Introduction

In this set of labs we will demonstrate how to take an existing Ionic/Cordova application and add the MobileFirst Platform v8.0 Cordova Plugin, then demonstrate some of the capabilities provided by MobileFirst. We will cover bootstrapping, application customization, using MFP 8.0 adapters, MFP 8.0 Operational Analytics, custom report and more.

The activity will start with an Ionic project that we have already created (see git repo below). The application is an Employee Directory application, named IBMEmployeeApp

The following exercise includes 11 micro labs (~20 min each)

- Lab # 1 - Understand the Ionic application and learn how to customize it
- Lab # 2 - Use Cordova CLI to add the new MFP v8.0 Cordova Plugin
- Lab # 3 - Load MFP framework and application Bootstrapping
- Lab # 4 - Use MFP CLI to register and manage your application
- Lab # 5 - Using MFP adapters frameworks (Server Side)
- Lab # 6 - Using MFP adapters frameworks (Client side)
- Lab # 7 - Overview MFP Operational Analytics
- Lab # 8 - How to capture custom events
- Lab # 9 - How to create custom charts and alerts.
- Lab # 10 - How to secure your application (Server side)
- Lab # 11 - How to secure your application (Client side)

Source code for labs

In order to get the latest code for the ionic application, run the following git command:

```
git clone https://github.com/eliranbi/MadridMFPLabAll
```

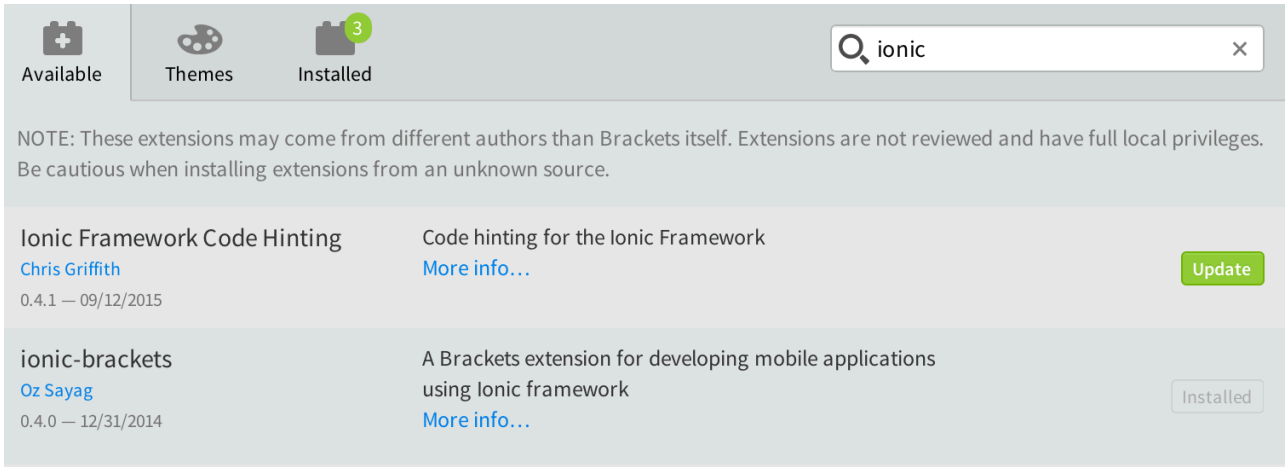
Once you clone the repo you will notice the following directories :

- **AdapterServices** - Is a folder that contian our MFP adapter projects, the adapters implement the back-end integration code.
- **IBMEmployeeApp** - Is our base Ionic Application that we are going to use for adding the MFP plugin.
- **LabDocuments** - Contain the workbooks for his lab.
- **snippets** - Contain a collection of copy/paste fragments to simplify making the required source code changes in the labs. They are labeled by lab name and task, and should be easy to locate and use.the snippets code foe the lab.

