链滴

HashBufferedInputStream: 实现 InputStream 在使用中同时计算 Hash 值

作者：vanlin

原文链接：https://ld246.com/article/1494830989979

来源网站：链滴

需求：

- 文件上传的同时计算 hash值
- 输入流使用是计算hash

实现：

上面两个需求其实都是一致的 都是 处理 InputStream 同时计算 Hash值

我们知道 Java IO是使用的 装饰模式， 那么我们只需要对 BufferedInputStream 进行封装即可

重点是Override public int read(final byte[] anB) throws IOException

这样仅支持连续完整的使用 public int read(final byte[] anB) throws IOException 读取完 流的情况所以还是有很大局限性。

```
package me.vanlin.test;

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.HashSet;
import java.util.Set;

import org.apache.commons.io.IOUtils;

/**
 * @author liuwl
 *
 */
public class HashBufferedInputStream extends BufferedInputStream {
    private final MessageDigest messageDigest;

    private static final Set<String> algorithms = new HashSet<>();

    static {
        algorithms.add("MD5");
        algorithms.add("SHA-1");
        algorithms.add("SHA-256");
        algorithms.add("SHA-512");
    }

    /**
     * @param anIn
     * @throws NoSuchAlgorithmException
     */
```

```
    public HashBufferedInputStream(final InputStream anIn, final String algorithm) throws NoS
chAlgorithmException {
        super(anIn);
        if (!algorithms.contains(algorithm)) {
            throw new RuntimeException("algorithm not support");
        }
        messageDigest = MessageDigest.getInstance(algorithm);
    }

    /**
     * @param anIn
     * @param anSize
     * @throws NoSuchAlgorithmException
     */
    public HashBufferedInputStream(final InputStream anIn, final int anSize, final String algorit
m) throws NoSuchAlgorithmException {
        super(anIn, anSize);
        if (!algorithms.contains(algorithm)) {
            throw new RuntimeException("algorithm not support");
        }
        messageDigest = MessageDigest.getInstance(algorithm);
    }

    @Override
    public int read(final byte[] anB) throws IOException {
        final int result = super.read(anB);
        if (result != -1) {
            messageDigest.update(anB, 0, result);
        }
        return result;
    }

    public String getHash() {
        final byte byteBuffer[] = messageDigest.digest();
        final StringBuffer hexStr = new StringBuffer();
        for (final byte element : byteBuffer) {
            final String hex = Integer.toHexString(0xff & element);
            if (hex.length() == 1) {
                hexStr.append('0');
            }
            hexStr.append(hex);
        }
        return hexStr.toString();
    }

    public static void main(final String[] args) throws NoSuchAlgorithmException, IOException {
        final File file = new File("E:\\me\\front-end.tar.gz");
        try (final FileInputStream inputStream = new FileInputStream(file);
                final HashBufferedInputStream hashBufferedInputStream = new HashBufferedInput
tream(inputStream, "MD5");
                final FileOutputStream outputStream = new FileOutputStream("E://me//testtesttest.
ar.gz");
                final BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(o
tputStream);) {
```

```
            IOUtils.copy(hashBufferedInputStream, bufferedOutputStream);
            System.out.println(hashBufferedInputStream.getHash());
        }

    }
}
```

代码如有纰漏请不吝赐教。