

Lab 3 - Load MFP framework & application

Bootstrapping

Labs 1 & 2 introduced the IBM Employee App and demonstrated how to add device platforms and the MobileFirst 8.0 Cordova plugin to the app. To enable the MobileFirst capabilities requires the addition of a few lines of code.

This lab will demonstrate how to add MFP JavaScript code to the Ionic application source and launch the new application in an emulator. This lab will also demonstrate using the Mobile Browser Simulator (MBS), one of the components provided with the MobileFirst, to preview the application. In contrast to the `cordova emulate` command, the `mfpdev app preview` command does not require a platform emulator (such as Android Virtual Device) - it operates entirely in the browser. The simulator also enables the use of developer tools such as the debugger, style editor and inspector.

The Mobile Browser Simulator requires a MobileFirst development server to operate.

Note: For this lab there are snippets files included in the `/snippets` folder of your root workspace which can be used to quickly copy/paste the large source code blocks below.

Steps

1. Open the **IBMEmployeeApp** project using Brackets.

Note: You may use any IDE you like to perform the labs. The examples shown use the Brackets IDE.

2. Open the `app.js` file in the folder `IBMEmployeeApp/www/js` and add the following code immediately after the declaration of `ibmApp` and just after the `//Add support for Cordova` and before the comment `//application config`:

```

6  var ibmApp = angular.module('ibmApp', ['ionic'])
7  // Add support for Cordova.
8  // application config.
9  ▼ ibmApp.config(function ($stateProvider, $urlRouterProvider, $ionicConfigProvider) {
10     // $urlRouterProvider - letting us specifsy the default route when loading the module
11     $urlRouterProvider.otherwise('/')
12     $stateProvider
13     ▼ .state('main', {
14         url: '/main',
15         templateUrl: 'partials/employee.html',
16         controller: 'mainCtrl',
17     ▼ resolve: {
18     ▼     employees: function (EmployeeService) {
19         return EmployeeService.getEmployeeList();
20     }
21     }
22 })
23
24 ▼ .state('splash', {
25     url: '/',
26     /* default url */
27     templateUrl: 'pages/splash.html',
28     controller: 'splashCtrl'
29 })
30
31 ▼ .state('detail', {
32     url: '/detail/:empId',
33     templateUrl: 'partials/details.html',
34     controller: 'employeeDetailCtrl',
35     ▼ resolve: {
36     ▼     employeeDetailList: function ($stateParams, EmployeeDetailsService) {
37         return EmployeeDetailsService.getEmployeeDetails($stateParams.empId);
38     },
39     ▼     empId: function ($stateParams) {
40         return $stateParams.empId;
41     }
42     }
43 })
44 }) // end of app config.
45 // Add MobileFirst configuration stuff.
46

```

```
//Adding support for cordova.
ibmApp.run(function($ionicPlatform) {
    console.log('>> ibmApp.run ...');
    $ionicPlatform.ready(function() {
        // Hide the accessory bar by default (remove this to show the access
ory bar above the keyboard
        // for form inputs)
        console.log('>> ibmApp.ready ...');
        if (window.cordova &&
            window.cordova.plugins &&
            window.cordova.plugins.Keyboard) {
            cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
        }
        if(window.StatusBar) {
            StatusBar.styleDefault();
        }
    });
});
```

```
6  var ibmApp = angular.module('ibmApp', ['ionic'])
7  // Add support for Cordova.
8  ibmApp.run(function($ionicPlatform) {
9      console.log('>> ibmApp.run ...');
10     $ionicPlatform.ready(function() {
11         // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
12         // for form inputs)
13         console.log('>> ibmApp.ready ...');
14         if (window.cordova &&
15             window.cordova.plugins &&
16             window.cordova.plugins.Keyboard) {
17             cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
18         }
19         if(window.StatusBar) {
20             StatusBar.styleDefault();
21         }
22     });
23 });
24 // application config.
```

Tip: Code snippets for this lab are available in the `snippets/lab3.js` file.

3. Within `app.js` scroll to the very bottom of the file and add the following code just after the `"// Add MobileFirst configuration stuff."` comment (locate snippet for copy/paste in the `/snippets` folder):

```
// Add MobileFirst configuration stuff.
var Messages = {
    // Add here your messages for the default language.
    // Generate a similar file with a language suffix containing the translated messages.
    // key1 : message1,
};

var wlInitOptions = {
    // Options to initialize with the WL.Client object.
    // For initialization options please refer to IBM MobileFirst Platform Foundation Knowledge Center.
};

function wlCommonInit() {
    console.log(">> wlCommonInit() ..." );
    var serverUrl = WL.App.getServerUrl(function(success){
        console.log(success);
    }, function(fail){
        console.log(fail);
    })
}
```

```
62 // Add MobileFirst configuration stuff.
63
64 ▼ var Messages = {
65     // Add here your messages for the default language.
66     // Generate a similar file with a language suffix containing the translated messages.
67     // key1 : message1,
68 };
69
70 ▼ var wlInitOptions = {
71     // Options to initialize with the WL.Client object.
72     // For initialization options please refer to IBM MobileFirst Platform Foundation Knowledge Center.
73 };
74
75 ▼ function wlCommonInit() {
76     console.log(">> wlCommonInit() ..." );
77     var serverUrl = WL.App.getServerUrl(function(success){
78         console.log(success);
79     }, function(fail){
80         console.log(fail);
81     })
82 }
```

4. Save the app.js file

Preview the application

There are two options available to preview the application :

- Use the **cordova emulate** command

- The emulate command launches an android virtual device or Xcode iOS simulator
- Use the **mfpdev app preview** command.
 - The preview command provides options of:
 - Simple browser rendering or
 - Mobile Browser Simulator rendering

The **mfpdev app preview** command provides easy preview, testing, and debugging of applications using the embedded browser debugger. The Mobile Browser Simulator feature also supports many Cordova device emulation controls for items such as GPS and accelerometer. However, this requires that the MobileFirst server is running, make sure you start the MFP server using the `./run.sh` command

We going to preview the application using the **mfpdev app preview** first.

