MobileFirst Platform Foundation development in Cordova applications

Overview

From http://cordova.apache.org/ (http://cordova.apache.org/):

> Apache Cordova is an open-source mobile development framework. It allows you to use standard web technologies such as HTML5, CSS3, and JavaScript for cross-platform development, avoiding each mobile platforms' native development language. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's sensors, data, and network status.

IBM MobileFirst Platform Foundation provides an SDK in the form of several Cordova plug-ins. Learn how to Add the MobileFirst Platform Foundation SDK to Cordova applications (../../adding-the-mfpf-sdk/cordova).

The MobileFirst SDK feature set provides the following:

- Use MobileFirst `WLResourceRequest` API to retrieve data from backend systems.
- Protect applications using the MobileFirst security framework and other security features such as Application Authentcity Protection, Remote Disble.
- Ability to navigate and share data between web and native views and/or call Native code using the MobileFirst `SendAction` API.
- Update an application's web resources using Direct Update.
- Implement Java or JavaScript adapters.

> For a complete list of MobileFirst SDK abilities visit the user documentation (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html).

Jump to:

Cordova application development

Applications developed with Cordova can be further enhanced by using the following Cordova-provided development paths and features:

Hooks

Cordova Hooks are scripts that provide developers with the ability to customize Cordova commands, enabling to create for example custom build flows. Read more about Cordova Hooks (http://cordova.apache.org/docs/en/dev/guide/appdev/hooks/index.html#Hooks%20Guide).

Merges

The Merges folder provides the ability to have platform-specific web resources (HTML, CSS and JavaScript files). These web resources are then deployed during the `cordova prepare` step to the appropriate native directory. Files placed under the **merges**/ folder will override matching files in the **www**/ folder of the relevant platform. Read more about the Merges folder (https://github.com/apache/cordova-cli#merges).

Cordova plug-ins

Using Cordova plug-ins can provide enhancements such as adding native UI elements (dialogs, tabbars, spinners and the like), as well as more advanced functionalities such as Mapping and Geolocation, loading of external content, custom keyboards, Device integration (camera, contacts, sensors, and so on).

You can find Cordova plug-ins on GitHub.com (https://github.com) and in popular Cordova plug-ins websites, such as Plugreg (http://plugreg.com/) and NPM (http://npmjs.org).

Example plug-ins:

- cordova-plugin-dialogs (https://www.npmjs.com/package/cordova-plugin-dialogs)
- cordova-plug-inprogress-indicator (https://www.npmjs.com/package/cordova-plugin-progress-indicator)
- cordova-plugin-statusbar (https://www.npmjs.com/package/cordova-plugin-statusbar)

3rd-party frameworks

Cordova application development can be further enhanced by using frameworks such as Ionic (http://ionicframework.com/), AngularJS (https://angularjs.org/), jQuery Mobile (http://jquerymobile.com/), Backbone (http://backbonejs.org/) and many others.

3rd-party packages

Applications can be modified using 3rd party packages to achieve requirements such as Minification & Concatenation of the application's web resources and more. Popular packages to do so are:

- uglify-js (https://www.npmjs.com/package/uglify-js)
- clean-css (https://www.npmjs.com/package/clean-css)

Previewing applications

A Cordova application's web resources can be previewed either in the iOS Simulator, Android Emulator, Windows Emulator or physical devices. In MobileFirst Platform Foundation, two additional live-preview options are available: IBM Mobile Browser Simulator and Simple Browser rendering.

- With Simple Browser rendering the application is presented as a web page in the desktop browser.
- The Mobile Browser Simulator is a web application that enables testing the application by simulating device features without needing to install device vendor native SDK.

> Learn more about Simple Browser rendering the IBM Mobile Browser Simulator in the user documentation.

Previewing the application web resources:

1. From a **Command-line** window, run the command:

```
mfpdev app preview
```

2. Select a preview option:

```
? Select how to preview your app: (Use arrow keys)
❯ browser: Simple browser rendering
mbs: Mobile Browser Simulator
```

3. Select a platform to preview (only added platform will be displayed):

```
❯◯ android
◯ ios
```

> **Tip:** Learn more about the various CLI commands in the Using CLI to manage MobileFirst artifacts (../using-mobileirst-developer-cli-to-manage-mobilefirst-artifacts/) tutorial.

Live preview

Applicative code (HTML, CSS and JS) can now be edited in real-time with live-preview.
After making a change to a resource, save the change and it will be immediately reflected in the browser.

Live reload

To achieve a similar effect while previewing in physical devices or simulators/emulators, add the **cordova-plugin-livereload** plug-in. For usage instructions, see the plug-ins GitHub page (https://github.com/omefire/cordova-plugin-livereload).

CrossWalk support

Cordova applications for the Android platform can have their default WebView replaced with the CrossWalk WebView (https://crosswalk-project.org/).
To add it:

1. From a **Command-line** line, run the command:

```
cordova plugin add cordova-plugin-crosswalk-webview
```

   This command will add the CrossWalk WebView to the application.

   Behind the scenes, the MobileFirst Cordova SDK will adjust the Android project activity for using it.

2. Build the project by running the command:

```
cordova build
```

Running the application on emulator or on a physical device

To emulate the application execute the Cordova CLI command `cordova emulate ios|android|windows`. For example:

```
cordova emulate ios
```

To run the application on a physical device, attached to the development workstation a run the Cordova CLI command `cordova run ios|android|windows`. For example:

```
cordova run ios
```

Further reading

Learn more about Cordova:

- Cordova Overview (https://cordova.apache.org/docs/en/latest/guide/overview/index.html)
- Cordova best practices, testing, debugging, cosiderations and keeping up (https://cordova.apache.org/docs/en/latest/guide/next/index.html#link-testing-on-a-simulator-vs-on-a-real-device)
- Get started with Cordova applications development (https://cordova.apache.org/#getstarted)

Tutorials to follow next

Get started by adding the MobileFirst SDK to your Cordova application (../../adding-the-mfpf-sdk/cordova), and review MobileFirst-provided features in the All Tutorials (../../all-tutorials/) section.