Lab 4 - Register and Manage Application

The MobileFirst platform provides a web based console to easily view and manage application instances, platforms and versions. Managing an application requires an application registration. Registering an application will also enable secure connectivity to the MobileFirst server and will enable the use of the adapter framework and analytics capabilities of the MFP server.

Steps

1. Open the **IBMEmployeeApp** project using a preferred IDE. Code snippets are available from the `snippets/lab4.js` file.

2. Open the `app.js` file in the folder `IBMEmployeeApp/www/js` locate the **wlCommonInit()** function and add the following code immediately after the last line of the code inside

```
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     }
```

```
//Calling to the MobileFirst Server
WLAuthorizationManager.obtainAccessToken().then(
        function (accessToken) {
          console.log(">> Success - Connected to MobileFirst Server");
        },
        function (error) {
          console.log(">> Failed to connect to MobileFirst Server");
        }
);
```

Your code should look like this:

```
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
83        //Calling to the MobileFirst Server
84        WLAuthorizationManager.obtainAccessToken().then(
85 ▼          function (accessToken) {
86              console.log(">> Success - Connected to MobileFirst Server");
87          },
88 ▼          function (error) {
89              console.log(">> Failed to connect to MobileFirst Server");
90          }
91        );
92
93
94    };
95
```
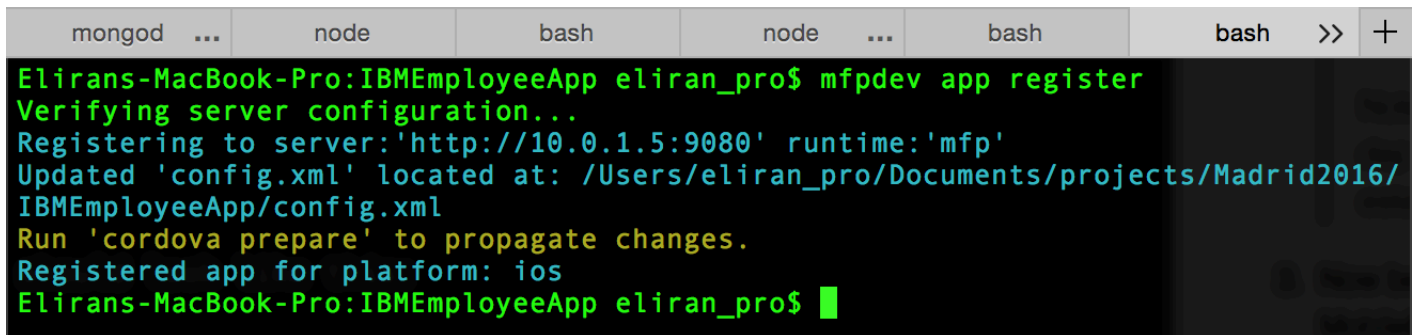
> **Note:** That the `WLAuthorizationManager.obtainAccessToken()` method the obtains an
> access token for the specified resource scope from the MobileFirst authorization server. The
> **WLAuthorizationManager** class is instantiated as a singleton and can be used anywhere in the
> application to obtain the client ID and authorization header. It is also called internally to handle
> MobileFirst OAuth-based security model protocol.
>
> **Important:** That **WLAuthorizationManager.obtainAccessToken()** method replaces the deprecated
> **WL.Client.connect()** method that has been used in earlier version of MFP to establishes a
> connection to the MobileFirst

3. Next register the application with the MobileFirst Server, **Run** the following command

   `mfpdev app register`

```
  mongod   ...      node          bash          node     ...      bash              bash      ≫  +
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$ mfpdev app register
Verifying server configuration...
Registering to server:'http://10.0.1.5:9080' runtime:'mfp'
Updated 'config.xml' located at: /Users/eliran_pro/Documents/projects/Madrid2016/
IBMEmployeeApp/config.xml
Run 'cordova prepare' to propagate changes.
Registered app for platform: ios
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$
```

4. The Operations Console is a browser-based interface through which you can manage artifacts deployed
   to your server. View the **Operations Console** by running `mfpdev server console`

```
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$ mfpdev server console
Opening console for server 'mydevserver'
Elirans-MacBook-Pro:IBMEmployeeApp eliran_pro$
```

