Using Analytics API in client applications

fork and edit tutorial (https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/#fork-destination-box) | report issue (https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/issues/new)

Overview

When collecting analytics data, the app writes to the devices file system until the `send` API is called. After the `send` API is called the app deletes the data from the file system and begins collecting again.

need to provide more text in the overview

Jump to:

- JavaScript API
- Java API
- Objective-C API

JavaScript API

JavaScript API is used in Cordova applications.

talks about a "custom event" - what's the "regular/default" event then?

Creating custom events in Cordova is simply just calling:

```
WL.Analytics.log("Custom event", {"property" : 'value'});
WL.Analytics.send();
```

Java API

JavaScript API is used in Android applications.

To collect custom analytics in Android you need to import WLAnalytics.
talks about a "custom event" - what's the "regular/default" event then?

```
import com.worklight.common.WLAnalytics;
```

Make sure you set the Logger context with.
Without a context you will not be able to collect analytics data.

```
WLAnalytics.setContext(this)
```

After setting the first two configurations you can start to log data like in the example below.

```
String json = "{\"property\" : \"value\"}";

try {
  WLAnalytics.log("Custom event", new JSONObject(json));
} catch (JSONException e) {
  e.printStackTrace();
}
```

Be sure to send your data with:

```
WLAnalytics.send();
```

Objective-C API

Objective-C API is used in iOS applications.

To collect custom analytics in iOS you need to import WLAnalytics.

```
#import "WLAnalytics.h";
```

After importing WLAnalytics you can now use the API to collect custom data like below:

```
NSDictionary *inventory = @{
    @"property" : @"value",
};

[[WLAnalytics sharedInstance] log:@"Custom event" withMetadata:inventory];
[[WLAnalytics sharedInstance] send];
```

Disabling log collection

It is possible to disable log collection using the following API:

JavaScript

```
WL.Analytics.disable();
```

Java

```
WLAnalytics.disable();
```

Objective-C

```
[[WLAnalytics sharedInstance] disable]
```