

Setting up the MobileFirst development environment

fork and edit tutorial (<https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/setting-up-your-development-environment/mobilefirst-development-environment.md>) | report issue (<https://github.ibm.com/MFPSamples/DevCenter/issues/new>)

Overview

This tutorial covers the basics of the IBM MobileFirst Platform Foundation development environment and tools.

- The MobileFirst Platform Foundation developer tools are supported on the following operating systems: Windows, Apple OS X and Linux.
- The MobileFirst Operations Console is supported in modern browsers such as: Internet Explorer 10+, latest Chrome, Safari and Firefox.

Jump to:

- MobileFirst Development Kit
- Applications and Adapters Development

MobileFirst Development Kit

The IBM MobileFirst Development Kit consists of the following:

MobileFirst Server

The IBM MobileFirst Server is a web application archive (.war) that is deployed on an IBM WebSphere Liberty profile Application Server.

From the MobileFirst Server's Operations Console a developer can:

- Register and deploy applications and adapters
- Optionally download native/Cordova application and adapter templates
- Configure application aspects:
 - Authentication and security
 - Application Authenticity
 - Push Notifications
 - Direct Update
- Generate DevOps scripts for continuous integration workflows and faster development cycles
- and more

The MobileFirst Development Server is installed as a stand-alone server and provides the following scripts in its **scripts** folder:

`.sh` scripts are for Linux and Mac.

`.cmd` scripts are for Windows.

- `console.[sh|cmd]` : Open the local MobileFirst Console
- `run.[sh|cmd]` : Run the local MobileFirst Server with trailing Liberty Server messages
- `start.[sh|cmd]` : Run the local MobileFirst Server in background mode
- `stop.[sh|cmd]` : Stop the current local MobileFirst Server instance

To learn more about the MobileFirst Operations Console, see the tutorial: [Using the MobileFirst Platform Operations Console \(../../quick-start/console/\)](#).

To download and install the MobileFirst Development Server, visit the [downloads \(/downloads/\)](#) page.

Adding the MobileFirst Server to Eclipse

The MobileFirst Server can be integrated into the Eclipse IDE, allowing to quickly starting and stopping of the development server.

1. From the **Servers** view in Eclipse, select **New → Server**.
2. If an IBM folder option does not exist, click on "Download additional server adapters".
3. Select **WebSphere Application Server Liberty Tools** and follow the on-screen instructions.
4. From the **Servers** view in Eclipse, select **New → Server**.
5. Select **IBM → WebSphere Application Server Liberty**.
6. Provide a server **name** and **hostname** and click **Next**.
7. Provide the path to the server's root directory, and select a JRE version to use.
8. Click **Next** followed by clicking **Finish**.

MobileFirst Command-line Interface

The IBM MobileFirst command-line interface (CLI) tool enables developers to:

- Manage MobileFirst Servers
- Register and configure applications
- Create, build, deploy and test adapters

To download and install the MobileFirst Developer CLI, visit the [downloads \(/downloads/\)](#) page.

Learn more about the various CLI commands in the [Using CLI to manage MobileFirst artifacts \(../../using-the-mfpf-sdk/using-cli-to-manage-mobilefirst-artifacts/\)](#) tutorial.

MobileFirst Platform Foundation client and server SDKs & APIs

MobileFirst Platform Foundation provides client-side SDKs for Cordova applications as well as for Native platforms (iOS, Android and Windows 8.1 Universal & Windows 10 UWP). Server-side APIs for adapter development are available as well.

To use the MobileFirst client SDKs, visit the [Adding the MobileFirst Platform Foundation SDK \(../../adding-the-mfpf-sdk/\)](#) tutorials.

To use the MobileFirst server-side APIs, visit the [Adapters \(../../adapters/\)](#) tutorials.

Applications and adapters development

You can use your preferred code editor or alternative IDEs, such as Atom.io, Visual Studio Code, Eclipse, IntelliJ and others, to implement applications and adapters.

For adapters development, refer to the [Adapters \(../../adapters/\)](#) category as well as to the [Developing Adapters in IDEs \(../../adapters/developing-adapters/\)](#) and [testing and debugging adapters \(../../adapters/testing-and-debugging-adapters/\)](#) tutorials.

Note: API auto-completion is available only in Typescript-supporting IDEs.

