iOS end-to-end demonstration

Overview

The purpose of this demonstration is to experience an end-to-end flow where an application and an adapter are registered using the MobileFirst Operations Console, an "skeleton" Xcode project is downloaded and edited to call the adapter, and the result is printed to the log - verifying a successful connection with the MobileFirst Server.

Prerequisites:

- Xcode
- MobileFirst Developer CLI (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))
- *Optional* Stand-alone MobileFirst Server (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))

1. Starting the MobileFirst Server

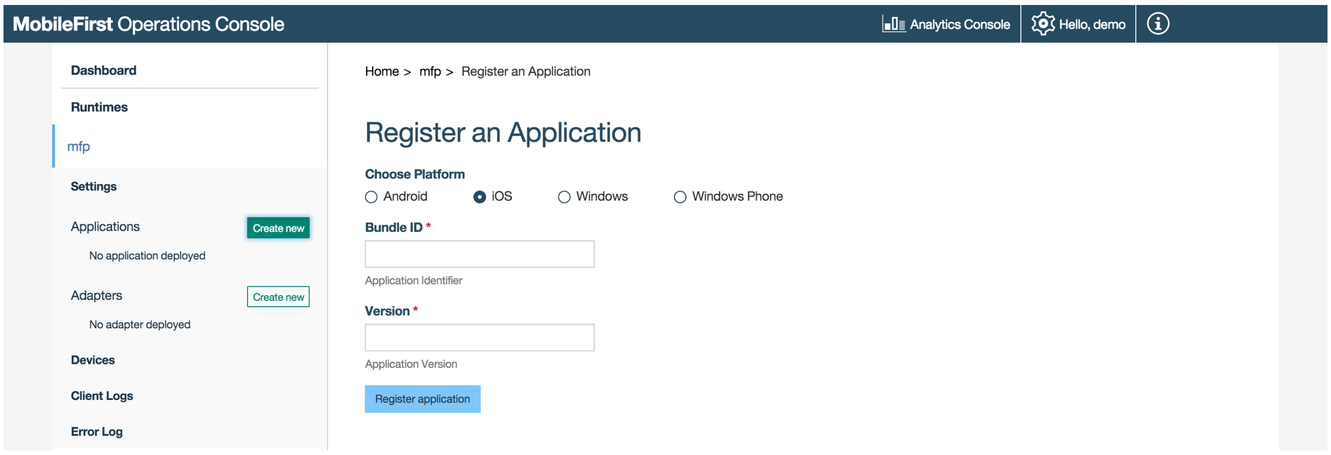
> If a remote server was already set-up, skip this step.

From a **Command-line** window, navigate to the server's **scripts** folder and run the command: `./start.sh`.

