Adding the MobileFirst Platform Foundation SDK to iOS Applications

Overview

The MobileFirst Platform Foundation SDK provides a set of API methods enabling a developer to implement various MobileFirst features, such as: authentication and security mechanisms, notifications, resource requests, collecting analytics data and more.

> For a complete list of MobileFirst SDK abilities visit the user documentation

In this tutorial you will learn how to add the MobileFirst Native SDK using CocoaPods to either a new or existing iOS application. You will also learn how to configure the MobileFirst Server to recognize to application, as well as find information about the files added to your project by the SDK.

> Note: a "skeleton" Xcode project pre-bundled with the MobileFirst Native SDK can be downloaded from the MobileFirst Operations Console. Review the console tutorials to learn how. TODO: missing link

Pre-requisites: Xcode and MobileFirst CLI installed in the developer workstation.

Jump to:

Adding the MobileFirst Native SDK manually

The MobileFirst Native SDK is provided via CocoaPods. Follow the below instructions to add it manually:

1. Create an Xcode project or use an existing Xcode project.
2. If CocoaPods (http://guides.cocoapods.org) is not installed in your development environment, install it as follows:
    - Open **Terminal**
    - Run the command: `sudo gem install cocoapods`
    - Run the command: `pod setup`

      Note: This command may take several minutes to complete.
3. Change directory to the location of the Xcode project.
4. Run the command: `pod init`. This creates a `Podfile`.
5. Using your favorite editor, open the `Podfile` file.
6. Comment out or remove the contents of the file.

7. Add the following lines and save the changes:

```
source 'https://github.com/CocoaPods/Specs.git'
pod 'IBMMobileFirstPlatformFoundation'
```

8. Run the command: `pod install`. This command adds the MobileFirst Native SDK, generates the Pod project, and integrates it with the Xcode project. **Note:** This command may take several minutes to complete.

> **Important**: From here on, use the `[ProjectName].xcworkspace` file in order to open the project in Xcode. Do **not** use the `[ProjectName].xcodeproj` file. A CocoaPods-based project is managed as a workspace containing the application (the executable) and the library (all project dependencies that are pulled by the CocoaPods manager).

9. Open the Xcode project by double-clicking the **.xcworkspace** file.
10. Right-click the project and select **Add Files To [ProjectName]**, select the `mfpclient.plist`, located in the root folder of the Xcode project.
11. Whenever you want to use the MobileFirst Native SDK, make sure that you import the framework:

```
#import <IBMMobileFirstPlatformFoundation/IBMMobileFirstPlatformFoundatio
n.h>
```

Note about Swift:

> Because Swift is designed to be compatible with Objective-C you can use the MobileFirst SDK from within an iOS Swift project, too. Create a Swift project and follow the same steps, as described at the beginning of the tutorial, to integrate the MobileFirst Native SDK. Use `import IBMMobileFirstPlatformFoundation` in any class that needs to use the SDK.

Note about iOS 9:

> If you are developing for iOS9, consider disabling ATS (http://iosdevtips.co/post/121756573323/ios-9-xcode-7-http-connect-server-error) in the application's `info.plist` to be able to test locally without security restrictions.

Now that the MobileFirst Native SDK was added to the Xcode project, the final step to perform is to ensure that the MobileFirst Server will recognize any future requests arriving to it from the application.

Configuring the MobileFirst Server to recognize the application

Configuring the MobileFirst Server recognize the application can be achieved in two ways:

- By registering the application in the MobileFirst Operations Console.
- By registering the application using the MobileFirst CLI.

Registering the application using the MobileFirst CLI

1. Open **Terminal** and navigate to the root of the Xcode project.
2. Run the command: `mfpdev app register`

The `mfpdev app register` CLI command generates several required files as well as registers the application in the MobileFirst Server.
These files are further explained in the Generated MobileFirst Native SDK artifacts section below.

Registering the application in the MobileFirst Operations Console

1. Open your browser of choice and load the MobileFirst Operations Console using the address `http://localhost:10080/mfpconsole/`. You can also open the console from **Terminal** using the CLI command `mfpdev server console`.
2. Click on the button to create a new application and follow the on-screen instructions.

Add image of console -> create new app

Generated MobileFirst Native SDK artifacts

Two MobileFirst-related artifacts are available in the Xcode project after it has been integrated with the MobileFirst Native SDK: the `mfpclient.plist` and the the application-descriptor.json file.

mfpclient.plist

Located at the root of the project, this file contains server configuration properties and is user-editable:

TODO: add image of file?- `protocol` – The communication protocol to MobileFirst Server. Either `HTTP` or `HTPS`. - `host` – The hostname of the MobileFirst Server instance. - `port` – The port of the MobileFirst Server instance. - `wlServerContext` – The context root path of the application on the MobileFirst Server instance. - `languagePreference` - Sets the default language for client sdk system messages

application-descriptor.json

Located at the root of the project, this file contains properties related to the application and is user-editable: TODO: add contents of file

Tutorials to follow next

Now that the application is integrated with the MobileFirst Native SDK you can follow the tutorials in the Native iOS development (../../native/ios/) category to learn more about authentication and security, server-side development, notifications, and more.