The MobileFirst Operations Console will start in your default browser, you can see the application 'Employee' under the "Application" list on the dashboard left side.
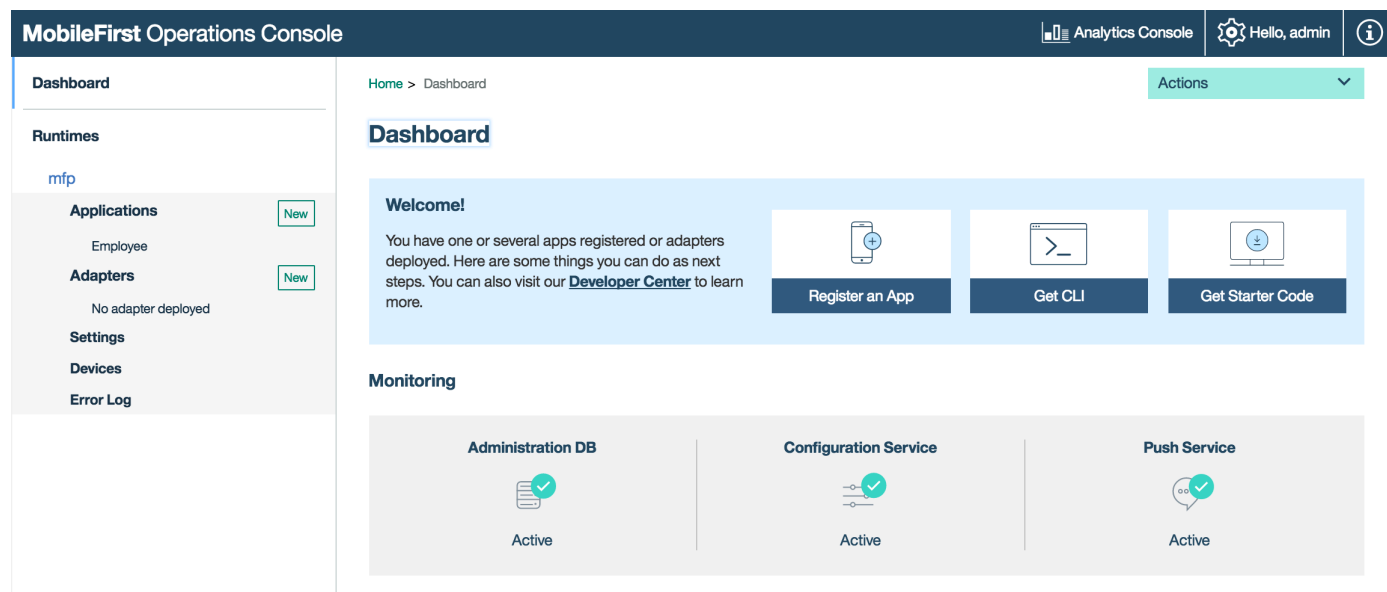
5. Use the cordova emulate command to preview the application on the emulator. `cordova emulate`

6. Check the console logs. While the iOS simulator is selected press the Simulator's debug menu and select the "Open console logs option"
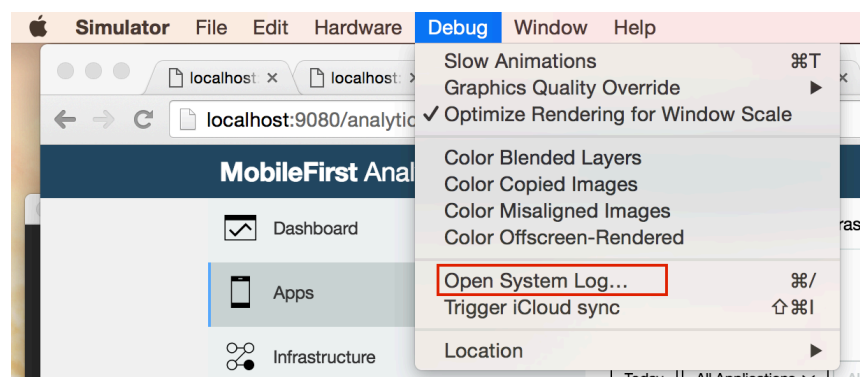
Then in the search box on the top right corner enter ">>" you should be able to see the connection success message

Notice the application has successfully connected to the MFP server.

7. Delete the ">>" and enter "Response Content" you will see that the server responded with "access_token"

8. Close the simulator and go back to the web console, lets take a look at the built-in management capabilities of the platform, you can also enter the following url in your browser
http://localhost:9080/mfpconsole/index.html#/dashboard

9. **Press** on the "Employee" application under the "Application" menu on the left side, you will see the Employee Management feature.

10. **Select** "Access Disable" option

11. In the "URL of latest version" input filed simply enter "http://www.cnn.com"

12. In the "Default notification message*" enter "Application disabled, please download latest version"

13. Press the "Save button"

Employee
iOS v 0.0.1 | com.ionicframework.ibmemployeeapp420875

✓ The 'com.ionicframework.ibmemployeeapp420875' application version 0.0.1 in iOS is disabled.   ✕

Management   Authenticity   Security   Log Filters   Configuration Files

Last modified: Apr 9, 2016, 11:29 AM

Application Access

Status: *

○ Active   ○ Active and Notifying   ● Access Disabled

URL of latest version

| http://www.cnn.com |

Specify a URL for the new version of the application (in the appropriate public or private app store).