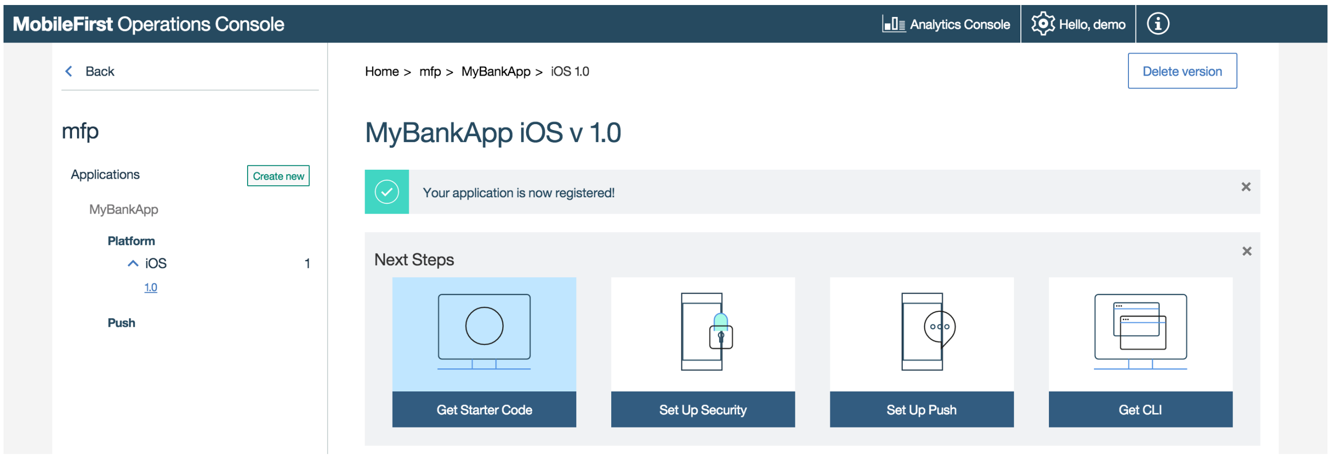
2. Creating an application

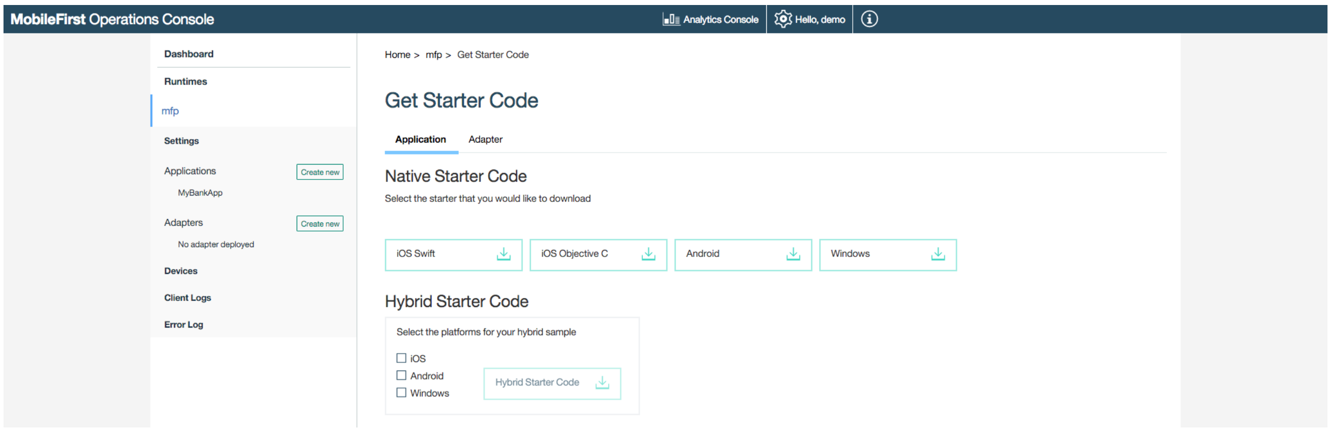
In a browser window, open the MobileFirst Operations Console by loading the URL: `http://your-server-host:server-port/mfpconsole`. If running locally, use: http://localhost:9080/mfpconsole (http://localhost:9080/mfpconsole). The username/password are *admin/admin*.

1. Click on the "Create new" button next to **Applications** and select the desired *platform*, *identifier* and *version* values.



2. Click on the **Get Starter Code** tile and select to download the iOS Starter Code.





3. Editing application logic

1. Open the Xcode project project by double-clickign the **.xcworkspace** file.
2. Select the **[project-root]/ViewController.m/swift** file and:

- Add the following header:

  In Objective-C:

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

  In Swift:

```
import IBMMobileFirstPlatformFoundation
```

- Paste the following code snippet, replacing the existing `viewDidLoad()` function:

  In Objective-C:

```
- (void)viewDidLoad {
    [super viewDidLoad];

    NSURL* url = [NSURL URLWithString:@"/adapters/javaAdapter/users/world"];
    WLResourceRequest* request = [WLResourceRequest requestWithURL:url method:WLHttpMethodGet];

    [request sendWithCompletionHandler:^(WLResponse *response, NSError *error) {
        if (error != nil){
             NSLog(@"Failure: %@",error.description);
        }
        else if (response != nill){
            // Will print "Hello world" in the Xcode Console.
            NSLog(@"Success: %@",response.responseText);
        }
    }];
}
```

