

Windows 8.1 Universal and Windows 10 UWP 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" Visual Studio 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:

- Configured Visual Studio 2013/5
- 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

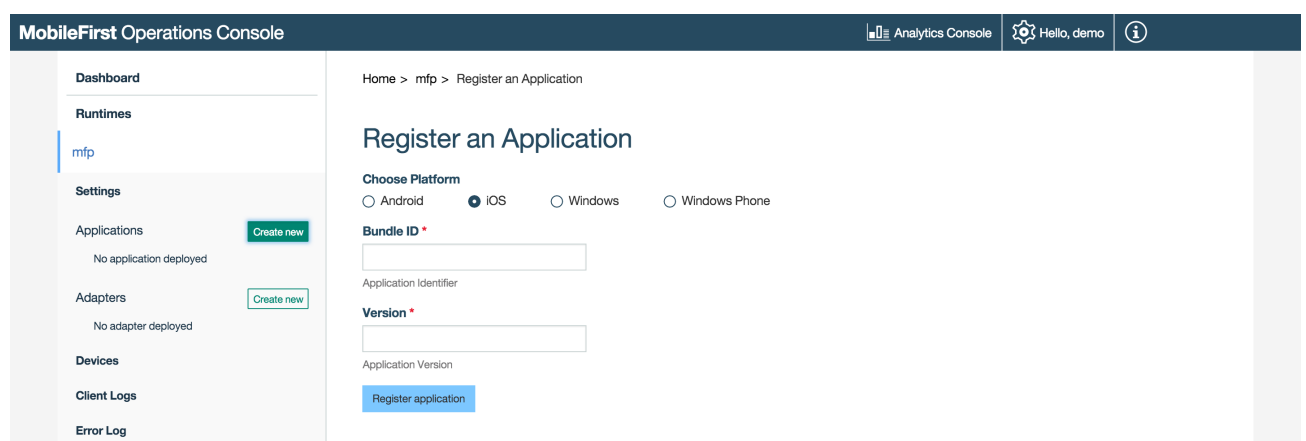
If a remote server was already set-up, skip this step.

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

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 Windows 8.1 or Windows 10 Starter Code.



3. Editing application logic

1. Open the Visual Studio project.
2. Select the solution's **MainPage.xaml.cs** file and paste the following code snippet:

```
IWorklightClient _newClient = WorklightClient.CreateInstance();

StringBuilder uriBuilder = new StringBuilder().Append("/adapters/javaAdapter/users/world");

WorklightResourceRequest rr = _newClient.ResourceRequest(uriBuilder.ToString(), "GET");

WorklightResponse resp= return Task.Run<WorklightResponse> (() => {
    rr.send();
});

if (resp.success) {
    Debug.WriteLine("Success: " + resp.ResponseText);
} else {
    Debug.WriteLine("Failure: " + resp.error);
}
```

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

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

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 Visual Studio, click on the **Start Debugging** 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](#))