

Java SHA Hashing Example

By mkyong (<http://www.mkyong.com/author/mkyong/>) | February 24, 2010 | Updated : August 30, 2012 | Viewed : 255,886 times

The SHA hash functions are a set of cryptographic hash functions designed by the National Security Agency (NSA) and published by the NIST as a U.S. Federal Information Processing Standard. SHA stands for Secure Hash Algorithm. The three SHA algorithms are structured differently and are distinguished as SHA-0, SHA-1, and SHA-2. The SHA-2 family uses an identical algorithm with a variable digest size which is distinguished as SHA-224, SHA-256, SHA-384, and SHA-512.

SHA-2 is believe the most secure hashing algorithm as this article is written, here are few examples for the SHA implementation. The possible **MessageDigest algorithm** are SHA-1, SHA-256, SHA-384, and SHA-512, you can check the reference for the detail.

1. File checksum with SHA-256

It will use SHA-256 hashing algorithm to generate a checksum for file "c:\\logging.log".

Java

```
package com.mkyong.test;

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

public class SHACheckSumExample
{
    public static void main(String[] args)throws Exception
    {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        FileInputStream fis = new FileInputStream("c:\\\\logging.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("Hex format : " + sb.toString());

        //convert the byte to hex format method 2
        StringBuffer hexString = new StringBuffer();
        for (int i=0;i<mdbytes.length;i++) {
            hexString.append(Integer.toHexString(0xFF & mdbytes[i]));
        }

        System.out.println("Hex format : " + hexString.toString());
    }
}
```

Output

```
Hex format : 21a57f2fe765e1ae4a8bf15d73fc1bf2a533f547f2343d12a499d9c0592044d4
Hex format : 21a57f2fe765e1ae4a8bf15d73fc1bf2a533f547f2343d12a499d9c0592044d4
```

Bash

2. Hashing String with SHA-256

It will use SHA-256 hashing algorithm to generate a hash value for a password "123456".

Java

```
package com.mkyong.test;

import java.security.MessageDigest;

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

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        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("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("Hex format : " + hexString.toString());
    }
}
```

Output

```
Hex format : 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92
Hex format : 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92
```

Bash

Reference

1. http://en.wikipedia.org/wiki/SHA_hash_functions
(http://en.wikipedia.org/wiki/SHA_hash_functions)
2. <http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html#AppA>
(<http://java.sun.com/j2se/1.4.2/docs/guide/security/CryptoSpec.html#AppA>)

Tags : [hashing \(http://www.mkyong.com/tag/hashing/\)](http://www.mkyong.com/tag/hashing/)
[java \(http://www.mkyong.com/tag/java/\)](http://www.mkyong.com/tag/java/) [sha \(http://www.mkyong.com/tag/sha/\)](http://www.mkyong.com/tag/sha/)

Share this article on

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

Reader also read :



Java – How to override equals and hashCode
(<http://www.mkyong.com/java/java-how-to-overrides-equals-and-hashcode/>)



Java builder design pattern example
(<http://www.mkyong.com/java/java-builder-design-pattern-example/>)



Java – How to join Arrays
(<http://www.mkyong.com/java/java-how-to-join-arrays/>)



Java – How to print an Array
(<http://www.mkyong.com/java/java-how-to-print-an-array/>)



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

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

33 Comments

Mkyong.com

1 Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾



Join the discussion...



StYleZ • 4 years ago

Hi @ll,

1. File checksum with SHA-256 -> //convert the byte to hex format method 2 = doesn't work correntcly!

Here my output:

Hex format : da84e5104ec02982515127adda821ffc533acf7f07bd9b5839f31239e888feea

Hex format : da84e5104ec02982515127adda821ffc533acf7f7bd9b5839f31239e888feea

As you may have noticed there is as 0("zero") missing.

Methond 1 is fine.

Thx for this Tutorail - it helped me alot!

Greets

StYleZ

2 ^ | ▾ • Reply • Share ›



ChaZ ➔ StYleZ • 2 years ago

What was your input?

^ | ▾ • Reply • Share ›



FixingError ➔ ChaZ • 18 days ago

The reason that the 2nd method would output wrong because it eats up leading zero of any hex-pair value. For example, a hex value of x07 will be output as 7 in the string. As a result, the 2nd method will output a wrong result for a hex value from x00 to x0F. Hope this help people in the future.

^ | ▾ • Reply • Share ›



John • 5 years ago

why don't you use DigestInputStream?