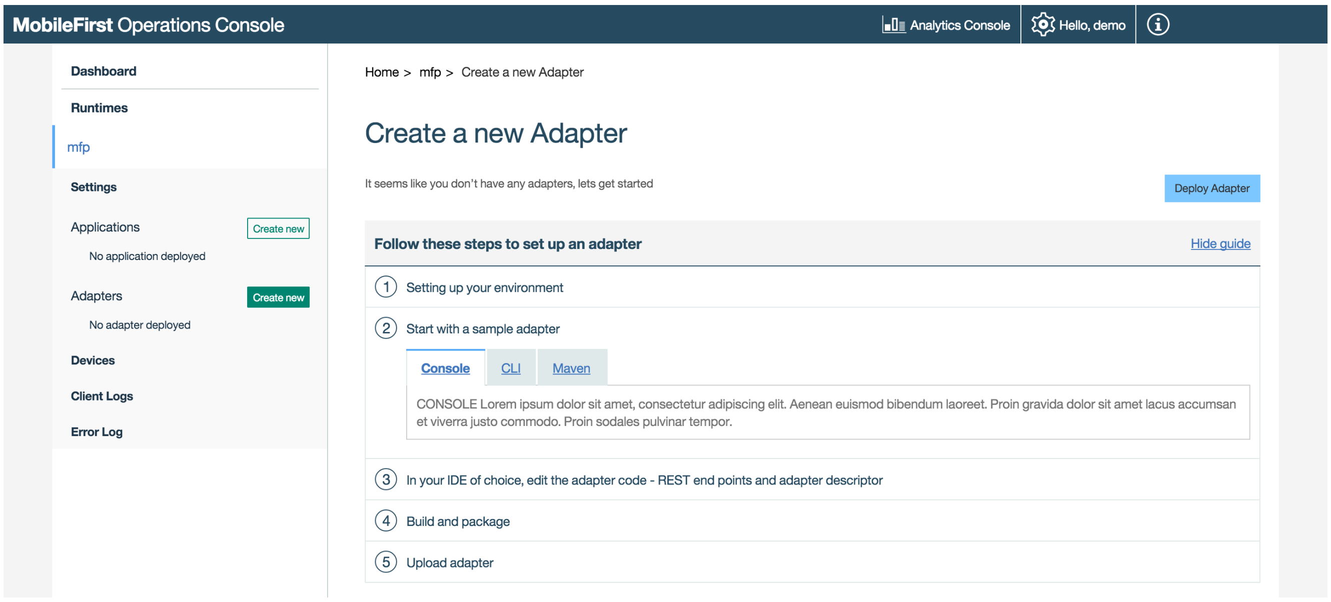
  In Swift:

```
override func viewDidLoad() {
    super.viewDidLoad()

    let url = NSURL(string: "/adapters/javaAdapter/users/world")
    let request = WLResourceRequest(URL: url, method: WLHttpMethodGet)

    request.sendWithCompletionHandler { (WLResponse response, NSError error) -> Void in
        if (error != nil){
            NSLog("Failure: " + error.description)
        }
        else if (response != nil){
            NSLog("Success: " + response.responseText)
        }
    }
}
```

4. Creating an adapter

1. Click on the "Create new" button next to **Adapters** and download the **Java** adapter sample.

> If Maven and MobileFirst CLI are not installed, follow the on-screen **Setting up your environment** instructions to install.



MobileFirst Operations Console — Analytics Console — Hello, demo

Dashboard
Runtimes
mfp
Settings
Applications — Create new
No application deployed
Adapters — Create new
No adapter deployed
Devices
Client Logs
Error Log

