Overview

Welcome to the Java Web Archive Toolkit

This wiki describes the overall packages and also includes some details on how the main classes are implemented.

The JWAT code was originally intended for use only in a number of JHove2 modules, but since the classes can be of use outside the JHove2 project, an independent project was created.

Note: Even though the repository is about 80mb, most of that is test data. The libraries themselves are very small!

Overview:

The goal of the **JWAT library** was to make a small package to **read**, **write** and **validate WARC**, **ARC** and **GZip** files.

All the parsers were implemented on the premise that input data would be supplied in the form of streams and not files.

So the basic operation of parsing and validating a file is a **sequential operation** where each record and its payload is only read once.

This is also the case when parsing/validating compressed **WARC/ARC** files where each record is GZip'ed. In which case the compressed data can also only be processed sequentially.

It is however possible to random access individual **WARC/ARC/GZip** records when working with the logical files and using a file offset. This is possible by using a simple **RandomAccessFileInputStream** present in the common package or using any other type of stream where you control exactly which records to present to a reader.

Since this project is intended to be of general use it also includes writers for **WARC**/**ARC**/**GZip**.

Features

- GZip support.
    - Reader with validation.
    - Writer with validation.
    - (Multi-file) GZip validating decompressor/compressor.
    - GZip Input/Output wrapper streams.
    - Uses the Inflater and Deflater classes directly.
- ARC support.
    - Reader with validation.
    - Writer with validation.
    - Supports GZip.
- WARC support.
    - Reader with validation.
    - Writer with validation.
    - Supports GZip.
- Encoding supported:
    - Base64, Base32 and Base16.
    - ISO8859-1.
    - UTF-8.
    - QuotedString.
    - EncodedWords.
- Useful common classes:
    - Advanced header line reader.
    - HttpHeader request/response parser/validation.
    - Content-Type parser/validation.
    - URI validation based on different profiles.
    - Various special purpose stream implementations.