public static void main(String[] args) throws Exception

{

RandomAccessFile raf = new RandomAccessFile("test.csv", "r");

long numSplits = 10; //from user input, extract it from args

long sourceSize = raf.length();

long bytesPerSplit = sourceSize/numSplits ;

long remainingBytes = sourceSize % numSplits;

int maxReadBufferSize = 8 \* 1024; //8KB

for(int destIx=1; destIx <= numSplits; destIx++) {

BufferedOutputStream bw = new BufferedOutputStream(new FileOutputStream("split."+destIx));

if(bytesPerSplit > maxReadBufferSize) {

long numReads = bytesPerSplit/maxReadBufferSize;

long numRemainingRead = bytesPerSplit % maxReadBufferSize;

for(int i=0; i<numReads; i++) {

readWrite(raf, bw, maxReadBufferSize);

}

if(numRemainingRead > 0) {

readWrite(raf, bw, numRemainingRead);

}

}else {

readWrite(raf, bw, bytesPerSplit);

}

bw.close();

}

if(remainingBytes > 0) {

BufferedOutputStream bw = new BufferedOutputStream(new FileOutputStream("split."+(numSplits+1)));

readWrite(raf, bw, remainingBytes);

bw.close();

}

raf.close();

}

static void readWrite(RandomAccessFile raf, BufferedOutputStream bw, long numBytes) throws IOException {

byte[] buf = new byte[(int) numBytes];

int val = raf.read(buf);

if(val != -1) {

bw.write(buf);

}

}

<https://stackoverflow.com/questions/19177994/java-read-file-and-split-into-multiple-files>

to get hash value: MD5

MessageDigest md = MessageDigest.getInstance("MD5");

try (InputStream is = Files.newInputStream(Paths.get("file.txt"));

DigestInputStream dis = new DigestInputStream(is, md))

{

/\* Read decorated stream (dis) to EOF as normal... \*/

}

byte[] digest = md.digest();

Use [DigestUtils](http://commons.apache.org/proper/commons-codec/apidocs/org/apache/commons/codec/digest/DigestUtils.html) from [Apache Commons Codec](http://commons.apache.org/codec/) library:

FileInputStream fis = new FileInputStream(new File("foo"));

String md5 = org.apache.commons.codec.digest.DigestUtils.md5Hex(fis);

fis.close();

Java code to get hash value

import java.io.\*;

import java.security.MessageDigest;

public class MD5Checksum {

public static byte[] createChecksum(String filename) throws Exception {

InputStream fis = new FileInputStream(filename);

byte[] buffer = new byte[1024];

MessageDigest complete = MessageDigest.getInstance("MD5");

int numRead;

do {

numRead = fis.read(buffer);

if (numRead > 0) {

complete.update(buffer, 0, numRead);

}

} while (numRead != -1);

fis.close();

return complete.digest();

}

// see this How-to for a faster way to convert

// a byte array to a HEX string

public static String getMD5Checksum(String filename) throws Exception {

byte[] b = createChecksum(filename);

String result = "";

for (int i=0; i < b.length; i++) {

result += Integer.toString( ( b[i] & 0xff ) + 0x100, 16).substring( 1 );

}

return result;

}

public static void main(String args[]) {

try {

System.out.println(getMD5Checksum("apache-tomcat-5.5.17.exe"));

// output :

// 0bb2827c5eacf570b6064e24e0e6653b

// ref :

// http://www.apache.org/dist/

// tomcat/tomcat-5/v5.5.17/bin

// /apache-tomcat-5.5.17.exe.MD5

// 0bb2827c5eacf570b6064e24e0e6653b \*apache-tomcat-5.5.17.exe

}

catch (Exception e) {

e.printStackTrace();

}

}

}

For SHA hash value (meta data)

public class ToHash {

public static void main(String[] args) {

byte[] data = "test".getBytes("UTF8");

MessageDigest digest = MessageDigest.getInstance("SHA-256");

byte[] hash = digest.digest(data);

System.out.println(new BASE64Encoder().encode(hash));

}

}

public class ToHash {

public static void main(String[] args) throws NoSuchAlgorithmException, UnsupportedEncodingException, FileNotFoundException, IOException {

// TODO code application logic here

// The name of the file to open.

String fileName = "C:\\Users\\ghasemi\\Desktop\\1.png";

BufferedReader br = null;

try {

String sCurrentLine;

br = new BufferedReader(new FileReader(fileName));

while ((sCurrentLine = br.readLine()) != null) {

byte[] data = sCurrentLine.getBytes("UTF8");

System.out.println(new BASE64Encoder().encode(data));

}

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

if (br != null) {

br.close();

}

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

}

To upload file in Google API:

<https://developers.google.com/drive/v3/web/quickstart/java>

examples:

https://github.com/google/google-api-java-client-samples/blob/master/drive-cmdline-sample/src/main/java/com/google/api/services/samples/drive/cmdline/DriveSample.java