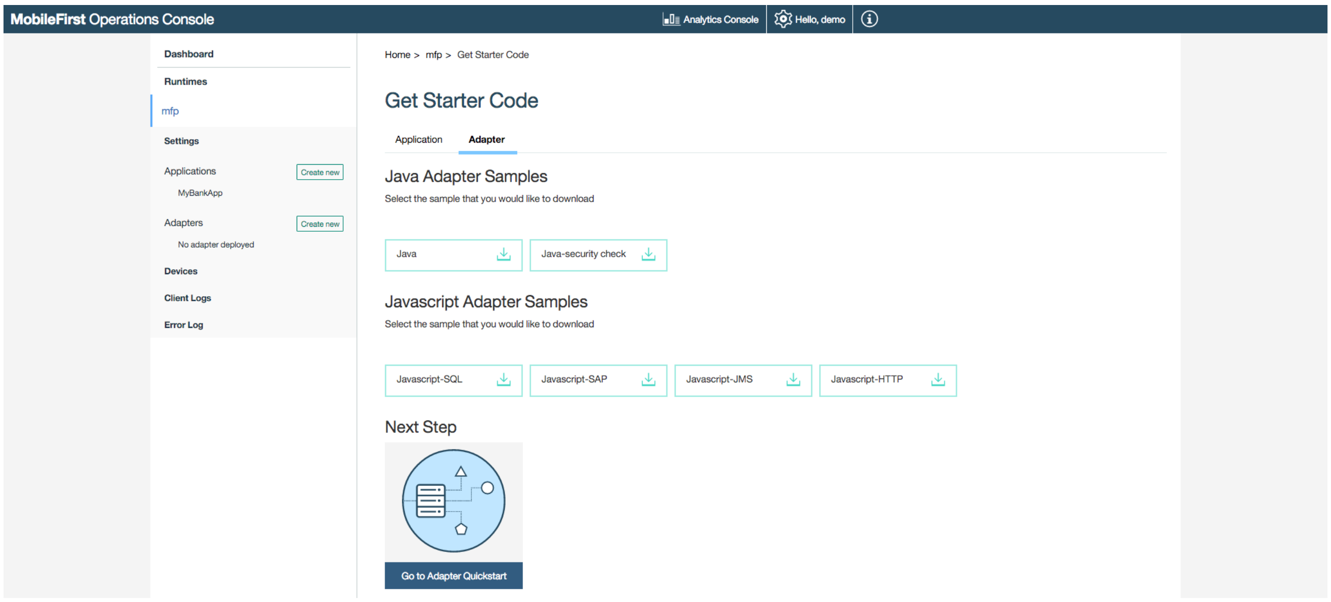
Home > mfp > Create a new Adapter

Create a new Adapter

It seems like you don't have any adapters, lets get started — Deploy Adapter

Follow these steps to set up an adapter — Hide guide

1. Setting up your environment
2. Start with a sample adapter
   Console | CLI | Maven
   CONSOLE Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor.
3. In your IDE of choice, edit the adapter code - REST end points and adapter descriptor
4. Build and package
5. Upload adapter



MobileFirst Operations Console — Analytics Console — Hello, demo

Dashboard
Runtimes
mfp
Settings
Applications — Create new
MyBankApp
Adapters — Create new
No adapter deployed
Devices
Client Logs
Error Log

Home > mfp > Get Starter Code

Get Starter Code

Application | Adapter

Java Adapter Samples
Select the sample that you would like to download
Java | Java-security check

Javascript Adapter Samples
Select the sample that you would like to download
Javascript-SQL | Javascript-SAP | Javascript-JMS | Javascript-HTTP

Next Step
Go to Adapter Quickstart

2. From a **Command-line** window, navigate to the adapter's Maven project root folder and run the command:

```
mfpdev adapter build
```

3. When the build finishes, run the command:

```
mfpdev adapter deploy
```

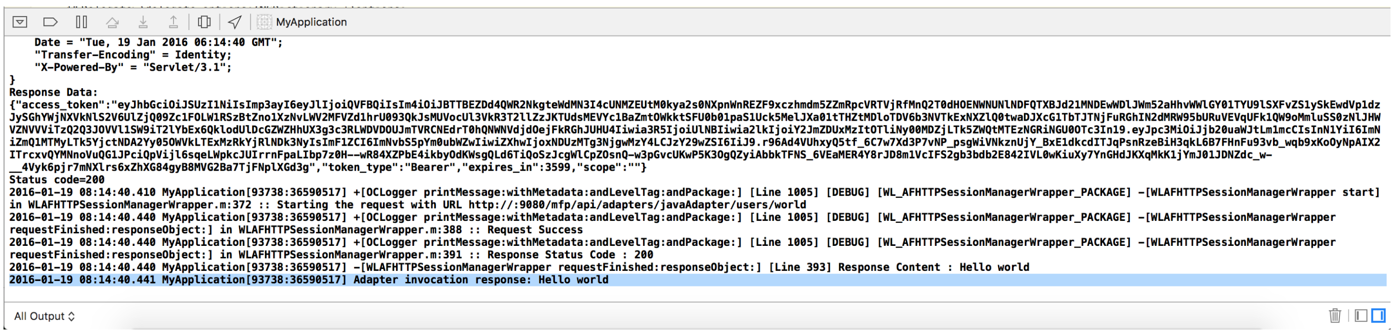
If using a remote MobileFirst Server, run the command:

```
mfpdev adapter deploy Replace-with-remote-server-name
```

5. Testing the application

1. In Xcode, select the **mfpclient.plist** file and edit the **host** property with the IP address of the MobileFirst Server.
2. Press the **Play** button.

