jwat-common

common package

org.jwat.common:

As the package name indicates this package includes various classes of general use but also more specific ARC/WARC classes. The classes can be classified as follows.

String encoding

The base classes are defined in various RFCs and are commonly used across the internet in many different contexts. Basically an input comprising of an array of 8bit characters is converted into a string of printable characters.

- **Base64.java**: Uses an alphabet of 64 characters and is the most widely used.
- **Base32.java**: Uses an alphabet of 32 characters and seems to be the default encoding for WARC digests.
- **Base16.java**: Uses an alphabet of 16 characters and is also more commonly called hexadecimal strings or just hex for short.
- **Base2.java**: Uses only 0s and 1s and represents the 8bit values as binary string representations.

InputStream / StringReader

- **ByteCountingInputStream.java**: An extended InputStream modified to keep track of the number of consumed bytes.
- **ByteCountingPushBackInputStream.java**: An extended Pushback Inputstream which also keeps track of the actual number of consumed bytes.
- **DigestInputStreamNoSkip.java**: An extended DigestInputStream which overrides the skip method to perform reads.
- **FixedLengthInputStream.java**: An InputStream wrapper with a fixed amount of data available, which must be consumed by either reading/skipping it, or ultimately be skipped when it is closed, if data is still available.
- **MaxLengthRecordingInputStream.java**: An Inputstream wrapper which can only consumed a maximum amount of data which in turn is recorded internally and available as a byte array afterwards.
- **CharCountingStringReader.java**: A StringReader that uses a String as Input and also keeps track of the number of consumed chars.

Common header and payload classes used by both ARC and WARC

- **ContentType.java**: Parses and validates a content-type header with optional parameters.
- **IPAddressParser.java**: Parses and validates an IPv4 or IPv6 address using regular expressions.

Digest.java
HeaderLine.java

- **HttpResponse.java**: Identifies and encapsulates valid http header blocks. The payload is accessible though an InputStream with optional digest value computation.
- **Payload.java**: Encapsulates an ARC/WARC payload and optionally computes a digest value.
- **PayloadOnClosedHandler.java**: An interface that must be implemented to receive notice of a payloads closure.

RandomAccessFile InputStream wrappers

- **RandomAccessFileInputStream.java**: Used to access a File as an InputStream. All the RandomAccessFile methods can be used, like seek to re-position the stream.
- **RandomAccessFileOutputStream.java**: Used to access a File as an OutputStream.