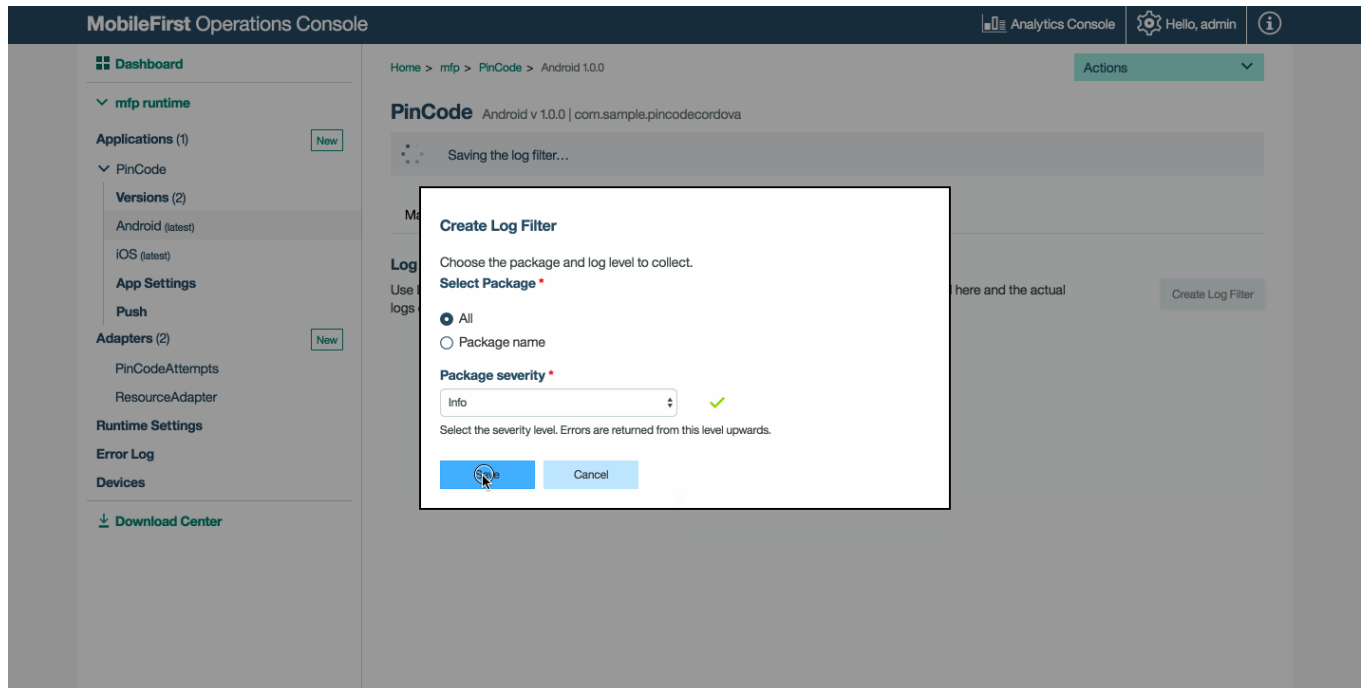# Configuring Log Filters

Overview

Administrators can control the MobileFirst client SDK log capture and levels from the **MobileFirst Operations Console → [your application] → [version] → Log Filters**.
Through `Log Filters` you can create a filter level that you can log at. The log level can be set globally (all logger instances) or for a specific package or packages.

For the application to fetch the configuration overrides that are set on the server, the `updateConfigFromServer` method must be called from a place in the code that is regularly run, such as in the app lifecycle callbacks.

Android

```
Logger.updateConfigFromServer();
```

iOS

```
[OCLogger updateConfigFromServer];
```

Cordova

```
WL.Logger.updateConfigFromServer();
```

Web

```
ibmmfpfanalytics.logger.updateConfigFromServer();
```

The `Logger` configuration values that the server returns take precedence over any values that are set on the client side. When the Client Log Profile is removed and the client tries to retrieve the Client Log Profile, the client receives an empty payload. In this case, the `Logger` configuration defaults to what was originally

configured on the client.

Forwarding server logs

The MobileFirst Operations Console also gives the server administrator the ability to persist logs and send those logs to the MobileFirst Analytics Console.

To forward server logs, navigate to the Runtime **Settings** screen and specify the logger package under **Additional Packages**.
The collected logs can then be viewed in the Analytics Console. This is useful for a user to take advantage of triaging adapter logs in the Analytics console without having to collect all server logs.

Last modified on