Tools used in labs

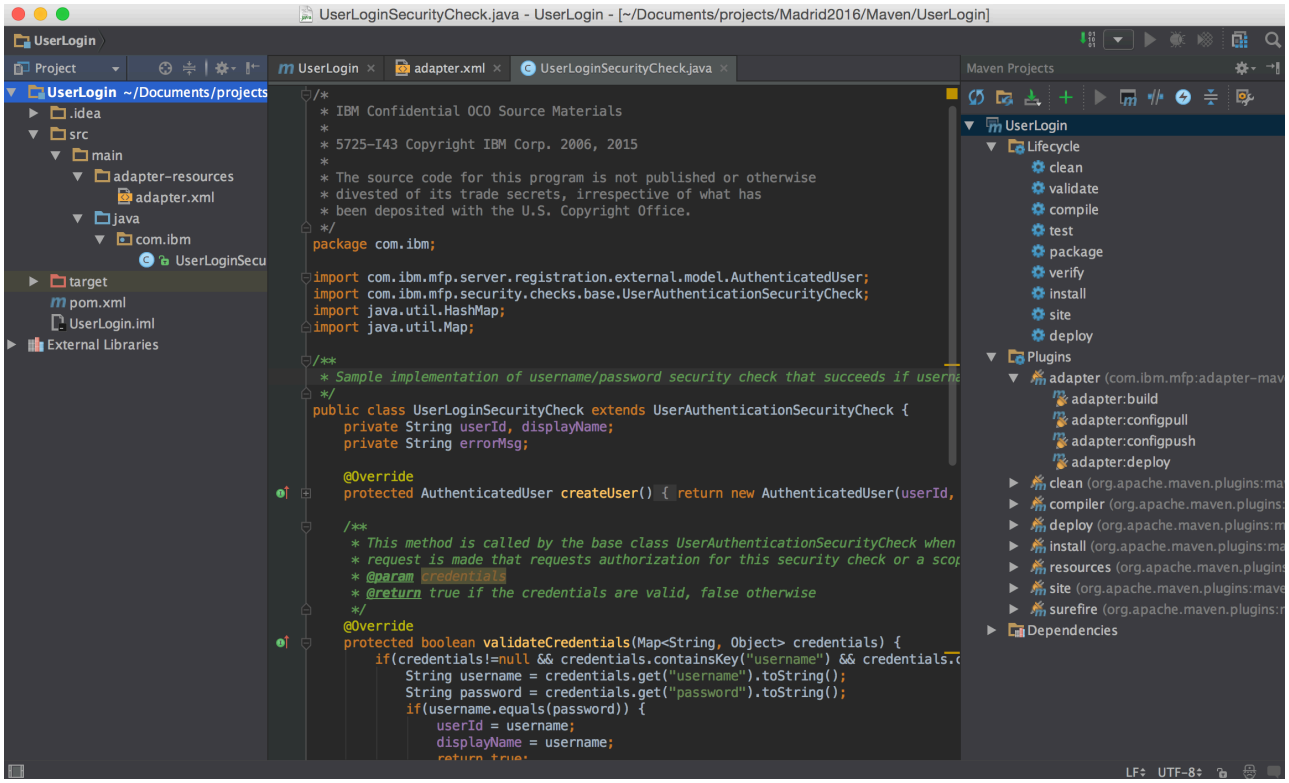
In this lab we will use the following tools :

1. The **MFP** mfpdev Command Line Interface (CLI) to interact with the MobileFirst Plaform, create projects, create adapters, deploy to the mfp server, view our mfp console etc.
2. Your choice of IDE to edit the code. The **Brackets IDE** was used throughout these labs and can be downloaded from here : http://brackets.io. Brackets is a modern, open source text editor that understands web design. You can also use the Brackets Extension manager to install additional plugins for code assistant and live preview. The extensions that are used in this tutorial are:
    - ionic-brackets.
    - Ionic Framework Code Hinting.
    - Brackets Beautify



3. **IntelliJ IDEA** will be used to create a MFP adapter and uset the built-in Maven project features, IntelliJ IDEA is a Java integrated development environment (IDE) for developing computer software. It is

developed by JetBrains, and is available as an Apache 2 Licensed community edition.
https://www.jetbrains.com/idea/download/



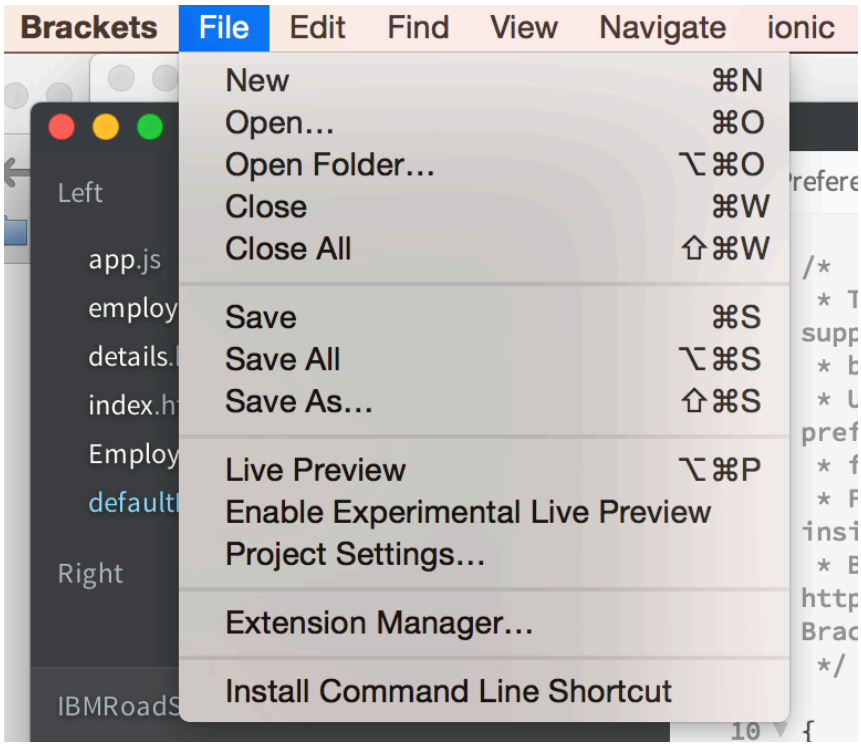
4. **Cordova** command line interface (CLI), This tool allows you to create new projects, build them on different platforms, and run on real devices or within emulators. The CLI is the main tool to use for the cross-platform workflow

5. **Ionic Framework** command line interface (CLI), The Ionic Framework command line utility makes it easy to start, build, run, and emulate Ionic apps.

