Operations and Analytics Consoles

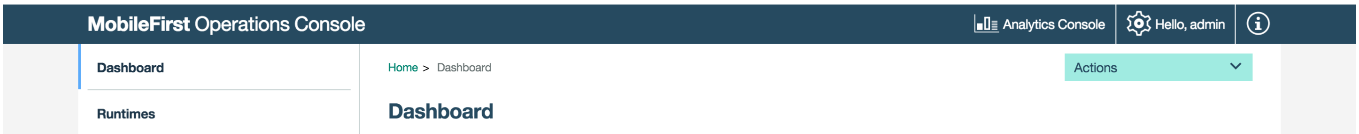
Overview

Configure the way the client works with the Analytics Server, configure and view a wide range of reports using the Operations and Analytics Consoles.

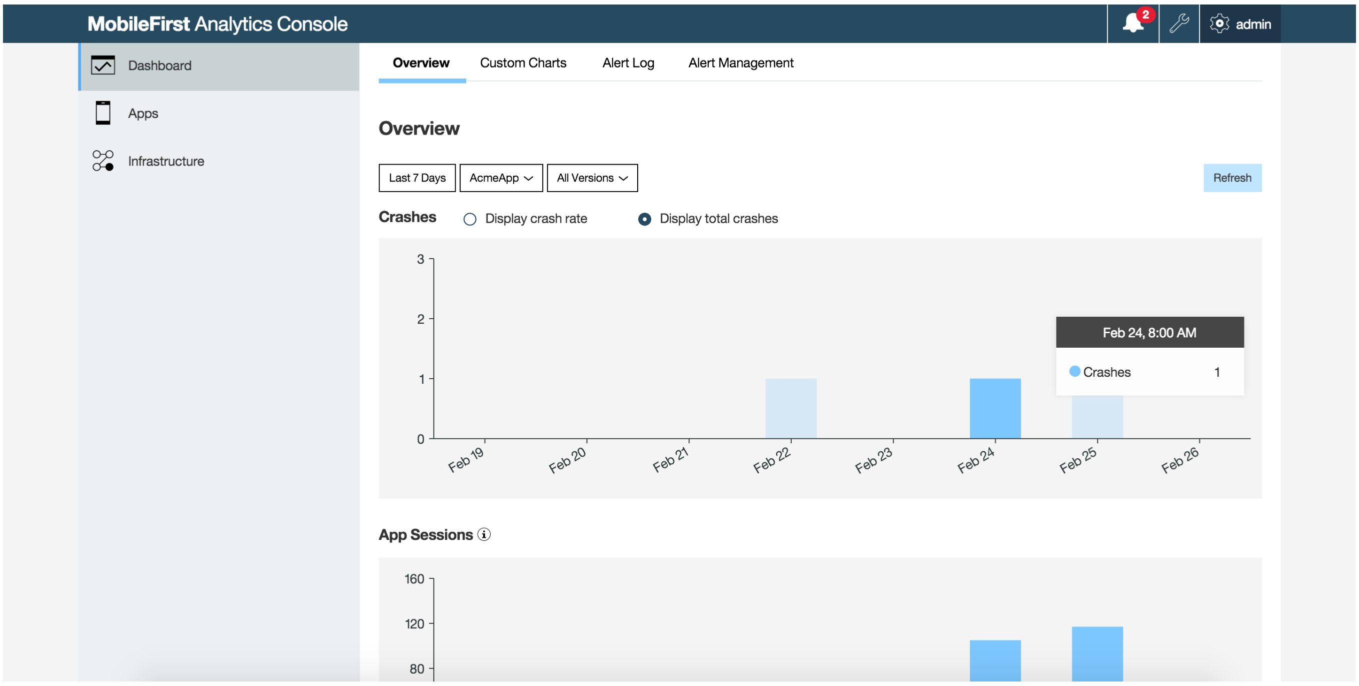
Analytics Console

From the Analytics Console, view and configure the Analytics reports. Manage alerts and view client logs.

You can open the Analytics Console from the MobileFirst Operations Console by clicking the **Analytics Console** link in the top-right navigation bar.



After navigating to the Analytics Console, the default **Dashboard** panel appears. If a client application has already sent logs and analytics data to the server, the relevant reports are populated. The **Apps** and **Infrastructure** can be chosen from the navigation bar.



Dashboard

In the **Dashboard** panel, you can review collected analytics data related to: application crashes, application sessions, and server processing time. Additionally you can create custom charts as well as manage alerts.

Apps

In the **Apps** panel, you can review in-depth analytics data related to: usage and devices (such as total device and app sessions, active users, app usage, new devices, model usage and operating system), as well as crash-related data. Search through client logs for specific apps and devices (**Apps → Client Log**

Search).

Infrastructure

In the **Infrastructure** panel you can review analytics data related to: session processing time, average request size, server requests, network requests, adapters response time, procedure response time, and size and adapters usage, as well as push notifications data such as notification requests and per mediator. You can also search through server logs.

> Learn more in the Analytics Workflows (../workflows/) tutorial.

Analytics features

App Analytics

From the **Apps → Usage and Devices**, you can view App Session charts and App Usage charts to find out which app is being used most frequently by your users.

Built-in Analytics

When you use the MobileFirst client SDK together with the MobileFirst Server, analytics data automatically gets collected for any request that your app makes to the MobileFirst Server. From **Dashboard → Overview** view basic device metadata that gets collected and reported to the MobileFirst Analytics Server.

Custom Analytics

You can have your app send custom data and create custom charts for your custom data.

> Learn how to send custom analytics in theAnalytics API (../analytics-api/) tutorial.

Custom charts

Custom charts allow you to visualize the collected analytics data in your analytics data store as charts that are not available by default in the MobileFirst Analytics Console (**Dashboard → Custom Charts**). This visualization feature is a powerful way to analyze business-critical data.

> Learn how to create custom charts in the Creating Custom Charts (custom-charts/) tutorial.

Manage alerts

Alerts provide a proactive means to monitor the health of your mobile apps without having to check the MobileFirst Analytics Console regularly.

From **Dashboard → Alert Management** you can configure thresholds which, if exceeded, trigger alerts to notify administrators. You can visualize the triggered alerts on the console or handle them by using a custom web hook. A custom web hook allows you to control who is notified when an alert is triggered, and how.

> Learn how to manage alerts in the Manage Alerts (alerts/) tutorial.

Monitor app crashes

App crashes are visualized on the Analytics Console (**Apps → Crashes**), where you can quickly view crashes and act on them accordingly. Crash logs are collected on the device by default and sent to the server once the application is running again. When crash logs are sent to the analytics server, they automatically populate the crash charts.

Monitor server and network data

The MobileFirst Analytics Console monitors network data when it is sent to the Analytics Server, and allows the user to query this information in different ways (**Infrastructure → Servers and Networks**).

Collect, search, and report on client logs

Client logs can be sent to the server and included in analytics reports.

To include logging information in a report:

1. In the Analytics Console, choose **Dashboard → Custom Charts**.
2. Choose **Client Logs** from the **Event Type** pull-down menu.

For more information on **Custom Charts**, see Creating Custom Charts (custom-charts/).

Logging data can be filtered. Log filters can be configured and saved on the Analytics Server, and then retrieved by client applications.

For information on configuring log filters, see Client log searches (log-filters/) tutorial.

For more information on sending logs from the client, see Client log collection (../../application-development/client-side-log-collection/).

MobileFirst Operations Console

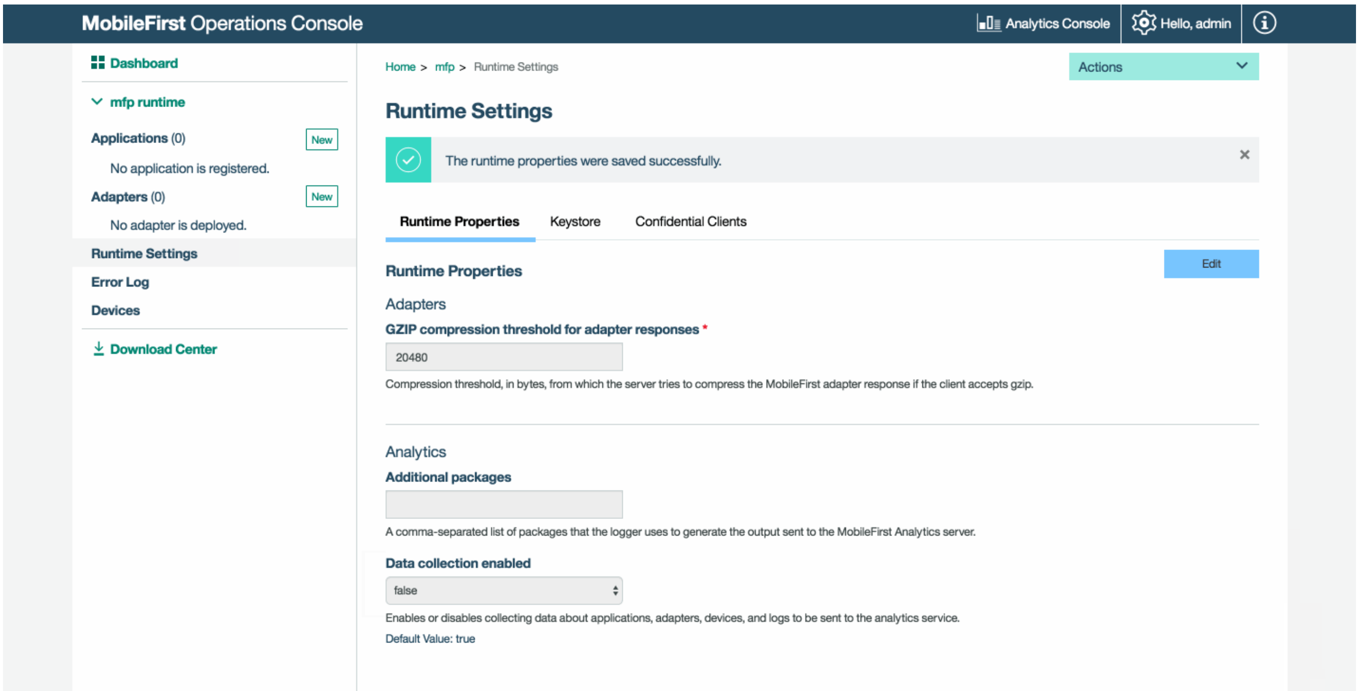
Configure and administer the Analytics Server with the Operations Console.

If you are in the Analytics Console, access the Operations Console by clicking the **Operations Console** button in the top navigation bar.

Enable/disable Analytics support

The collection of data for analysis by the Analytics server is enabled by default. You can disable it, for example to save processing time.

1. In the navigation sidebar, click on **Runtime settings**. To avoid inadvertent changes, runtime properties are displayed in read-only mode.
2. To make the settings editable, click the **Edit** button. If you logged in with a role other than *administrator* or *deployer*, the **Edit** button is not visible because you are not allowed to modify runtime properties.
3. From the **Data collection enabled** drop-down menu, select **false** to disable data collection.
4. Click **Save**.
5. Click the **Read Only** button to lock the properties again.



No application is registered.

No adapter is deployed.

The runtime properties were saved successfully.

Runtime Properties

Adapters

GZIP compression threshold for adapter responses *

20480

Compression threshold, in bytes, from which the server tries to compress the MobileFirst adapter response if the client accepts gzip.

Analytics

Additional packages

A comma-separated list of packages that the logger uses to generate the output sent to the MobileFirst Analytics server.

Data collection enabled

false

Enables or disables collecting data about applications, adapters, devices, and logs to be sent to the analytics service.

Default Value: true

Role-based access control

Content in the MobileFirst Analytics Console is restricted by predefined security roles.
The MobileFirst Analytics Console displays different content based on the security role of the logged-in user. The following table shows the security roles and their access to the MobileFirst Analytics Console.

Role	Role name	Viewing Access	Editing Access
Administrator	analytics_administrator	Everything.	Everything.
Infrastructure	analytics_infrastructure	Everything.	Everything.
Developer	analytics_developer	Everything except for the Administration pages.	Everything.
Support	analytics_support	Everything except for the Administration pages.	Everything.
Business	analytics_business	Everything except for the Administration and Infrastructure pages.	Everything.

> For information on setting up roles, see Configuring user authentication for MobileFirst Server administration (../../installation-configuration/production/server-configuration#configuring-user-authentication-for-mobilefirst-server-administration).

Elasticsearch

Behind the scenes, running search queries and storing data for Analytics is **Elasticsearch 1.5x**. Elasticsearch is a real-time distributed search and analytics engine that provides the ability to explore data at speed and at a scale. Elasticsearch is used for full-text search, structured search.

Elasticsearch is used for storing all mobile and server data in JSON format on the MobileFirst Operational Analytics server within Elasticsearch instances.
The Elasticsearch instances are queried in real-time in order to populate the MobileFirst Operational Analytics Console.

MobileFirst Operational Analytics does not hide any Elasticsearch functionality. All functionality is exposed and available to the user according to his knowledge of Elasticsearch.

For more information about Elasticsearch functionality beyond that predefined in MobileFirst Operational Analytics, see the Elasticsearch documentation.

> Read more in the Elasticsearch documentation (https://www.elastic.co/guide/en/elasticsearch/reference/1.5/index.html).

Elasticsearch properties

Elasticsearch properties are available through JNDI variables or environment entries.
One of the more useful JNDI properties to get started with viewing the Elasticsearch data is:

```
<jndiEntry jndiName="analytics/http.enabled" value="true"/>
```

This JNDI property allows you to view your Operational Analytics raw data in JSON format and to access your Elasticsearch instance through the port that is defined by Elasticsearch. The default port is 9500.

> **Note**: This setting is not secure and should not be enabled on a production environment.

Elasticsearch REST API

Being able to access an Elasticsearch instance provides the ability to run custom queries and view more detailed information about the Elasticsearch cluster.

Search and view data

You can view all your data by visiting the tenant's `_search` REST endpoint.

```
http://localhost:9500/*/_search
```

View cluster health

```
http://localhost:9500/_cluster/health
```

View information on current nodes

```
http://localhost:9500/_nodes
```

View the current mappings

```
http://localhost:9500/*/_mapping
```

> Elasticsearch exposes many more REST endpoints. To learn more, visit the Elasticsearch documentation (https://www.elastic.co/guide/en/elasticsearch/reference/1.5/index.html).

Related blog posts

- More on Instrumenting Custom Analytics (file:////home/travis/build/MFPSamples/DevCenter/_site/blog/2016/01/22/howto-custom-in-app-behavior-analytics/)
- More on Instrumenting Webhooks (file:////home/travis/build/MFPSamples/DevCenter/_site/blog/2015/10/19/using-mfp-adapters-endpoint-analytics-alerts-webhooks/)