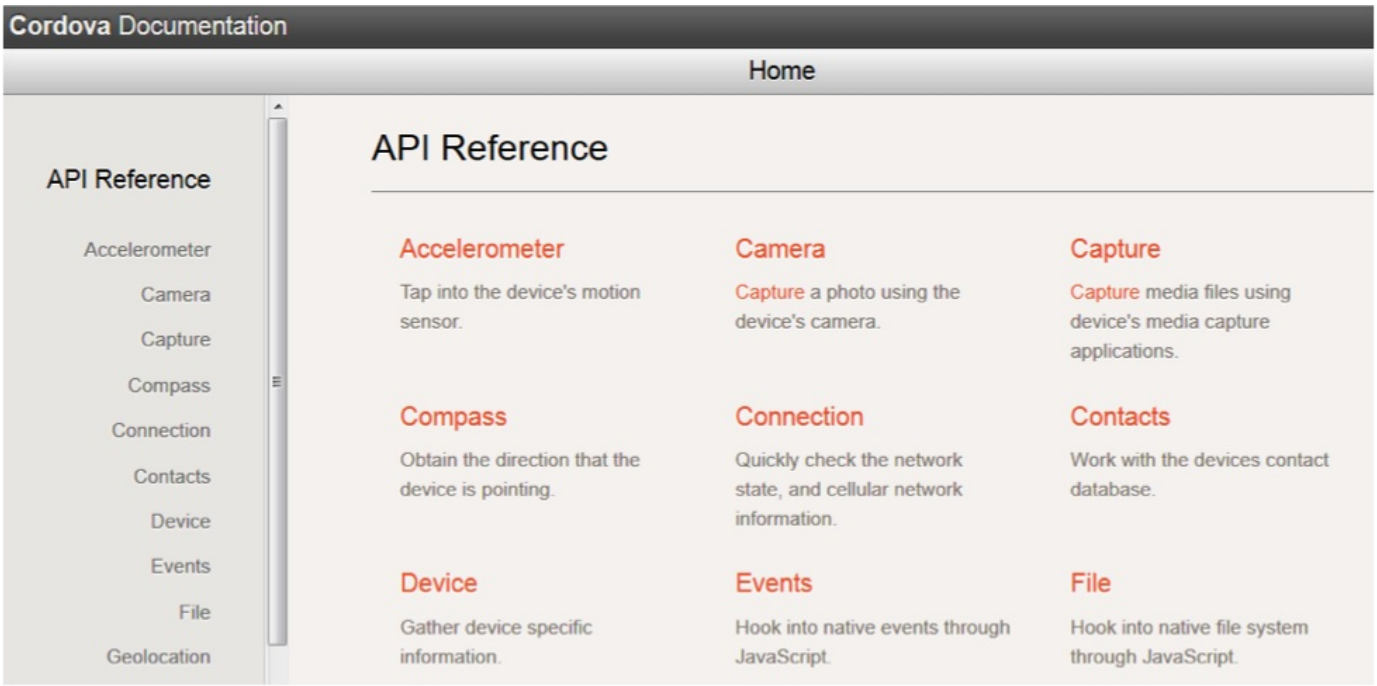
This tutorial covers the following topics:

- What is Apache Cordova?
- Apache Cordova in MobileFirst projects
- Usage samples
- Using native pages in hybrid applications
- Extending hybrid functionality with Cordova plug-ins

What is Apache Cordova?