2 ^ | ▾ • Reply • Share ›



Zeza • 2 years ago

... ..



Hi all,

i'm very new in programming,

i have urgent task: i have to send message signed by some key

i have an example on Csharp but i need it written on Java, please help me!

```
using System;  
using System.Collections.Generic;  
using System.Windows.Forms;  
using System.Text;  
using System.Net;  
using System.Net.Security;  
using System.Globalization;  
using System.Security.Policy;  
using System.Security.Cryptography;  
using System.Security.Cryptography.X509Certificates;
```

```
namespace uwcfcs.sample  
{
```

[see more](#)

1 ^ | v • [Reply](#) • [Share](#) ›



Ayoub • 10 days ago

Hi all,

I found that the hash for the input "hello world!" gives different results .

Hex format : 323b1637c7999942fbebfe5d42fe15dbfe93737577663afa0181938d7ad4a2ac

Hex format : 323b1637c7999942fbebfe5d42fe15dbfe93737577663afa181938d7ad4a2ac

like what @StYleZ said there is a 0("zero") missing.

^ | v • [Reply](#) • [Share](#) ›



Savani • 3 months ago

Hi Mkyong,

How we can compare two hashed values for the same salt ? Please share sample code / links etc.

Regards, Savani

^ | v • [Reply](#) • [Share](#) ›



Yuriy Tereschuk • a year ago

@Mkyong, can you update the second reference in References part, because it's not valid.

^ | v • [Reply](#) • [Share](#) ›



diya • a year ago

how do i generate hash value by passing multiple parameters? If i need to send 2 strings and generate a hash value ..is it possible?

^ | v • [Reply](#) • [Share](#) ›



UN-deathx • a year ago



Thank's

^ | v • Reply • Share ›



Iwan • 3 years ago

I think the best way to create the hex format is using the runtime libs.
Use this:

```
BigInteger bigInt = new BigInteger(1, mbytes);  
output = bigInt.toString(16);
```

Got that tip from <http://stackoverflow.com/quest...>

^ | v • Reply • Share ›



Straville → Iwan • 2 years ago

Do not use this 'advice'

Using BigInteger leaves off leading zeros, making this work at some cases and broken in some.

This nonsense just wasted 2 hours of my time...

3 ^ | v • Reply • Share ›



mery • 3 years ago

Method 2 gives another result..

A problem by "01"

Hex format : 8a47c4856ca6de2a016f3a2ab10ef79362ecfc73b038bccfa6eff48afcef4244

Hex format : 8a47c4856ca6de2a16f3a2ab1ef79362ecfc73b038bccfa6eff48afcef4244

^ | v • Reply • Share ›



sheeysong • 4 years ago

Hi Yong,

Thanks for this simple toHexString() code block, can you shed your light to reverse it back to the original ascii String in Java? I tried to write a block of code to convert your MessageDigest byte[] back to String (String HexByteToString(byte[] digestByte), but not quite right somehow.

Thanks,

~Jing

^ | v • Reply • Share ›



Bambat • 4 years ago

```
String digestAlgorithm = "SHA-256";  
String fileName = "/tmp/test";  
String result;  
// Obtain a message digest object.  
MessageDigest md = MessageDigest.getInstance(digestAlgorithm);  
  
// Calculate the digest for the given file.  
DigestInputStream in = new DigestInputStream(  
    new FileInputStream(fileName), md);  
byte[] buffer = new byte[8192];  
while (0 < in.read(buffer)){  
    // get digest
```

```
// get digest
byte[] digest = md.digest();
// convert the byte to hex format
StringBuilder sb = new StringBuilder();
for (byte d : digest) {
    sb.append(String.format("%02x", d));
}
result = sb.toString();
```

^ | v • Reply • Share ›



Sudhakar • 4 years ago

I tried, { 2. Hashing String with SHA-256 } Example.

```
//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));
}
```

When i refresh the page. this code gets looped.

Example:

First Output: 1x2x3x

Second Output: 1x2x3x1x2x3x

Third Output: 1x2x3x1x2x3x1x2x3x

But second one works good.

^ | v • Reply • Share ›



Daniel Serodio • 4 years ago

You forgot to close the FileInputStream in the first example.

^ | v • Reply • Share ›



BillR • 4 years ago

getBytes() is platform dependent. you should specify an encoding so that it works no matter what encoding is used on your platform.