Preview the Ionic app

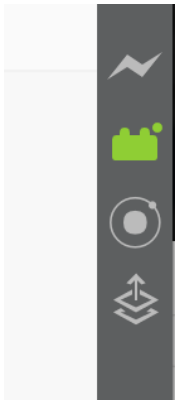
Lets start with preview our existing Ionic Employee Directory application.

1. Start Brackets and choose “Open folder” and select the IBMEmployeeApp folder.

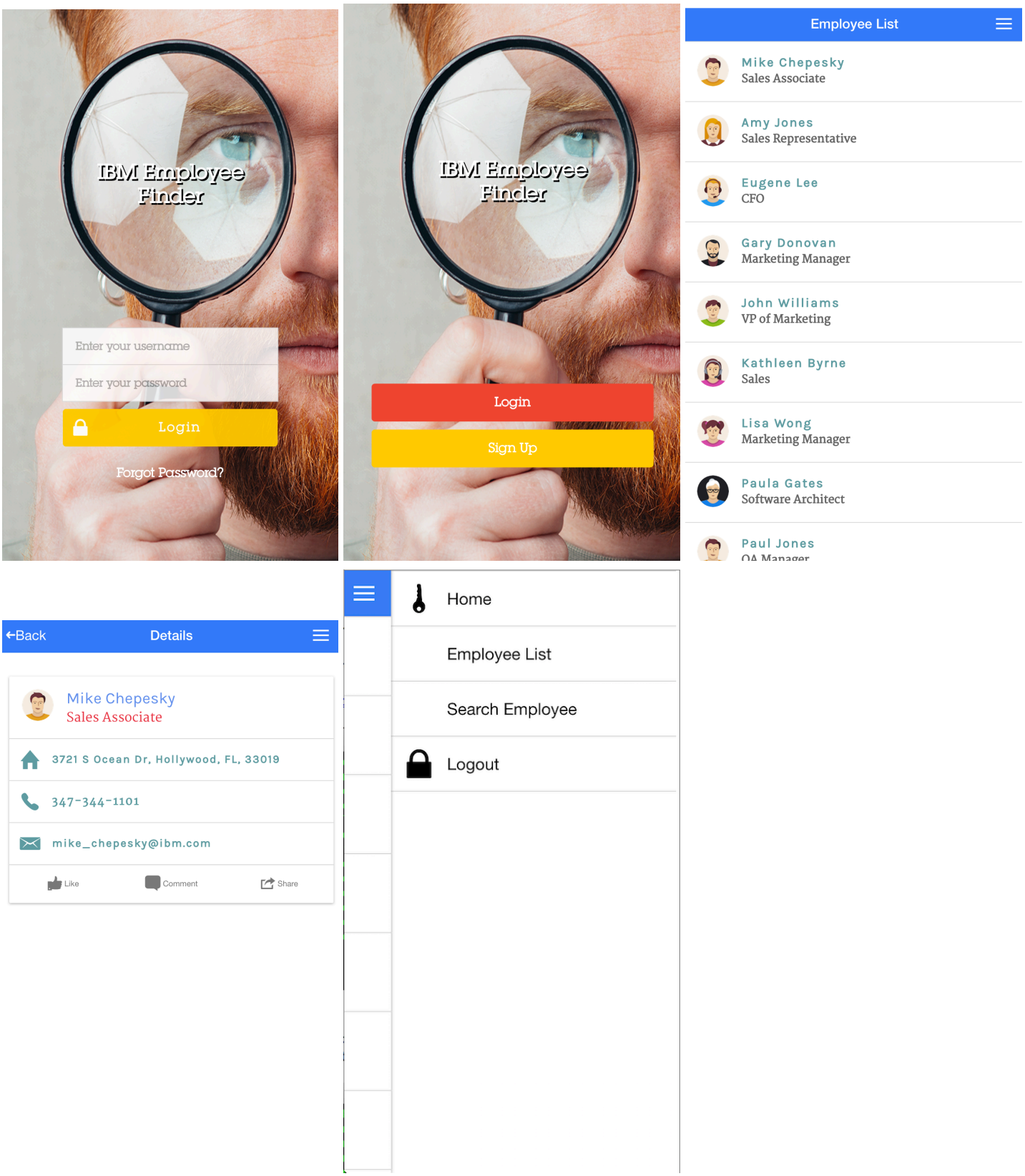


2. Use the Brackets file Navigator to locate and click on **IBMEmployeeApp/www/index.html**.

3. Click the **lighting** icon in the upper right portion of the Brackets window to preview the completed application in the browser.



Fro the username and password use the combination of **demo/demo** to login



Summary

You now have the completed and operational Ionic application in your initial workspace. In the next lab we will

reset our workspace to a known starting point using 'git checkout'. In later labs you will be able to use 'git checkout' to get the working project code, in case you get into trouble.