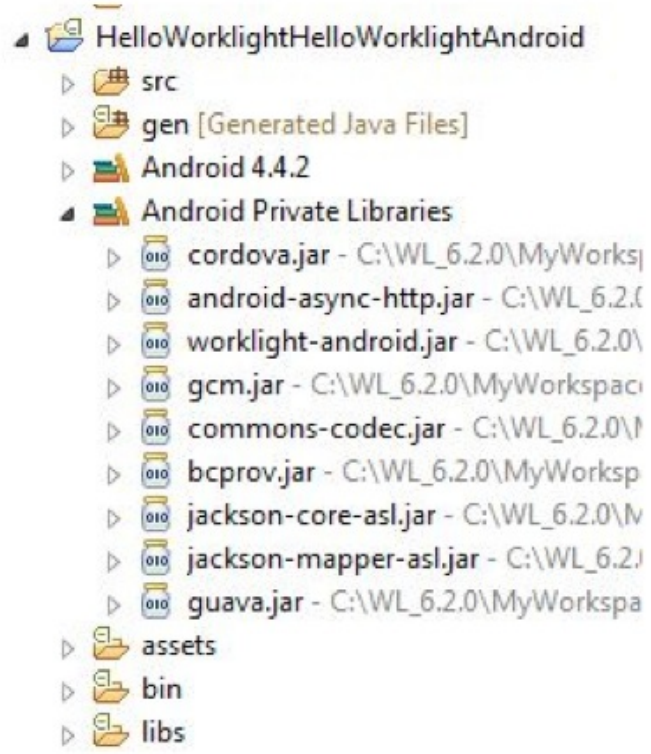
Apache Cordova is an open source framework for mobile development.
With the JavaScript API provided by Cordova, developers can access native mobile device features and even execute native code by using JavaScript.

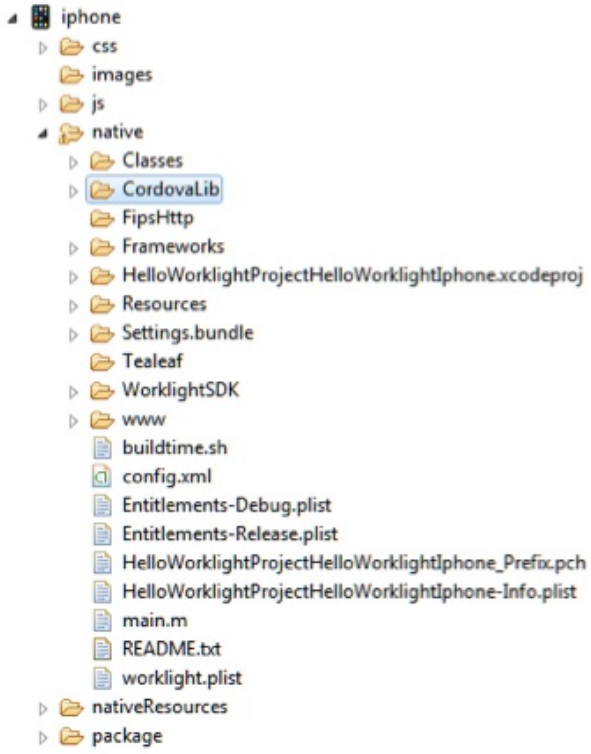
The Cordova framework is integrated into all Mobile environments in IBM MobileFirst Platform Foundation (Android, BlackBerry 10, iOS, Windows Phone 8, and Windows 8).



