Java MD5 Hashing Example

By mkyong (http://www.mkyong.com/author/mkyong/) | February 23, 2010 | Updated : August 30, 2012

> MD5 is one in a series of message digest algorithms designed by Professor Ronald Rivest of MIT (Rivest, 1994). When analytic work indicated that MD5's predecessor MD4 was likely to be insecure, MD5 was designed in 1991 to be a secure replacement. (Weaknesses were indeed later found in MD4 by Hans Dobbertin.)

MD5 is a widely used hashing algorithm in many companies and industries, here are two examples for the MD5 implementation.

1. File checksum with MD5

It will use MD5 hashing algorithm to generate a checksum for file "c:\\loging.log".

```
package com.mkyong.test;

import java.io.FileInputStream;
import java.security.MessageDigest;

public class MD5CheckSumExample
{
    public static void main(String[] args)throws Exception
    {
        MessageDigest md = MessageDigest.getInstance("MD5");
        FileInputStream fis = new FileInputStream("c:\\loging.log");

        byte[] dataBytes = new byte[1024];

        int nread = 0;
        while ((nread = fis.read(dataBytes)) != -1) {
          md.update(dataBytes, 0, nread);
        };
        byte[] mdbytes = md.digest();

        //convert the byte to hex format method 1
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < mdbytes.length; i++) {
          sb.append(Integer.toString((mdbytes[i] & 0xff) + 0x100, 16).substring(1));
        }

        System.out.println("Digest(in hex format):: " + sb.toString());

        //convert the byte to hex format method 2
        StringBuffer hexString = new StringBuffer();
        for (int i=0;i<mdbytes.length;i++) {
            String hex=Integer.toHexString(0xff & mdbytes[i]);
            if(hex.length()==1) hexString.append('0');
            hexString.append(hex);
        }
        System.out.println("Digest(in hex format):: " + hexString.toString());
    }
}
```

Output

```
Digest(in hex format):: e72c504dc16c8fcd2fe8c74bb492affa
Digest(in hex format):: e72c504dc16c8fcd2fe8c74bb492affa
```

2. Hashing String with MD5

It will use MD5 hashing algorithm to generate a hash value for a password "123456".

```
package com.mkyong.test;

import java.security.MessageDigest;

public class MD5HashingExample
{
    public static void main(String[] args)throws Exception
    {
        String password = "123456";

        MessageDigest md = MessageDigest.getInstance("MD5");
        md.update(password.getBytes());

        byte byteData[] = md.digest();

        //convert the byte to hex format method 1
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < byteData.length; i++) {
         sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
        }

        System.out.println("Digest(in hex format):: " + sb.toString());

        //convert the byte to hex format method 2
        StringBuffer hexString = new StringBuffer();
        for (int i=0;i<byteData.length;i++) {
            String hex=Integer.toHexString(0xff & byteData[i]);
            if(hex.length()==1) hexString.append('0');
            hexString.append(hex);
        }
        System.out.println("Digest(in hex format):: " + hexString.toString());
    }
}
```

Output

```
Digest(in hex format):: e10adc3949ba59abbe56e057f20f883e
Digest(in hex format):: e10adc3949ba59abbe56e057f20f883e
```

Reference

1. http://en.wikipedia.org/wiki/MD5 (http://en.wikipedia.org/wiki/MD5)
2. http://forums.sun.com/thread.jspa?threadID=5169003 (http://forums.sun.com/thread.jspa?threadID=5169003)

Tags : hashing (http://www.mkyong.com/tag/hashing/)
java (http://www.mkyong.com/tag/java/) md5 (http://www.mkyong.com/tag/md5/)

Share this article on

Twitter (https://twitter.com/intent/tweet?text=Java MD5 Hashing Example&url=http://www.mkyong.com/java/java-md5-hashing-example/&via=mkyong) Facebook (https://www.facebook.com/sharer/sharer.php?u=http://www.mkyong.com/java/java-md5-hashing-example/) Google+ (https://plus.google.com/share?url=http://www.mkyong.com/java/java-md5-hashing-example/)

Reader also read :

Java – How to split a string (http://www.mkyong.com/java/java-how-to-split-a-string/)

Java – How to delay few seconds (http://www.mkyong.com/java/java-how-to-delay-few-seconds/)

Java – Check if key exists in HashMap (http://www.mkyong.com/java/java-check-if-key-exists-in-hashmap/)

