import java.security.Key;

import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;

import java.util.Base64;

import java.util.concurrent.TimeUnit;

import java.util.Random;

//class

public class AES\_ECB {

private static final String ALGO = "AES/ECB/PKCS5PADDING";

private byte[] keyValue;

//constructor

public AES\_ECB(String key){

keyValue = key.getBytes();

}

// Key Generation

private Key generateKey() throws Exception{

Key k = new SecretKeySpec(keyValue,"AES");

return k;

}

// Encryption

public String encrypt(String data) throws Exception{

Key k = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.ENCRYPT\_MODE,k);

byte[] encVal = c.doFinal(data.getBytes());

return Base64.getEncoder().encodeToString(encVal);

}

// main method

public static void main(String[] args){

try{

long startTime = System.currentTimeMillis();

for(int i=0;i<10000;i++){

AES\_ECB aes = new AES\_ECB("This is a key123");

// message

byte[] r = new byte[32];

Random rn = new Random();

rn.nextBytes(r);

String s = new String(r);

String encryptedString = aes.encrypt(s);

//System.out.println(encryptedString);

}

long endTime = System.currentTimeMillis();

long timeElapsed = endTime - startTime;

System.out.println("Execution time in milliseconds: " + timeElapsed);

}catch (Exception e){

System.out.println("Error while encrypting: " + e.toString());

}

}

}