

Reading a WARC file

Out of date API

Note that this example uses an out of date API and will not compile against e.g. JWAT 0.9.1. See the [JWAT-Tools](#) source code for more up-to-date examples.

Insert short page description here

Not final!

TestWarc.java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.Collection;
import java.util.Iterator;

import org.jwat.warc.WarcReader;
import org.jwat.warc.WarcReaderFactory;
import org.jwat.warc.WarcRecord;
import org.jwat.warc.WarcValidationError;

public class TestWarc {

    static String warcFile =
"/home/nicl/Downloads/IAH-20080430204825-00000-blackbook.warc";
//static String warcFile = "/home/nicl/Downloads/MYWARC.warc";

    public static void main(String[] args) {
        File file = new File( warcFile );
        try {
            InputStream in = new FileInputStream( file );

            int records = 0;
            int errors = 0;

            WarcReader reader = WarcReaderFactory.getReader( in );
            WarcRecord record;

            while ( (record = reader.getNextRecord()) != null ) {
                printRecord(record);
                printRecordErrors(record);

                ++records;

                if (record.hasErrors()) {
                    errors += record.getValidationErrors().size();
                }
            }
        }
    }
}
```

```

        System.out.println("-----");
        System.out.println("          Records: " + records);
        System.out.println("          Errors: " + errors);
        reader.close();
        in.close();
    }
    catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public static void printRecord(WarcRecord record) {
    System.out.println("-----");
    System.out.println("          Version: " + record.bMagicIdentified + " " +
record.bVersionParsed + " " + record.major + "." + record.minor);
    System.out.println("          TypeIdx: " + record.warcTypeIdx);
    System.out.println("          Type: " + record.warcTypeStr);
    System.out.println("          Filename: " + record.warcFilename);
    System.out.println("          Record-ID: " + record.warcRecordIdUri);
    System.out.println("          Date: " + record.warcDate);
    System.out.println("Content-Length: " + record.contentLength);
    System.out.println(" Content-Type: " + record.contentType);
    System.out.println(" Truncated: " + record.warcTruncatedStr);
    System.out.println(" InetAddress: " + record.warcInetAddress);
    System.out.println(" ConcurrentTo: " + record.warcConcurrentToUriList);
    System.out.println(" RefersTo: " + record.warcRefersToUri);
    System.out.println(" TargetUri: " + record.warcTargetUriUri);
    System.out.println(" WarcInfo-Id: " + record.warcWarcInfoIdUri);
    System.out.println(" BlockDigest: " + record.warcBlockDigest);
    System.out.println(" PayloadDigest: " + record.warcPayloadDigest);
    System.out.println("IdentPloadType: " +
record.warcIdentifiedPayloadType);
    System.out.println("          Profile: " + record.warcProfileStr);
    System.out.println("          Segment#: " + record.warcSegmentNumber);
    System.out.println(" SegmentOrg-Id: " + record.warcSegmentOriginIdUrl);
    System.out.println("SegmentTLength: " + record.warcSegmentTotalLength);
}

public static void printRecordErrors(WarcRecord record) {
    if (record.hasErrors()) {
        Collection<WarcValidationError> errorCol = record.getValidationErrors();
        if (errorCol != null && errorCol.size() > 0) {
            Iterator<WarcValidationError> iter = errorCol.iterator();
            while (iter.hasNext()) {
                WarcValidationError error = iter.next();
                System.out.println( error.error );
                System.out.println( error.field );
                System.out.println( error.value );
            }
        }
    }
}

```

}
}
}
}

}