5. **Run** the application in the browser we can run the application using the *lighting* button within Brackets or use the following command and select the first option **browser: Simple browser rendering**

```
mfpdev app preview
```

6. Use the `cordova emulate` command which will allow you to choose between the platform you choose to add: android or iOS.

Note: that iOS will only be available if you are working from a machine running OS X.

```
cordova emulate
```

If you added more than one device platform to Cordova, then you can select a specific device. Otherwise the default device emulator should start.

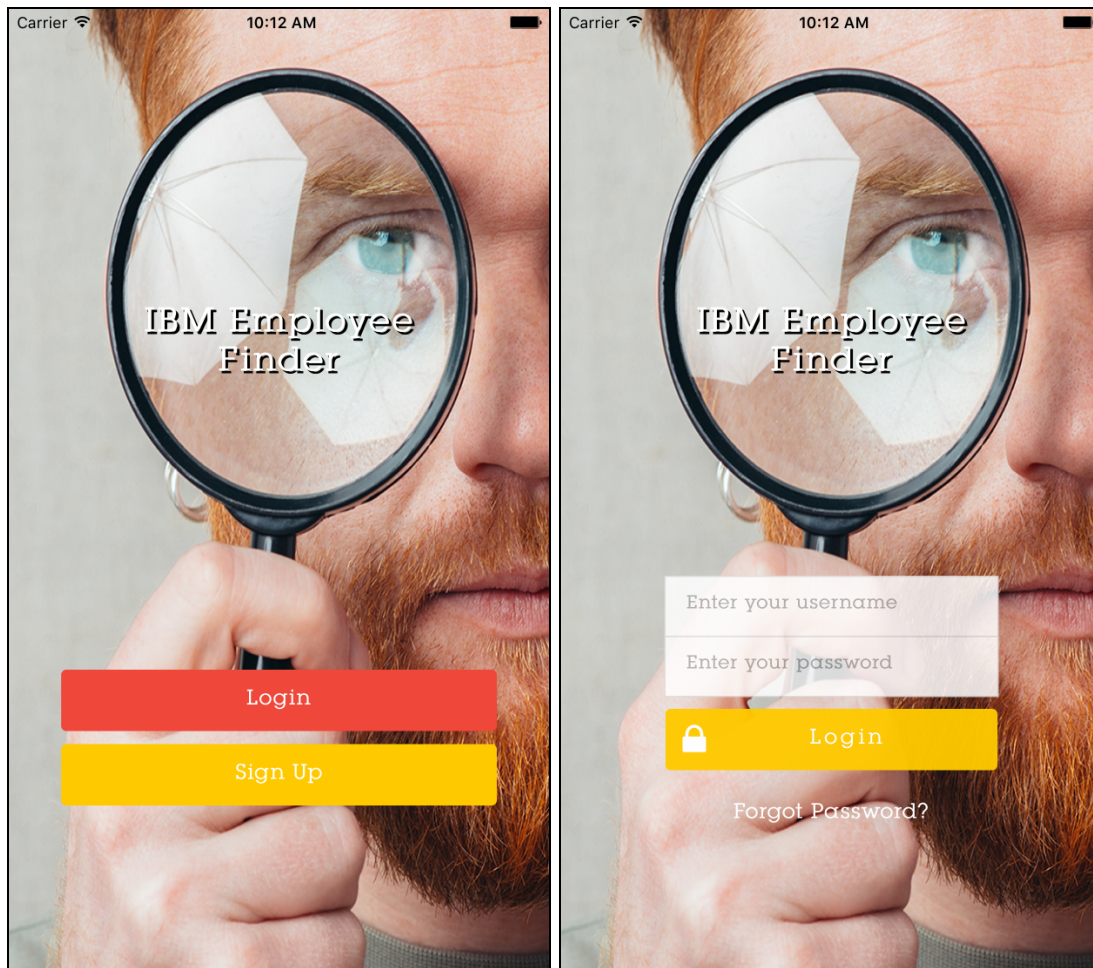
Note: If you are running on the provided VM, you will only have the **Nexus_5_API_22_x86** device available.

```
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$ cordova emulate
Running command: /Users/eliran_pro/Documents/projects/Madrid2016/IBMEmployeeApp/hooks/after_prepare/010_add_platform_class.js /User
s/eliran_pro/Documents/projects/Madrid2016/IBMEmployeeApp
add to body class: platform-ios
Building project : /Users/eliran_pro/Documents/projects/Madrid2016/IBMEmployeeApp/platforms/ios/Employee.xcodeproj
  Configuration : Debug
  Platform      : emulator

** BUILD SUCCEEDED **

No target specified for emulator. Deploying to iPhone-6s-Plus, 9.2 simulator
com.ionicframework.ibmemployeeapp420875: 47457
logPath: /Users/eliran_pro/Documents/projects/Madrid2016/IBMEmployeeApp/platforms/ios/cordova/console.log
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$
```

1. In a few moments, the application will start up in an iOS emulator window.



2. Close the Emulator when finished.

Summary

This lab demonstrated the MobileFirst project to use the services provided by the platform by adding code to the main app.js file.

If you were unable to complete this lab, you can catch up by running this command:

```
git checkout -f step-3
```