The adapter response is then printed in the Xcode Console.



```
    Date = "Tue, 19 Jan 2016 06:14:40 GMT";
    "Transfer-Encoding" = Identity;
    "X-Powered-By" = "Servlet/3.1";
}
Response Data:
{"access_token":"eyJhbGciOiJSUzI1NiIsImp3ayI6eyJlIjoiQVFBQiIsIm4iOiJBTTBEZDd4QWR2NkgteWdMN3I4cUNMZEUtM0kya2s0NXpnWnREZF9xczhmdm5ZZmRpcVRTVjRfMnQ2T0dHOENWNUNlNDFQTXBJd21MNDEwWDlJWm52aHhvWWlGY01TYU9lSXFvZS1ySkEwdVp1dz
JySGhYWjNXVkNlS2V6UlZjQ09Zc1FOLW1RSzBtZno1XzNvLWV2MFVZd1hrU093QkJsMUVocUl3VkR3T2llZzJKTUdsMEVYc1BaZmtOWkktSFU0b01paS1Uck5MelJXa01tTHZtMDloTDV6b3NVTkExNXZlQ0twaDJXcG1TbTJTNjFuRGhIN2dMRW95bURuVEVqUFk1QW9oMmluSS0zNlJHW
VZNVVViTzQ2Q3JOVVl1SW9iT2lYbEx6QklodUlDcGZWZHhUX3g3c3RLWDVDOUJmTVRCNEdrT0hQNWNVdjdOejFkRGhJUHU4Iiwia3R5IjoiUlNBIiwia2lkIjoiY2JmZDUxMzItOTliNy00MDZjLTk5ZWQtMTEzNGRiNGU0OTc3In19.eyJpc3MiOiJjb20uaWJtLm1mcCIsInN1YiI6ImN
iZmQ1MTMyLTk5YjctNDA2Yy05OWVkLTExMzRkYjRlNDk3NyIsImF1ZCI6ImNvbS5pYm0ubWZwIiwiZXhwIjoxNDUzMTg3NjgwMzY4LCJzY29wZSI6IiJ9.r96Ad4VUhxyQ5tf_6C7w7Xd3P7vNP_psgWiVNkznUjY_BxE1dkcdITJqPsnRzeBiH3qkL6B7FHnFu93vb_wqb9xKoOyNpAIX2
ITrcxvQYMNnoVuQG1JPciQpVijl6sqeLWpkcJUIrrnFpaLIbp7z0H--wR84XZPbE4ikbyOdKWsgQLd6TiQoSzJcgWlCpZOsnQ-w3pGvcUKwP5K3OgQZyiAbbkTFNS_6VEaMER4Y8rJD8m1VcIFS2gb3bdb2E842IVL0wKiuXy7YnGHdJKXqMkK1jYmJ01JDNZdc_w-
__4Vyk6pjr7mNXlrs6xZhXG84gyB8MVG2Ba7TjFNplXGd3g","token_type":"Bearer","expires_in":3599,"scope":""}
Status code=200
2016-01-19 08:14:40.410 MyApplication[93738:36590517] +[OCLogger printMessage:withMetadata:andLevelTag:andPackage:] [Line 1005] [DEBUG] [WL_AFHTTPSessionManagerWrapper_PACKAGE] -[WLAFHTTPSessionManagerWrapper start]
in WLAFHTTPSessionManagerWrapper.m:372 :: Starting the request with URL http://:9080/mfp/api/adapters/javaAdapter/users/world
2016-01-19 08:14:40.440 MyApplication[93738:36590517] +[OCLogger printMessage:withMetadata:andLevelTag:andPackage:] [Line 1005] [DEBUG] [WL_AFHTTPSessionManagerWrapper_PACKAGE] -[WLAFHTTPSessionManagerWrapper
requestFinished:responseObject:] in WLAFHTTPSessionManagerWrapper.m:388 :: Request Success
2016-01-19 08:14:40.440 MyApplication[93738:36590517] +[OCLogger printMessage:withMetadata:andLevelTag:andPackage:] [Line 1005] [DEBUG] [WL_AFHTTPSessionManagerWrapper_PACKAGE] -[WLAFHTTPSessionManagerWrapper
requestFinished:responseObject:] in WLAFHTTPSessionManagerWrapper.m:391 :: Response Status Code : 200
2016-01-19 08:14:40.440 MyApplication[93738:36590517] -[WLAFHTTPSessionManagerWrapper requestFinished:responseObject:] [Line 393] Response Content : Hello world
2016-01-19 08:14:40.441 MyApplication[93738:36590517] Adapter invocation response: Hello world
```

Note: Xcode 7 enables Application Transport Security (ATS) (https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS9.html#//apple_ref/doc/uid/TP40016198-SW14) by default.
To complete the tutorial, disable ATS (http://iosdevtips.co/post/121756573323/ios-9-xcode-7-http-connect-server-error).

1. In Xcode, right-click the **[project]/info.plist file → Open As → Source Code**
2. Paste the following:

```
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>
```

3. Press the **Play** button.

Next steps

Learn more on using adapters in applications, and how to integrate additional services such as Push Notifications, using the MobileFirst security framework and more:

- Review the Server-side development tutorials (../../adapters/)
- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)