Apache Cordova in MobileFirst projects

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

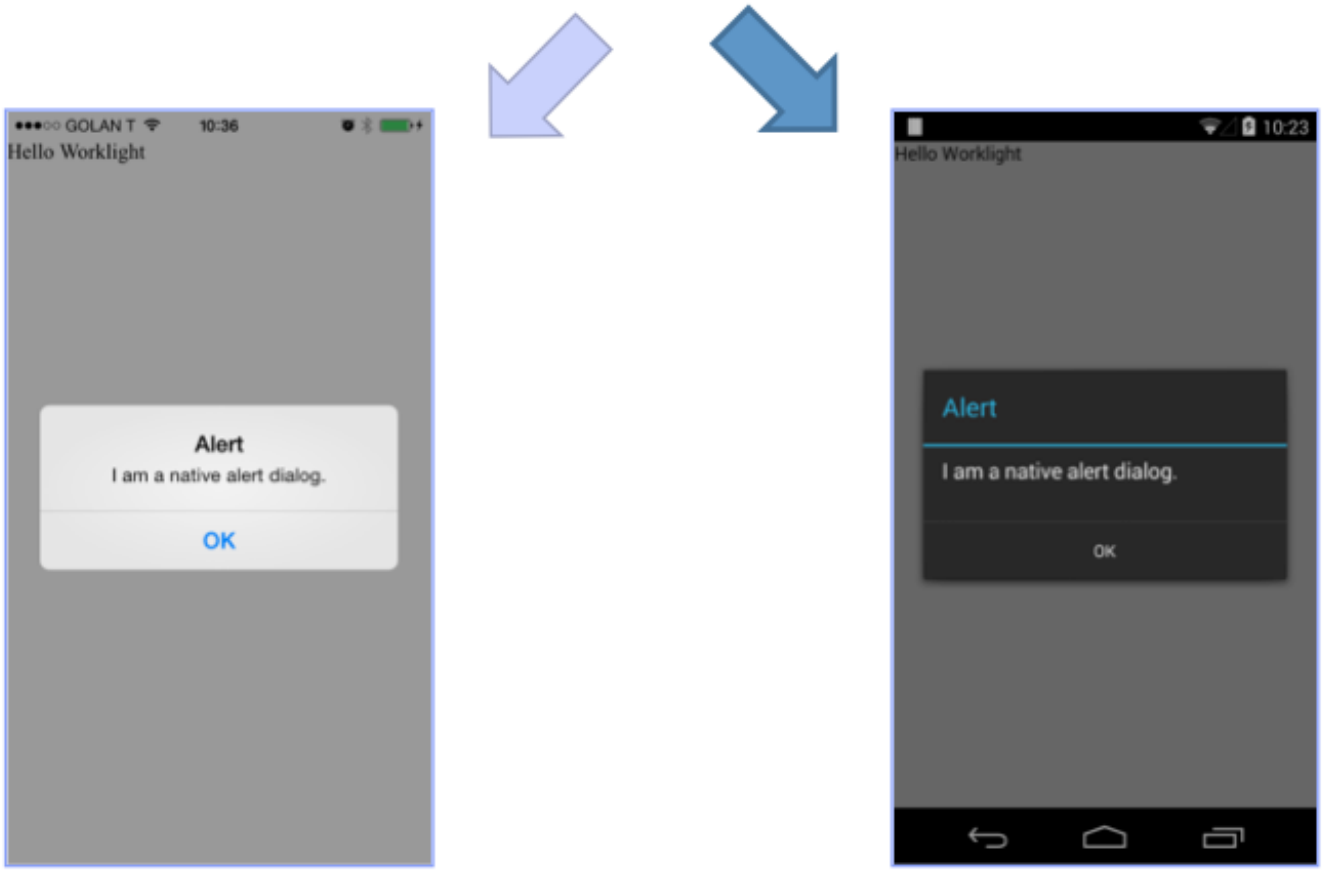




Usage examples

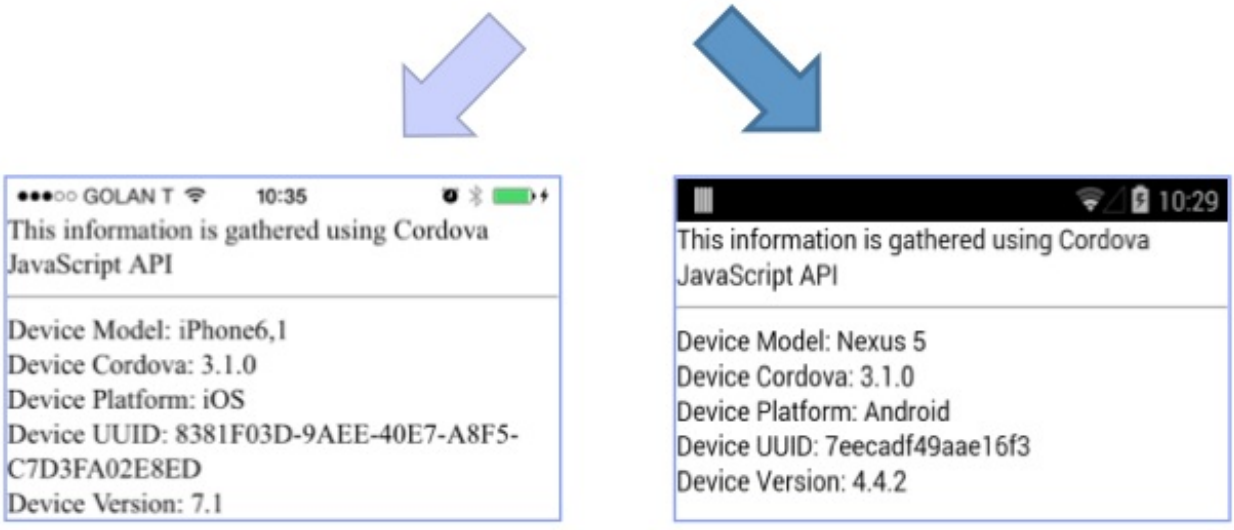
Calling a native alert notification from JavaScript on iOS and Android devices

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



Getting device hardware and software information

```
var element = document.getElementById('deviceProperties');<br />
    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/)
- Apache Cordova Wiki (http://wiki.apache.org/cordova/)

Using native pages in hybrid applications

You can do a lot with web technologies alone. 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

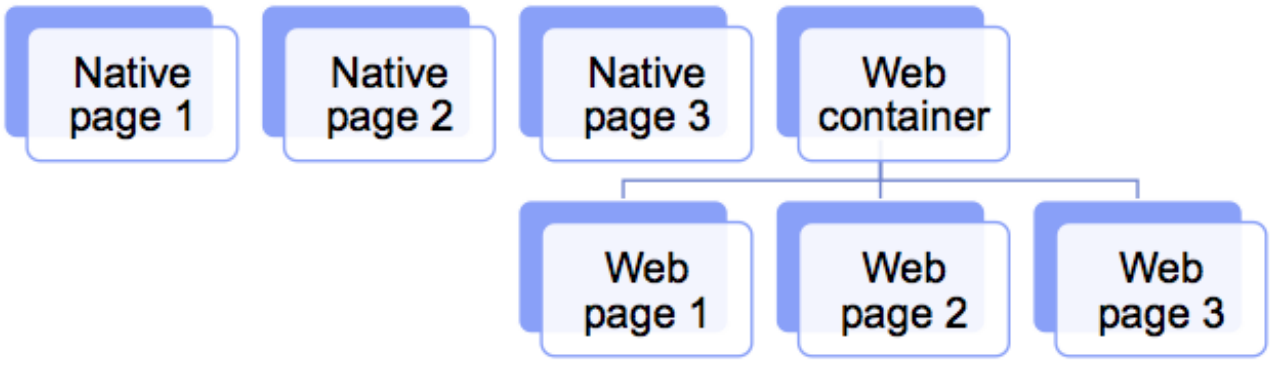
IBM MobileFirst Platform Foundation provides ways to augment your web application with native pages.
By using hybrid coding, you can perform the following actions:

- 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:

1. A developer writes one or more native pages.
2. A developer writes one or more web pages.

With the MobileFirst 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 enable communication with 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"
   }
);
```

Extending hybrid functionality with Cordova plug-ins

In some cases, a native page is not required, but native actions still need native code for their execution.

Examples:

- Retrieving device data that is not available on a JavaScript level, such as 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 native functionality that does not require a user interface is to implement a Cordova plug-in.

The tutorials in the Adding native functionality to hybrid applications (../../adding-native-functionality/) category explain how to integrate native pages in hybrid applications.