Logging in Android Applications

Overview

This tutorial provides the required code snippets in order to add logging capabilities in Android applications.

Anything else to add here?

Persisting log capture

MobileFirst Platform SDK for Android cannot persistently capture log data until the `com.worklight.common.Logger.setContext(Context)` method is called.

Need to mention where in the app lifecycle should the above code be used

Logging example

The below code snippet will output to the Android Studio LogCat view:

```
import com.worklight.common.Logger;

public class MathUtils{
  private static final Logger logger = Logger.getInstance(MathUtils.class.getName());
  public int sum(final int a, final int b){
  int sum = a + b;
  logger.debug("sum called with args " + a + " and " + b + ". Returning " + sum);
  return sum;
  }
}
```

Additional API methos for specific tasks

- Log capture is enabled by default. To turn log capture on or off:

  ```
  Logger.setCapture(false)
  ```

- The default capture level is DEBUG in development and FATAL in production. To control the capture level (verbosity):

  How do we know this? who/what and how sets this DEBUG and FATAL differences. Also, this collides what the text written in the overview of logging, where it is said that the logging can be controled from the analytics console.

  ```
  Logger.setLevel(Logger.FATAL)
  ```

- Log sending is enabled by default. To turn automatic log sending on or off:

```
Logger.setAutoSendLogs(false)
```

> For more information about the `Logger` API, see the API reference in the user documentation.