Default notification message *

| Application disabled, please download latest |

The message that the user receives when opening the application.
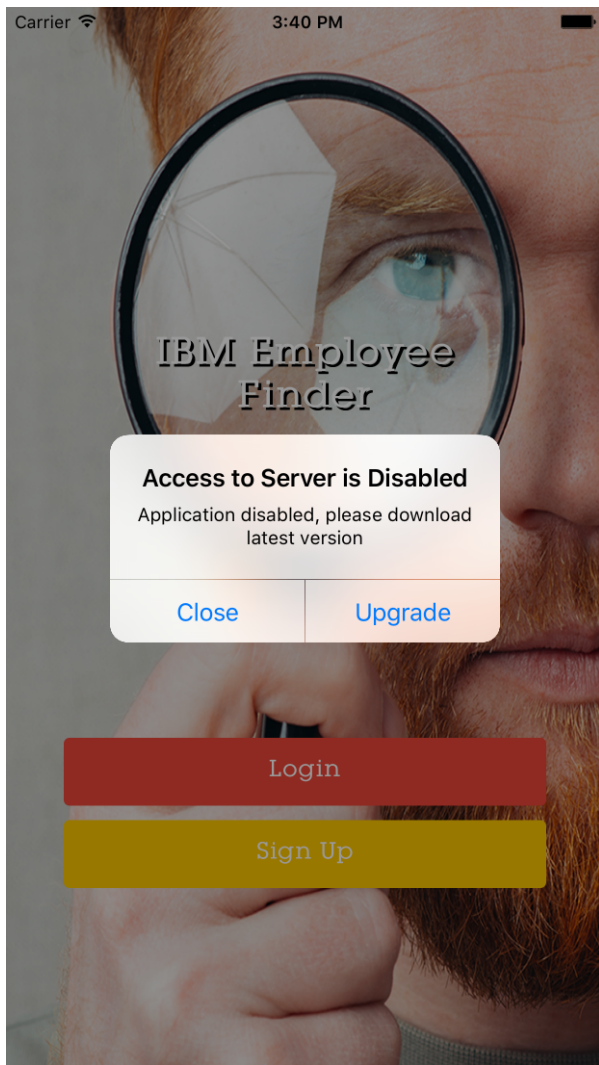
Supported locales:

Add locales so that notification text appears in additional languages.
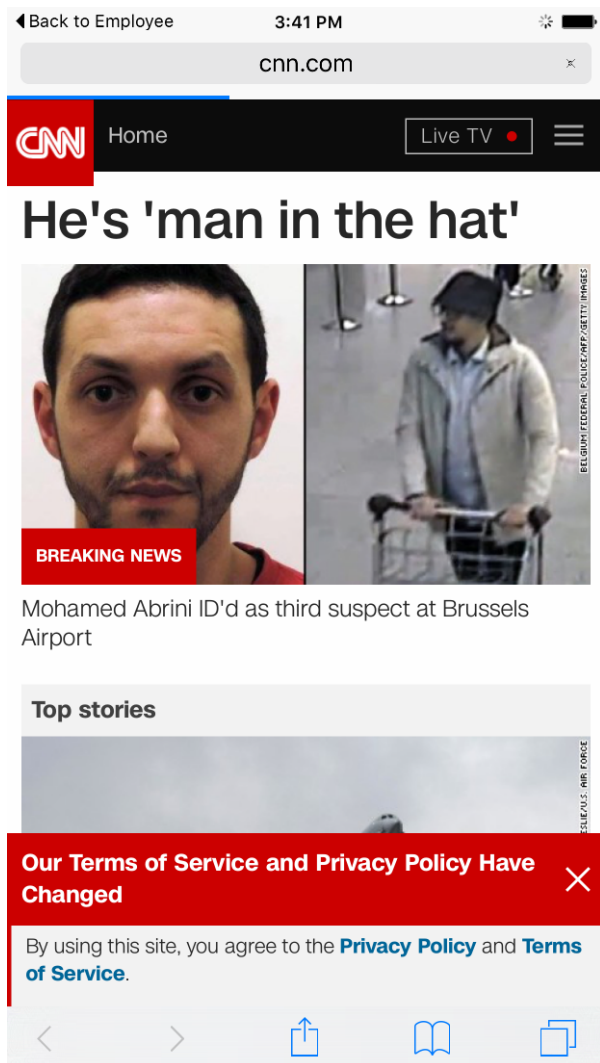
| **No locale supported** | Update Locales |

---

14. **Run** the application again using the cordova emulate command `cordova emulate`

    Notice the "Access disabled" notification when the application starts.

15. **Press** the "Upgrade" button, you will see the browser on your phone open the cnn.com website.

Summary

In this lab, you added javascript code to connect to the MobileFirst server when the application start, and used the built-in management features of the platform.

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

```
git checkout -f step-4
```