

Application Authenticity

Overview

By issuing an HTTP request, any entity can access the HTTP services (APIs) that IBM MobileFirst Platform Foundation Server offers.

The out-of-the-box Application Authenticity security check ([../authentication-concepts/](#)) ensures that an application that tries to connect to a MobileFirst Server instance is the authentic one and was not tampered with or modified by a third-party attacker.

To enable Application Authenticity protection you can either follow the on-screen instructions in the MobileFirst Operations Console → [your-application] → Authenticity, or review the information below.

Availability

Application Authenticity is available in all supported platforms (iOS, Android, Windows 8 Universal, Windows 10 UWP) in both Cordova and Native applications.

Note: Application Authenticity is **not available** in the MobileFirst Development Server. To test, use a remote application server such as a QA, UAT or Production server.

Jump to:

- [Authenticity flow \(authenticity-flow\)](#)
- [Enabling authenticity \(enabling-application-authenticity\)](#)
- [Disabling authenticity \(disabling-application-authenticity\)](#)
- [Configuring authenticity \(configuring-application-authenticity\)](#)

Authenticity Flow

Application Authenticity is based on certificate keys that are used to sign the application bundles. Only the developer or the enterprise who have the original private key that was used to create the application are able to modify, repack, and re-sign the bundle.

Once an application has successfully registered with the MobileFirst Server, and passed the Authenticity challenge, an Authenticity token is granted. For as long as the token is valid, the Authenticity challenge will not occur again. See [Configuring authenticity \(configuring-authenticity\)](#) to learn how this can be customized.



The challenge token in the diagram is processed by compiled native code, so that third-party attackers cannot see the logic of this processing.

Enabling Application Authenticity

In order to enable Application Authenticity in your Cordova or Native application, the application's binary file needs to be signed using the MobileFirst-supplied command line tool. Eligible binary files are: `ipa` for iOS, `apk` for Android and `appx` for Windows 8 Universal & Windows 10 UWP.

1. Open a **Command-line** window and run the command: `java -jar path-to-mfp-server-authenticity-tool.jar path-to-binary-file`

For example:

```
java -jar /Users/idanadar/Desktop/mfp-server-authenticity-tool.jar /Users/idanadar/Desktop/MyBankApp.ipa
```

The result of the command above is a `.authenticity_data` file generated next to the `MyBankApp.ipa` file, called `MyBankApp.authenticity_data`.

2. Open the MobileFirst Operations Console in your browser of choice.
3. Select your application from the left-side pane and click on the Authenticity menu item.
4. Click on "Upload Authenticity File" to upload the `.authenticity_data` file.

Once the `.authenticity_data` file is uploaded, Application Authenticity is now enabled.

The screenshot shows the MobileFirst Operations Console interface. On the left, a sidebar contains a 'Back' link, the application name 'mfp', and a list of applications under 'Applications' with a 'Create new' button. The selected application is 'com.worklight.MyBankApp'. Under 'Platform', 'iOS' is selected with a count of '1', and '1.0' is listed under 'Push'. The main content area shows the breadcrumb 'Home > mfp > com.worklight.MyBankApp > iOS 1.0' and a 'Delete version' button. The title is 'com.worklight.MyBankApp iOS v 1.0'. Below this are tabs for 'Management', 'Authenticity' (selected), 'Security', and 'Configuration Files'. The 'Application Authenticity' section has a sub-header 'Activating Application Authenticity protection ensures that the only the authentic application binary is able to communicate with the MobileFirst Server.' Below this is a message box with an orange warning icon: 'For security reasons, it is recommended that you enable Application Authenticity. Upload file to enable.' The status is 'Status: Disabled' with an 'Upload Authenticity File' button. A section titled 'Follow these steps to set up Application Authenticity' lists three steps: 1. Download the Application Authenticity tool, 2. Create the Authenticity File, and 3. Upload the Authenticity File.

Disabling Application Authenticity

In order to disable Application Authenticity, click the "Delete Authenticity File" button.

This screenshot shows the same MobileFirst Operations Console interface after the configuration change. The breadcrumb and title remain the same. The 'Authenticity' tab is still selected. A green success message box at the top states: 'Successfully uploaded the Authenticity File.' The status is now 'Status: Enabled' with a 'Delete Authenticity File' button. The 'Follow these steps to set up Application Authenticity' section remains unchanged, listing the same three steps.

Configuring Application Authenticity

The Application Authenticity out-of-the-box security check can be configured with the following property:

- `expirationSec`: Defaults to 3600 seconds / 1 hour. Defines the duration until the Authenticity token expires.

To configure the `expirationSec` property:

1. Load the MobileFirst Operations Console and navigate to **[your application] → Security → Security Check Configurations** and click on **Create New**.
2. Search for the "appAuthenticity" scope element.
3. Set a new value in seconds.

[Back](#)

Home > mfp > com.worklight.MyBankApp > iOS 1.0

[Delete version](#)

mfp

Applications

[Create new](#)

com.worklight.MyBankApp

Platform

[^](#) iOS

1

[1.0](#)

Push

com.worklight.MyBankApp iOS v 1.0

Configure Security Check Parameters

Scope element

Expiration (seconds) *

Expiration (seconds)

Default Value: 3600

[Add](#)[Cancel](#)

out-of-the-box security checks or

[Create New](#)[Create New](#)

You didn't create security check configuration yet

Get started by clicking "Create New"