Java – Convert String to int (http://www.mkyong.com/java/java-convert-string-to-int/)

Java – Display double in 2 decimal points (http://www.mkyong.com/java/java-display-double-in-2-decimal-points/)

About the Author

mkyong

Founder of Mkyong.com (http://mkyong.com) and HostingCompass.com (http://hostingcompass.com), love Java and open source stuff. Follow him on Twitter (https://twitter.com/mkyong), or befriend him on Facebook (http://www.facebook.com/java.tutorial) or Google Plus (https://plus.google.com/110948163568945735692?rel=author). If you like my tutorials, consider make a donation to these charities (http://www.mkyong.com/blog/donate-to-

> charity/).

Comments

19 Comments Mkyong.com 1 Login

Recommend 1 Share Sort by Best

Join the discussion…

Wilson • 3 years ago

Is there any way convert the digested "e10adc3949ba59abbe56e057f20f883e" key to original password "123456". If you know please let us know.

Thanks
Wilson.

4 • Reply • Share ›

Saifur Rahman Mohsin Wilson • a year ago

You can do this by generating a rainbow table and then comparing the hash with original text. However longer texts will not give the correct output for obvious reasons..!

1 • Reply • Share ›

Nishu • 9 months ago

Its giving a different hash every time with the same input file which should not be the case with MD5 algorithm. I think there is some problem in this code

• Reply • Share ›

Ahmed ElZayady • 10 months ago

Hi .. The string method above isn't working fine if the string contains a £ sign, don't know why. On other hand, the function provided in this link http://www.asjava.com/core-jav... did work fine with £. May be it is something you might be interested to look into.

• Reply • Share ›

Murali Mohan • a year ago

The world is a fast-changing place. Check out Apache Shiro's cryptography capabilities.

https://www.youtube.com/watch?...

• Reply • Share ›

Napster85 • 2 years ago

What about if you use this:

sb.append(String.format("%02X", buffer[i]));

;)

• Reply • Share ›

itsvenkis • 2 years ago

Hi!!! I really unable to understand why you had

Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1)

instead of

Integer.toString((byteData[i] & 0xff), 16)

If it is just for padding zeros you could have checked the length and added a zero. May be I am missing the whole point. Can you pelase help me understand?

^ | ˅ • Reply • Share ›

none → itsvenkis • 2 years ago

Yes, it's for padding, and It's just easier syntax than introducing conditional branching.

^ | ˅ • Reply • Share ›

elton • 2 years ago

hello sir,
i want a simple java code to calculate md5 of any input i give from a keyboard

^ | ˅ • Reply • Share ›

gli00001 • 3 years ago

new BigInteger(1, md.digest()).toString(16) should save the loop using stringBuffer

^ | ˅ • Reply • Share ›

siri • 3 years ago

I have 100 file (which are not stored on my computer)that are located on a server and I have path to each one of them, I need a code to calculate the MD5 of each one of the and store it in an excel sheet individually.

^ | ˅ • Reply • Share ›

natesan • 3 years ago

Dear Sir,
in this example (ff) variable showing some errors.how to rectify the error. thank you sir.the error was ff cannot be resolved to a variable.

^ | ˅ • Reply • Share ›

nordin • 3 years ago

md5 is useful when hashing filenames for some reason. I'll use this snippet of code to hash imagefile names, it's temporary for a few weeks and than I'll remove them. The advantage is that this hash method is less cpu intensive and the chances to have the same hash values are very very small. Furthermore, this is a nice simple example, so using a sha-1 instead of md5 is almost the same. Thank you for your snippet :-)

^ | ˅ • Reply • Share ›

Cristian • 3 years ago

Hello, I am wondering why you would use MD5, an algorithm thats has already been broken instead of something like GrandCentral which creates a password digest based on the time of day using SHA-512. GrandCentral also includes the SimpleCrypto class which can generate hashes

using SHA-512. GrandCentral also includes the SimpleCrypto class which can generate hashes and checksums using MD5, SHA1, SHA-256, SHA-384 and SHA-512. Just a suggestion.
http://code.google.com/p/grand...

