Developing Applications

Development Concepts and Overview

When you develop your app with the MobileFirst Foundation set of tools, you must develop or configure a variety of components and elements. Learning about the components and elements involved when developing your app helps your development proceed smoothly. In addition to getting familiar with these concepts, you will also learn about MobileFirst-provided APIs for Native, Cordova and Web applications, such as JSONStore and WLResourceReuest, as well as learn how to debug applications, use Direct Update to refresh the web resources, Live Update to segment your userbase as well as how to handle apps, adapters and other artifacts using the MobileFirst CLI.

Continue reading to learn more about the MobileFirst Components or select a tutorial from the sidebar navigation.

Jump to

Applications

Applications are built for a target MobileFirst Server and have a server-side configuration on the target server. You must register your applications on the MobileFirst Server before you can configure them.

Applications are identified by the following elements:

- An app ID
- A version number
- A target deployment platform

Note: The version number is not applicable to web applications. You cannot have multiple versions of the same web application.

These identifiers are used on both the client-side and the server-side to ensure that apps are deployed correctly and use only resources that are assigned to them. Different parts of IBM MobileFirst Foundation use various combinations of these identifiers in different ways.

The app ID depends on the target deployment platform:

Android
For Android apps, the identifier is the application package name.

iOS
For iOS apps, the identifier is the application bundle ID.

Windows
For Windows apps, the identifier is application assembly name.

Web
For web apps, the identifier is a unique ID that is assigned by the developer.

If apps for different target platforms all have the same app ID, then the MobileFirst Server considers all of these apps to be the same app with different platform instances. For example, the following apps are considered to be different platform instances of the same app:

- An iOS app with a bundle ID of `com.mydomain.mfp`.
- An Android app with a package name of `com.mydomain.mfp`.
- A Windows 10 Universal Windows Platform app with an assembly name of `com.mydomain.mfp`.
- A web app with an assigned ID of `com.mydomain.mfp`.

The target deployment platform for the app is independent of whether the app was developed as a native app or as a Cordova app. For example, the following apps are both considered to be iOS apps in IBM MobileFirst Foundation:

- An iOS app that you develop with Xcode and native code
- An iOS app that you develop with Cordova cross-platform development technologies

Application configuration

As mentioned, an application is configured on both the client-side and the server-side. For native and Cordova iOS, Android, and Windows applications, the client configuration is stored in a client properties file (**mfpclient.plist** for iOS, **mfpclient.properties** for Android, or **mfpclient.resw** for Windows). For web applications, the configuration properties are passed as parameters to the SDK initialization method (../adding-the-mfpf-sdk/web). The client configuration properties include the application ID and information such as the URL of the MobileFirst Server runtime and security keys that are required to access to the server. The server configuration for the app includes information like app management status, web resources for Direct Update, configured security scopes, and log configuration.

> Learn how to add the MobileFirst Client SDKs in the Adding the MobileFirst Foundation SDK tutorials (../adding-the-mfpf-sdk)

The client configuration must be defined before you build the application. The client-app configuration properties must match the properties that are defined for this app in the MobileFirst Server runtime. For example, security keys in the client configuration must match the keys on the server. For non-web apps, you can change the client configuration with the MobileFirst CLI.

The server configuration for an app is tied to the combination of app ID, version number, and target platform. You must register your app to a MobileFirst Server runtime before you can add server-side configurations for the app. Configuring the server side of an app is typically done with the MobileFirst Operations Console. You can also configure the server side of an app with the following methods:

- Grab existing JSON configuration files from the server with the `mfpdev app pull` command, update the file, and upload the changed configuration with the `mfpdev app push` command.
- Use the **mfpadm** program or Ant task. For information about using mfpadm, see Administering MobileFirst applications through the command line (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.admin.doc/admin/c_administering_ibm_worklight_applications_through_command_lin view=kc#administeringworklightapplicationsthroughthecommandline) and Administering MobileFirst applications through Ant (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.admin.doc/admin/c_administering_ibm_worklight_applications_through_ant.html?view=kc#administeringibmworklightapplicationsthroughant).

- Use the REST API of the MobileFirst administration service. For information about the REST API, see REST API for the MobileFirst Server administration service (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.apiref.doc/apiref/c_restapi_oview.html?view=kc#restservicesapi).

You can also use these methods to automate configuring your MobileFirst Server.

Remember: You can modify the server configuration even while a MobileFirst Server is running and receiving traffic from apps. You do not need to stop the server to change the server configuration for an app.

On a production server, the app version typically corresponds to the version of the application published to an app store. Some server configuration elements like the configuration for app authenticity, are unique to the app published to the store.

