

Logging in Android Applications

fork and edit tutorial (<https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/using-the-mfpf-sdk/client-side-log-collection/android.md>) | report issue (<https://github.ibm.com/MFPSamples/DevCenter/issues/new>)

Overview

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

Prerequisite: Make sure to read the overview of client-side log collection (../).

Persisting Log Capture

MobileFirst Platform SDK for Android cannot persistently capture log data until the `com.worklight.common.Logger.setContext(Context)` method is called. This is best called in the `onCreate` method of your main Android activity.

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){
        logger.setLevel(LEVEL.DEBUG);
        int sum = a + b;
        logger.debug("sum called with args " + a + " and " + b + ". Returning " + sum);
        return sum;
    }
}
```

Additional API Methods For Specific Tasks

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

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

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

```
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.

