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 (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html).

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 the application, as well as find information about the MobileFirst configuration files that are added to the project.

Pre-requisites:

- Xcode and MobileFirst developer CLI installed on the developer workstation.
- *Optional* MobileFirst Server to run a locally.
- Make sure you have read the Setting up your MobileFirst development environment (../../setting-up-the-mobilefirst-development-environment) tutorial.

Jump to:

- Adding the MobileFirst Native SDK
- Generated MobileFirst Native SDK artifacts
- Tutorials to follow next

Adding the MobileFirst Native SDK

Follow the below instructions to add the MobileFirst Native SDK to either a new or existing Xcode project, and register the application in the MobileFirst Server.

Before starting, make sure the MobileFirst Server is running.
If using a locally installed server: From a **Command-line** window, navigate to the server's **scripts** folder and run the command: `./start.sh`.

Creating and registering the application

1. Create an Xcode project or use an existing one (Swift or Objective-C).
2. Open a **Command-line** window and navigate to the root of the Xcode project.
3. Run the command:

   ```
   mfpdev app register
   ```

   The `mfpdev app register` CLI command first connects to the MobileFirst Server to register the application, followed by generating the `mfpclient.plist` file at the root of the Xcode project, and adding to it the metadata that identifies the MobileFirst Server.

> **Tip:** The application registration can also be performed from the MobileFirst Operations Console:
>
> 1. Open your browser of choice and load the MobileFirst Operations Console using the address `http://localhost:9080/mfpconsole/`. You can also open the console from the **Command-line** using the CLI command `mfpdev server console`.
> 2. Click on the "Create new" button next to "Applications" to create a new application and follow the on-screen instructions.
> 3. After successfully registering your application you can optionally download a "skeleton" Xcode project pre-bundled with the MobileFirst Native SDK.

4. Run the command:

   ```
   mfpdev app pull
   ```

   The `mfpdev app pull` CLI command creates the **mobilefirst** folder at the root of the Xcode project and downloads into it the `application-descriptor.json` file, containing application configuration data. These files are further explained in the Generated MobileFirst Native SDK artifacts section below.

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

Adding the SDK

1. The MobileFirst Native SDK is provided via CocoaPods. If CocoaPods (http://guides.cocoapods.org) is not installed in your development environment, install it as follows:
   - Open a **Command-line** window and navigate to the root of the Xcode project.
   - Run the command: `sudo gem install cocoapods` followed by `pod setup`. **Note:** These commands may take several minutes to complete.
2. Run the command: `pod init`. This creates a `Podfile`.
   - Using your favorite code editor, open the `Podfile`.
   - Comment out or delete the contents of the file.
   - Add the following lines and save the changes:

     ```
     // remove source line below before going live with the tutorials
     source 'https://hub.jazz.net/git/oper2000/imf-client-sdk-specs-inhouse.git'
     use_frameworks!
     pod 'IBMMobileFirstPlatformFoundation'
     ```

3. Run the command: `pod install`. This command adds the MobileFirst Native SDK files, adds the **mfpclient.plist** file and generates a Pod 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).

4. Open the Xcode project by double-clicking the **.xcworkspace** file.

5. In Xcode, right-click the project entry, click on **Add Files To [ProjectName]** and select the **mfpclient.plist** file, located at the **[ProjectName]/Pods/IBMMobileFirstPlatformFoundation/IBMMobileFirstPlatformFoundation/Resources** folder.

Referencing the SDK

Whenever you want to use the MobileFirst Native SDK, make sure that you import the MobileFirst Platform Foundation framework:

Objective-C:

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

Swift:

```
import IBMMobileFirstPlatformFoundation
```

Note about iOS 9:

> If you are developing for iOS 9, 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.

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` file and the `application-descriptor.json` file.

mfpclient.plist

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

- `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 in the **<xcode-project-root-directory>/mobilefirst** folder, this file contains application configuration settings such as its `bundleId` and `version` and is user-editable.

The file can be edited either locally or via the MobileFirst Operations Console.
If edited locally, the MobileFirst Server can be updated by running the CLI command: `mfpdev app push`.
The file can also be updated by pulling from the server its latest revision by running the CLI command: `mfpdev app pull`.

```
{
    "applicationKey": {
        "bundleId": "com.sampleone.bankApp",
        "version": "1.0",
        "clientPlatform":"ios"
    },

    ...
    ...
    ...
}
```

Tutorials to follow next

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