

[Home](#) [News](#) [Resources](#) [Tutorials](#) [Whitepapers](#) [Job Board](#) [Academy](#) [About](#)

# Java Code Geeks

JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

[Android](#) [Core Java](#) [Desktop Java](#) [Enterprise Java](#) [Java Basics](#)You are here: [Home](#) [crypto](#) Encrypt/Decrypt string with DES

### About Byron Kiourtoglou

Byron is a master software engineer working in the IT and Telecom domains. He is always fascinated by SOA, middleware services and mobile development. Byron is co-founder and Executive Editor at Java Code Geeks.



## Encrypt/Decrypt string with DES

by Byron Kiourtoglou on November 11th, 2012 | Filed in: [crypto](#) Tags: [core java](#), [crypto](#)

In this example we shall show you how to encrypt and decrypt a String with DES. DES in computing refers to the [Data Encryption Standard](#) and is supported by Java. To encrypt and decrypt a String with DES one should perform the following steps:

- Generate a [SecretKey](#) using DES algorithm, with the [KeyGenerator](#) `generateKey()` API method.
- Initialize two [Ciphers](#), one in encryption mode and the other one in decryption mode. Use them to encrypt the String message and then decrypt the encrypted String.
- The encryption is performed in the `String encrypt(String str)` method. It encodes the string into a sequence of bytes using the named charset, storing the result into a new byte array. Then it calls `doFinal(byte[] input)` API method of Cipher to make the encryption. It uses the `com.sun.mail.util.BASE64EncoderStream` to encode the encrypted byte array and returns the String created from the byte array.
- The decryption is performed in the `String decrypt(String str)` method. It uses the `com.sun.mail.util.BASE64DecoderStream` to decode the String to byte array. Then it calls `doFinal(byte[] input)` API method of Cipher to make the decryption. It creates a new string based on the specified charset from the decrypted byte array,

as described in the code snippet below.

```
001 package com.javacodegeeks.snippets.core;
002
003 import java.security.InvalidKeyException;
004 import java.security.NoSuchAlgorithmException;
005
006 import javax.crypto.Cipher;
007 import javax.crypto.KeyGenerator;
008 import javax.crypto.NoSuchPaddingException;
009 import javax.crypto.SecretKey;
010
011 import com.sun.mail.util.BASE64DecoderStream;
012 import com.sun.mail.util.BASE64EncoderStream;
013
014 public class EncryptDecryptStringWithDES {
015
016     private static Cipher ecipher;
017     private static Cipher dcipher;
018
019     private static SecretKey key;
020
021     public static void main(String[] args) {
022
023         try {
024
025             // generate secret key using DES algorithm
026             key = KeyGenerator.getInstance("DES").generateKey();
027
028             ecipher = Cipher.getInstance("DES");
029             dcipher = Cipher.getInstance("DES");
030
031             // initialize the ciphers with the given key
032
033             ecipher.init(Cipher.ENCRYPT_MODE, key);
034
035             dcipher.init(Cipher.DECRYPT_MODE, key);
036
037             String encrypted = encrypt("This is a classified message!");
038
039             String decrypted = decrypt(encrypted);
040
041             System.out.println("Decrypted: " + decrypted);
042
043         }
044         catch (NoSuchAlgorithmException e) {
045             System.out.println("No Such Algorithm:" + e.getMessage());
046             return;
047         }
048         catch (NoSuchPaddingException e) {
049             System.out.println("No Such Padding:" + e.getMessage());
050             return;
051         }
052         catch (InvalidKeyException e) {
053             System.out.println("Invalid Key:" + e.getMessage());
```

### Newsletter

**45987** insiders are already updates and complimentary **Join them now** to gain to the latest news in the Java insights about Android, Scala related technologies.

### Email address:

### Join Us



With unique authentication the the around the the k enco

If you have a blog with unique content then you should check our partners program. You can join for Java Code Geeks and I

### Carrer Opportunities

Director of Engineering (October 25th, 2014)

Senior Application Developer October 24th, 2014

Programmer III-IV ( FULL TIME ) 2014

Director of Engineering ( 24th, 2014

Machine Learning Principles TIME ) October 24th, 201

### Tags

[android](#) [Android Core](#) [arraylist](#) [awt](#) [class](#) [core java](#) [data type](#) [dom](#) [Eclipse](#) [enterprise](#) [gwt](#) [hashtable](#) [hibernate](#) [io](#) [lang](#) [linkedlist](#) [logging](#) [regex](#) [security](#) [spring](#) [swing](#) [threads](#) [treemap](#)

```
054         return;
055     }
056 }
057 }
058
059 public static String encrypt(String str) {
060
061     try {
062         // encode the string into a sequence of bytes using the named charset
063         // storing the result into a new byte array.
064
065         byte[] utf8 = str.getBytes("UTF8");
066
067         byte[] enc = ecipher.doFinal(utf8);
068
069         // encode to base64
070
071         enc = BASE64EncoderStream.encode(enc);
072
073         return new String(enc);
074     }
075 }
076
077 catch (Exception e) {
078     e.printStackTrace();
079 }
080
081 return null;
082 }
083
084 public static String decrypt(String str) {
085
086     try {
087         // decode with base64 to get bytes
088
089         byte[] dec = BASE64DecoderStream.decode(str.getBytes());
090
091         byte[] utf8 = dcipher.doFinal(dec);
092
093         // create new string based on the specified charset
094
095         return new String(utf8, "UTF8");
096     }
097 }
098
099 catch (Exception e) {
100     e.printStackTrace();
101 }
102
103 return null;
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
```

**Output:**

Decrypted: This is a classified message!

This was an example of how to encrypt and decrypt a String with DES in Java.

## Do you want to know how to develop your skillset to become a **Java Rockstar**?

### Subscribe to our newsletter to start **Rocking right now!**

To get you started we give you two of our best selling eBooks for **FREE!**

#### JPA Mini Book

Learn how to leverage the power of JPA in order to create robust and flexible Java applications. With this Mini Book, you will get introduced to JPA and smoothly transition to more advanced concepts.

#### JVM Troubleshooting Guide

The Java virtual machine is really the foundation of any Java EE platform. Learn how to master it with this advanced guide!

Email address: