MobileFirst Platform Foundation Operational Analytics

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/analytics/index.md) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

Overview

MobileFirst Platform Foundation Operational Analytics collects data about applications, adapters, devices, logs, and your own custom events to give a high-level view of the client interaction with the IBM MobileFirst Platform Server.

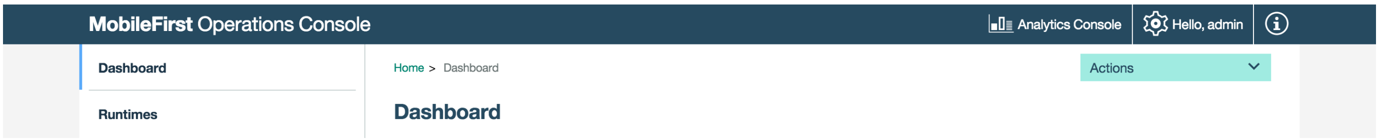
Operational Analytics is bundled by default as part of the MobileFirst Platform Foundation Development Kit (../setting-up-your-development-environment/mobilefirst-development-environment).

Jump to

- MobileFirst Analytics Console
- Elasticsearch
- Processing received logs
- Tutorials to follow next

MobileFirst Analytics Console

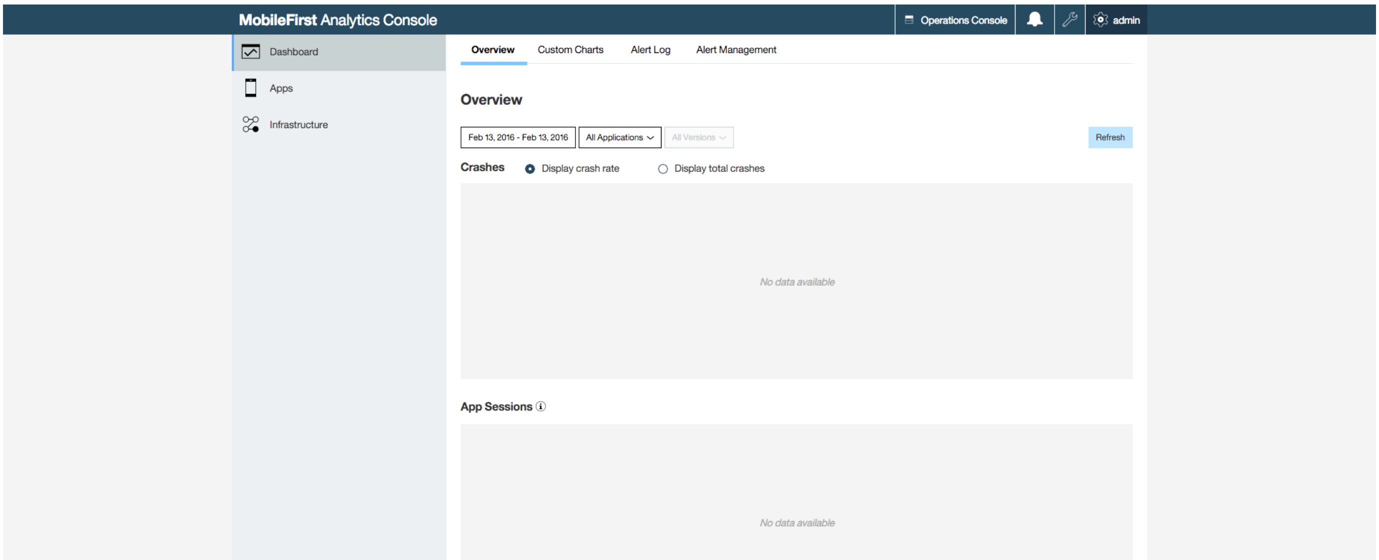
You can open the Analytics Console from the MobileFirst Operations Console.



From the Analytics Console you can then:

- See an overview of the available data
- Create custom charts
- Manage alerts
- what else?

Replace image with a console full of various data



Feb 13, 2016 - Feb 13, 2016

All Applications

All Versions

● Display crash rate ○ Display total crashes

Custom Charts

Add overview and then link to the specific tutorial

Manage Alerts

Add overview and then link to the specific tutorial

what-else-needs-to-be-added?

Elasticsearch

Behind the scenes, running search queries and storing data for Operational 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.

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

Elasticsearch is used for storing all mobile and server data in JSON format in the Operational Analytics server in 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. If knowledge how to fully utilize Elasticsearch, debug Elasticsearch, or optimize Elasticsearch instances is present, Operational Analytics does not prevent using it.

Need to mention how and where can the below be done. is it in the console or a browser or command line? where?

Elasticsearch properties

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

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

This JNDI property will allow you to view your Operational Analytics raw data in JSON format and allow you to access your Elasticsearch instance through the port defined by Elasticsearch (default port 9500).

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

Viewing data You can view all your data by visiting the tenant's search REST endpoint.
Being able to access an Elasticsearch instance provides the ability to run custom queries and view more detailed information about the Elasticsearch cluster.

```
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.

Processing received logs

The below text needs to be clarified to mention at a high-level what are the available options and paths to log data and display it. It should tell what is done, where does it go through, what does it pass through and its final destination. Also need to add a diagram

Logs get be processed in two ways:

1. Use the Analytics API set to create custom logging events
2. Develop and deploy a **JavaScript** adapter

WLClientLogReceiver JavaScript adapter

If you do not wish to use MobileFirst Operational Analytics, you can still collect and process client logs by creating an adapter. By default, MobileFirst Platform Foundation is configured to forward client logs to an adapter named `WLClientLogReceiver`, which exposes a method called `log`.

and...?? how do I use it?! should be considered to move to another tutorial as this is just the overview of the analytics category

If MobileFirst Operational Analytics and the adapter have both been configured, MobileFirst Platform Foudation sends logs to both endpoints. Using both may be useful if you want to take benefit of MobileFirst Operational Analytics log searching capabilities AND do some custom operations on client logs by using your own adapter.

Server preparation

review this text; I would not use the word "useless" It is useless to persistently capture log output in browsers and devices in production unless that captured data can be sent for diagnostic inspection. You can send captured data by calling the send method explicitly in your application.

this delves too much into techincalities that should be mentioned in specific tutorials. Consider pointing to a different tutorial and mention it there. You can turn the automatic behavior on or off with API calls that are specific to your environment, or by calling the `send` method explicitly in your application.

Tutorials to follow next