

How to generate MD5 Hash in Java - String Byte Array digest Example

There are multiple ways to generate MD5 hash in Java program. Not only Java API provides convenient method to generate MD5 hash, you can also use popular open source frameworks, like Spring and Apache commons Codec to generate MD5 digest in Java. MD5 is popular Message Digest Algorithm, which is most commonly used to check data integrity e.g. comparing MD5 check-sum to see, if any file is altered or not. Though, MD5 was not considered a good cryptographic algorithm for security purpose due to several vulnerability found on it, it's still good enough or checking integrity of file. MD5 hashing algorithm generate a 128 bit or 16 byte long hash value. *MD5 hash values*, also known as *MD5 digest* is mostly represented as 32 character Hex String. You can generate MD5 hash from a [byte array](#), or [String](#) directly using Java, Spring and Apache commons codec. Spring and Apache commons codec has identical API e.g. class name `DigestUtils` is same and allows you to directly generate MD5 hash as Hex String, while if you use Java than you need to convert byte array to Hex String, as `java.security.MessageDigest.digest()` method returns MD5 hash as byte array. Earlier we have seen, [How to encode and decode String in base64 encoding](#), and In this Java tutorial we will see, *How to generate MD5 hash or digest using Java*, Spring and Apache commons code library.

How to Generate MD5 hash in Java - example



In this part, we will see some example to generate MD5 hash in Java. Following Java program generates MD5 hash or digest of a String, by converting into byte array. Java Security package provides `MessageDigest`, which can generate MD5 hash. `MessageDigest`'s `digest()` method accept a byte array, and return a byte array of hash value. Since many times we need MD5 hash as Hex String, I have converted that byte array into [Hex String](#). Apart from core Java example, which doesn't use any dependency, there are couple of more open source solution of generating MD5 digest. Two of the most popular open source library, Apache Commons Codec and Spring Framework provides utility method to create MD5 Hash in both byte array and Hex String format. Both Apache commons code and Spring provides, `DigestUtils` class with overloaded method `md5()` and `md5Hex()`, which can accept either [String](#) or byte array and can return either 16 element byte array or 32 characters Hex String. If you are already using either Spring or Apache Commons Codec, than its best to use `DigestUtil`, as it only need a line of code to generate MD5 hash in Java. here is complete code example of generating MD5 digest in Java.

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.commons.codec.digest.DigestUtils;

/**
 * Java program to generate MD5 hash or digest for String. In this example
 * we will see 3 ways to create MD5 hash or digest using standard Java API,
 * Spring framework and open source library, Apache commons codec utilities
 *
 * Generally MD5 has are represented as Hex String so each of this function
 * will return MD5 hash in hex format.
 *
 * @author Javin Paul
 */
public class MD5Hash {

    public static void main(String args[]) {

        String password = "password";

        System.out.println("MD5 hash generated using Java          : " + md
5Java(password));
        System.out.println("MD5 digest generated using           : " + md
5Spring(password));
        System.out.println("MD5 message created by Apache commons codec : "
+ md5ApacheCommonsCodec(password));
    }
}
```

Menu

- About Me
- core java tutorial
- data structure and algorithm
- java interview questions
- coding interview questions
- java certification resources
- java multithreading tutorials
- best practices
- java collection tutorials
- Privacy Policy

Categories

- programming (230)
- coding (88)
- interview questions (42)
- jsp-servlet (33)
- database (28)
- linux (28)
- Eclipse (19)
- Java 8 (19)
- design patterns (18)
- books (14)
- JDBC (13)
- Java xml tutorial (9)

Followers

Join this site
with Google Friend Connect

Members (3547) [More »](#)



Already a member? [Sign in](#)

Recommended Best Practices

- 3 Method and Constructor overloading best practices
- How string in switch case internally works in Java 7?
- Top 10 Programming Best Practices to name variables and methods
- How SSL, HTTPS and Certificates works in Java Application?
- 7 tips while dealing with password in Java?
- Introduction of How Android works for Java Programmers
- How substring() method works in Java?

Follow by Email

Email address...

Blog Archive

- 2015 (119)
- 2014 (106)
- ▼ 2013 (136)
 - December (5)
 - November (7)
 - October (3)
 - September (3)

```

    }

    public static String md5Java(String message){
        String digest = null;
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] hash = md.digest(message.getBytes("UTF-8"));

            //converting byte array to Hexadecimal String
            StringBuilder sb = new StringBuilder(2*hash.length);
            for(byte b : hash){
                sb.append(String.format("%02x", b&0xff));
            }

            digest = sb.toString();

        } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(StringReplace.class.getName()).log(Level.SEVERE, null, ex);
        } catch (NoSuchAlgorithmException ex) {
            Logger.getLogger(StringReplace.class.getName()).log(Level.SEVERE, null, ex);
        }
        return digest;
    }