^ | v • Reply • Share ›



venkateswarlu • 4 years ago

what is package com.mkyong.test;

^ | v • Reply • Share ›



Sudhakar → venkateswarlu • 4 years ago

com.mkyong.test;

Its your current package or folder.

You are writing your code in com/mkyong/test folder.

^ | v • Reply • Share ›



Santis • 4 years ago

Hi,

Thank you for your article. Second method of converting the byte to hex is more efficient than first

one (about 30%).

But I have different problem. When I compared Java SHA-256 hashing with Linux program (echo 123456 | sha256sum) I've got different results (for "123456"):

JAVA: 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92

LINUX: e150a1ec81e8e93e1eae2c3a77e66ec6dbd6a3b460f89c1d08aecf422ee401a0

I can't find reason, do you know one?

^ | v • Reply • Share ›



einsty → Santis • 4 years ago

Note that you will need to use echo -n 123456 | sha256sum

The echo command includes a carriage return i believe and that is getting hashed as well... that's the reason your result will be different.

^ | v • Reply • Share ›



Santis → einsty • 4 years ago

That's the point. Thank you.

^ | v • Reply • Share ›



Joe • 5 years ago

Thank you so much, this tutorial is very useful and clear.

^ | v • Reply • Share ›



Bart Oudhoff • 5 years ago

Thanks, this helped me out a lot!

^ | v • Reply • Share ›



Neha → Bart Oudhoff • 3 years ago

But , this is not helped me a lot!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

^ | v • Reply • Share ›



Lucky Luck • 5 years ago

Hi,

Can you please tell me that how can we convert the SHA-256 Hash into simple text, using javascript, c#.net?

Thanks

^ | v • Reply • Share ›



Jersey Jim • 5 years ago

Another byte to hex format method:

```
java.math.BigInteger number = new java.math.BigInteger(1, byteData);  
System.out.println(String.format("%1$032x", number));
```

I saw this idea in the comments at: <http://www.spiration.co.uk/pos...>

^ | v • Reply • Share ›



Jersey Jim → Jersey Jim • 5 years ago

-oops should have been:

```
java.math.BigInteger number = new java.math.BigInteger(1, byteData);
System.out.println(String.format("%1$064x", number));
```

Sorry about that!

^ | v • Reply • Share ›



forber • 5 years ago

Wrong:(2. Hashing String with SHA-256)

System.out.println(Hex format : " + hexString.toString());

Right

System.out.println("Hex format : " + hexString.toString());

^ | v • Reply • Share ›



mkyong → forber • 5 years ago

Article is updated, thanks for point out the typo mistake.

^ | v • Reply • Share ›



Dipika → mkyong • 3 years ago

Hi Sir,

Its really a very good job you are doing

Just a suggestion, there is no searching option available on this site.

Could you please add the same.

^ | v • Reply • Share ›



Rishabh → mkyong • 3 years ago

Hi Sir,

Its really a very good job you are doing

Just a suggestion, there is no searching option available on this site.

Could you please add the same.

^ | v • Reply • Share ›

ALSO ON MKYONG.COM

Logback – Disable logging in Unit Test

2 comments • 2 months ago



fmpdmb — My understanding is that you should only be seeing the logback logging if you logback.xml file contains debug="true",

Windows 10 – Can not create new folder

1 comment • a day ago

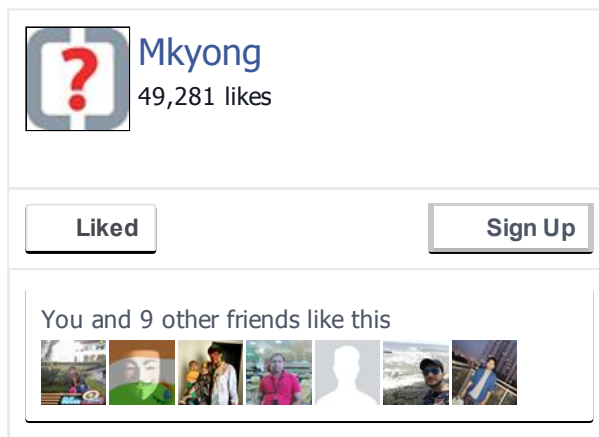
How to register a servlet filter in Spring MVC

3 comments • 9 months ago



dit j. — so and what about url-pattern?

Java – How to overrides equals and hashCode



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 ([//mkyong.com/go/liquidweb/](https://mkyong.com/go/liquidweb/)), and the caches are served by CloudFlare CDN.

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