• Reply • Share ›

mkyong ➦ Cristian • 3 years ago

MD5 is not broken, it has chance (really small percentage to generate same hash), SHA is always recommended. But, due to the popularity of MD5, many are still using it. This is an example to show you how to use MD5 in Java, again SHA is always recommended. And thanks for your suggested GrandCentral library, it look promising :)

• Reply • Share ›

Cristian Rivera ➦ mkyong • 3 years ago

Yes, I'm sorry for my poor use of words. I meant that yes it is widely uses do to it's popularity but there is still a better chance of cracking an MD5 hash rather than an SHA hash. Also thank you for your comment about GrandCentral I am developing it to provide developers with and easy to use tool that they can rely on to perform certain tasks easily. I will be updating later this weekend to v1.2 if you are interested.

• Reply • Share ›

santosh • 5 years ago

sir,
give me suggestion for password hashing using md 5 algorithm in java.

• Reply • Share ›

Rocky Madden • 6 years ago

Been moving towards always using SHA hashing myself, especially with recent CPU processor support/improvements being added each generation. Trusty old MD5 doesn't let down most of the time though.

• Reply • Share ›

mkyong ➦ Rocky Madden • 6 years ago

ya, SHA is recommended because MD5 will causing the collision issue (same checksum for different files).

• Reply • Share ›

ALSO ON **MKYONG.COM** WHAT'S THIS?

Java – Math.pow example

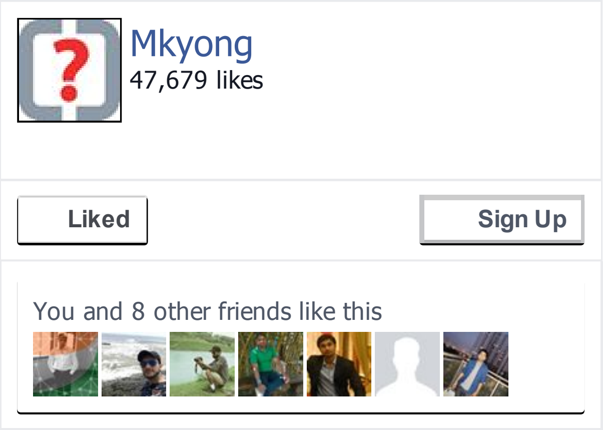
2 comments • 8 months ago

Avat **Goran Qaqnass** — thanks

Maven Jetty Plugin Examples

2 comments • 6 months ago

Avat **Ucup Timposu** — thanks sir



Mkyong
47,679 likes

You and 8 other friends like this

Favorites Links

Android Getting Started (http://developer.android.com/training/index.html)

Google App Engine – Java (https://cloud.google.com/appengine/docs/java/)

Spring 2.5.x Documentation (http://docs.spring.io/spring/docs/2.5.x/reference/index.html)

Spring 3.2.x Documentation (http://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/)

Spring 4.1.x Documentation (http://docs.spring.io/spring/docs/4.1.x/spring-framework-reference/html/)

Java EE 5 Tutorial (http://docs.oracle.com/javaee/5/tutorial/doc/docinfo.html)

Java EE 6 Tutorial (http://docs.oracle.com/javaee/6/tutorial/doc/docinfo.html)

Java EE 7 Tutorial (https://docs.oracle.com/javaee/7/tutorial/index.html)

Java 6 API (http://docs.oracle.com/javase/6/docs/api/overview-summary.html)

Java 7 API (http://docs.oracle.com/javase/7/docs/api/overview-summary.html)

Java 8 API (http://docs.oracle.com/javase/8/docs/api/overview-summary.html)

JSF Home Page (https://javaserverfaces.java.net/)

JSP Home Page (https://jsp.java.net/)

Maven Central Repository (http://search.maven.org/)

Hibernate ORM (http://hibernate.org/orm/)

JAX-WS Home Page (https://jax-ws.java.net/)

JAX-RS Home Page (Jersey) (https://jax-ws.java.net/)

Partners & Bookmarks

Java Code Geeks (http://www.javacodegeeks.com/)

TestNG Founder (http://beust.com/weblog/)

DZone (https://dzone.com)

About Mkyong.com

Mkyong.com is for Java and J2EE developers, all examples are simple and easy to understand, and well tested in my development environment.

Mkyong.com is created, written by, and maintained by Yong Mook Kim, aka Mkyong. It is built on WordPress (https://wordpress.org/), hosted by Liquid Web (http://mkyong.com/go/liquidweb/), and the caches are served by CloudFlare CDN.

Copyright © 2008-2015 Mkyong.com, all rights reserved. Privacy Policy (/privacy-policy/)