    /*
     * Spring framework also provides overloaded md5 methods. You can pass
     input
     * as String or byte array and Spring can return hash or digest either
     as byte
     * array or Hex String. Here we are passing String as input and getting
     * MD5 hash as hex String.
     */
    public static String md5Spring(String text){
        return DigestUtils.md5Hex(text);
    }

    /*
     * Apache commons code provides many overloaded methods to generate md5
     hash. It contains
     * md5 method which can accept String, byte[] or InputStream and can re
     turn hash as 16 element byte
     * array or 32 character hex String.
     */
    public static String md5ApacheCommonsCodec(String content){
        return DigestUtils.md5Hex(content);
    }

}

```

Output:

```

MD5 hash generated using Java           : 5f4dcc3b5aa765d61d8327deb882cf99
MD5 digest generated using              : 5f4dcc3b5aa765d61d8327deb882cf99
MD5 message created by Apache commons code : 5f4dcc3b5aa765d61d8327deb882cf99

```

That's all on **How to generate MD5 hash or digest in Java**. As you have seen, there is more than one way to generate MD5 digest, both in byte array as well as Hex String format. Just remember to specify file encoding to `String.getBytes()` method, until you have not using any application wide [character encoding](#). Since `String.getBytes()` uses platform specific encoding which could be different in your windows development machine

► August (13)

► July (12)

► June (9)

► May (14)

► April (18)

▼ March (16)

How to Convert and Print Byte array to Hex String ...

10 Exception handling Best Practices in Java Progr...

How to Reverse Array in Java - Int and String Arra...

How to check if two String are Anagram in Java - P...

How to generate MD5 Hash in Java - String Byte Arr...

10 Famous Laws of Computer Programming and Softwar...

Can You Overload or Override Static methods in Jav...

Top 15 Data Structures and Algorithm Interview Que...

Difference between Struts 1 and Struts 2 framework...

Bitwise and BitShift Operators in Java - AND, OR, ...

Difference between Singleton Pattern vs Static Cla...

How to Increase Console Buffer Size in Eclipse IDE...

ReentrantLock Example in Java, Difference between ...

5 books to learn Spring framework and Spring MVC f...

How to create Immutable Class and Object in Java -...

Steps to Create JUnit Test in Eclipse and Netbeans...

► February (18)

► January (18)

► 2012 (217)

► 2011 (144)

► 2010 (33)

Copyright by Javin Paul 2010-2015. Powered by [Blogger](#).

and production Linux Server. Also, consider using Spring or Apache Commons Code to generate MD5 Hash value, if you are already using these libraries. It's always best to reuse library code than writing your own MD5 hash function, to avoid testing overhead.

Other **Java Programming Tutorials** from Javarevisited Blog

[How to sort HashMap and Hashtable based on values in Java](#)

[3 ways to print array in Java](#)

[4 ways to traverse Map in Java](#)

[Difference between transient and volatile variable in Java](#)

[Java Best practices for method overloading in Java](#)

You might like:-

- [Why String is Immutable or Final in Java](#)
- [How to find PID of process listening on a port in Linux? netstat and lsof command examples](#)
- [10 Singleton Pattern Interview Questions in Java - Answered](#)
- [Top 20 String Algorithm Questions from Coding Interviews](#)

Recommended by

Posted by Javin Paul 



+7 Recommend this on Google

Labels: [Array](#), [core java](#), [data structure and algorithm](#), [Java Programming Tutorials](#), [programming](#)

Location: [United States](#)

2 comments :

[SARAL SAXENA](#) said...

Hi Javin gr8 article fe things I want to add in this..

The MessageDigest class can provide you with an instance of the MD5 digest.

Always when working with strings and the crypto classes be sure to always specify the encoding you want the byte representation in. If you just use string.getBytes() it will use the platform default. (Not all platforms use the same defaults)

```
import java.security.*;
```

```
..
```

```
byte[] bytesOfMessage = yourString.getBytes("UTF-8");
```

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

```
byte[] thedigest = md.digest(bytesOfMessage);
```

If you have a lot of data take a look at the .update(byte[]) method which can be called repeatedly. Then call .digest() to obtain the resulting hash.

[March 20, 2013 at 10:54 PM](#)

[Javin @ What is ClassLoader in Java](#) said...

@Saral, Indeed, specifying character encoding is very important, but, instead of hard coding them at code e.g. getBytes("UTF-8"), which works well on test programs, is not very convenient on real project. Consider using an application wide character encoding, which you can provide by using System property -Dfile.encoding="UTF-8". This allows you to change and manage character encoding easily. Regarding update(), yes that's preferred way if input is large.

[March 20, 2013 at 11:55 PM](#)

Post a Comment

Enter your comment...

Comment as: **VISHWARUP** ▼

[Sign out](#)

[Publish](#)

[Preview](#)

☐ Notify me

12/7/2015

How to generate MD5 Hash in Java - String Byte Array digest Example

Subscribe to: [Post Comments \(Atom \)](#)