MobileFirst Server

The server-side of your mobile app is MobileFirst Server. MobileFirst Server gives you access to features like application management and application security, as well giving your mobile app secure access to your other backend systems through adapters.

MobileFirst Server is the core component that delivers many IBM MobileFirst Foundation features, including the following features:

- Application management
- Application security, including authenticating devices and users and verifying application authenticity
- Secure access to backend services through adapters
- Updating Cordova app Web resources with Direct Update
- Push notifications and push subscriptions
- App analytics

You need to use MobileFirst Server throughout your app's lifecycle from development and test through to production deployment and maintenance.

> A preconfigured server is available for you to use when you develop your app. For information about the MobileFirst Development Server to use when you develop your app, see Setting up the MobileFirst Development Environment (../setting-up-your-development-environment).

MobileFirst Server consists of the following components. All of these components are also included in the MobileFirst Development Server. In simple cases, they are all running on the same application server, but in a production or test environment, the components can be run on different application servers. For information about possible topologies for these MobileFirst Server components, see Topologies and network flows (http://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.installconfig.doc/install_config/c_mfp_server_topologies.html).

MobileFirst Operations Console and the MobileFirst Server administration service

The operations console is a web interface that you can use to view and edit the MobileFirst Server configurations. You can also access the MobileFirst Analytics Console from here. The context root for the operations console in the development server is **/mfpconsole**.

The administration service is the main entry point for managing your apps. You can access the administration service through a web-based interface with the MobileFirst Operations Console. You can also access the administration service with the **mfpadm** command-line tool or the administration service REST API.

> Learn more about the MobileFirst Operations Console features (../setting-up-your-development-environment/console).

MobileFirst runtime

The runtime is the main entry point for a MobileFirst client app. The runtime is also the default authorization server for the IBM MobileFirst Foundation OAuth implementation.

In advanced and rare cases, you can have multiple instances of a device runtime in a single MobileFirst Server. Each instance has its own context root. The context root is used to display the name of a runtime in the operations console. Use multiple instances in cases where you require different server-level configuration such as secret keys for keystore.

If you have only one instance of a device runtime in MobileFirst Server, you do not typically need to know the runtime context root. For example, when you register an application to a runtime with the `mfpdev app register` command when the MobileFirst Server has only one runtime, the application is registered automatically to that runtime.

MobileFirst Server push service

The push service is your main access point for push-related operations like push notifications and push subscriptions. To contact the push services, client apps use the URL of the runtime but replace the context root with /mfppush. You can configure and manage the push service with the MobileFirst Operations Console or the push service REST API.

If you run the push services in a separate application server from the MobileFirst runtime, you must route the push service traffic to the correct application server with your HTTP server.

MobileFirst Analytics and the MobileFirst Analytics Console

IBM MobileFirst Analytics is an optional component that provides a scalable analytics feature that you can access from the MobileFirst Operations Console. This analytics feature lets you search for patterns, problems and platform usage statistics across logs and events that are collected from devices, apps, and servers.

From the MobileFirst Operations Console, you can define filters to enable or disable data forwarding to the analytics service. You can also filter the type of information that is sent. On the client side, you can use the client-side log capture API to send events and data to the analytics server.

After you install and configure MobileFirst Server into the topology that you want, any further configuration of MobileFirst Server and its applications can be done entirely through any of the following methods:

- The MobileFirst Operations Console
- The MobileFirst Server administration service REST API
- The **mfpadm** command-line tool

After the initial installation and configuration, you do not need to access any application server console or interface to configure IBM MobileFirst Foundation.
When you deploy your app to production, you can deploy your app to the following MobileFirst Server production environments:

On-premises

For information about installing and configuring MobileFirst Server for your on-premises environment, see Installing IBM MobileFirst Server (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.installconfig.doc/admin/c_installation.html#installation).

On the cloud

For information, see Deploying MobileFirst Server to the cloud (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.deploy.doc/topics/c_deploy.html#deployment).

Adapters

Adapters in IBM MobileFirst Foundation securely connect your back-end systems to client applications and cloud services.

You can write adapters in either JavaScript or Java™, and you can build and deploy adapters as Maven projects.
Adapters are deployed to a MobileFirst runtime in MobileFirst Server.

In a production system, adapters typically run in a cluster of application servers. Implement your adapters as REST services with no session information and stored locally on the server to ensure that your adapter works well in a clustered environment.

An adapter can have user-defined properties. These properties can be configured on the server side without redeploying the adapter. For example, you can change the URL that your adapter uses to access resources when you move from test to production.

You can deploy an adapter to a MobileFirst runtime from the MobileFirst Operations Console, by using the mfpdev adapter deploy command, or directly from Maven.

> Learn more about adapters and how to develop JavaScript and Java adapters in theAdapters category (../adapters).

Tutorials to Follow Next