

Application Authenticity

Overview

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

The predefined 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 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 8.1 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

Once an application has passed the Authenticity challenge, an Authenticity scope 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

To enable Application Authenticity in your Cordova or Native application, the application's binary file needs to be signed using the application-authenticity tool. Eligible binary files are: `ipa` for iOS, `apk` for Android and `appx` for Windows 8.1 Universal & Windows 10 UWP.

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

For example:

```
java -jar /Users/your-username/Desktop/application-authenticity-tool.jar /Users/your-username/Desktop/MyBankApp.ipa
```

The result of the command above is an `.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 navigation sidebar and click on the Authenticity menu item.
4. Click on "Upload Authenticity File" to upload the `.authenticity_data` file.

When the `.authenticity_data` file is uploaded, Application Authenticity is enabled.

MobileFirst Operations Console

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

com.worklight.MyBankApp iOS v 1.0

Management **Authenticity** Security Configuration Files

Application Authenticity

Activating Application Authenticity protection ensures that the only the authentic application binary is able to communicate with the MobileFirst Server.

For security reasons, it is recommended that you enable Application Authenticity. Upload file to enable.

Status: **Disabled** Upload Authenticity File

Follow these steps to set up Application Authenticity

- 1 Download the Application Authenticity tool
- 2 Create the Authenticity File
- 3 Upload the Authenticity File

Disabling Application Authenticity

To disable Application Authenticity, click the "Delete Authenticity File" button.

MobileFirst Operations Console

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

com.worklight.MyBankApp iOS v 1.0

Management **Authenticity** Security Configuration Files

Application Authenticity

Activating Application Authenticity protection ensures that the only the authentic application binary is able to communicate with the MobileFirst Server.

Successfully uploaded the Authenticity File.

Status: **Enabled** Delete Authenticity File

Follow these steps to set up Application Authenticity

- 1 Download the Application Authenticity tool
- 2 Create the Authenticity File
- 3 Upload the Authenticity File

Configuring Application Authenticity

The Application Authenticity predefined security check can be configured with the following property:

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

Once an authenticity check has been performed, it will not be performed again until the token has expired based on the set value.

To configure the `expirationInSec` 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.

The screenshot displays the MobileFirst Operations Console interface. The top navigation bar includes the 'MobileFirst Operations Console' title, an 'Analytics Console' link, a user profile 'Hello, demo', and an information icon. The main content area shows the breadcrumb 'Home > mfp > com.worklight.MyBankApp > iOS 1.0' and a 'Delete version' button. A modal dialog titled 'Configure Security Check Parameters' is open, featuring a 'Scope element' field with 'appAuthenticity', an 'Expiration (seconds)' field with '5000' (marked with a red asterisk), and a 'Default Value: 3600'. The dialog has 'Add' and 'Cancel' buttons. The background interface shows a sidebar with 'mfp' and 'Applications' sections, and a main area with 'com.worklight.MyBankApp' and 'Platform' filters set to 'iOS 1.0'. There are also 'Create new' and 'Create New' buttons visible in the background.