Logging in native Android (Java)

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/7.0/advanced-client-side-development/remote-controlled-client-side-log-collection/logging-android.html) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

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

Logging example for native Android (Java):

Outputs to LogCat

```
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;
   }
}
```

API calls 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):

`Logger.setLevel(Logger.LEVEL 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 (http://ibm.biz/knowctr#SSHS8R_7.0.0/com.ibm.worklight.apiref.doc/html/refjava-worklight-android-native/html/com/worklight/common/Logger.html), see the Reference part of the user documentation.