Java中常用的加密算法MD5,SHA,RSA PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/461/2021_2022_Java_E4_B8_AD_E5_B8_B8_c104_461626.htm 1. MD5加密，常用于加密用户名密码，当用户验证时。protected byte[] encrypt(byte[] obj) ...{ try ...{ MessageDigest md5 = MessageDigest.getInstance("MD5"). md5.0update(obj). return md5.digest(). } catch (NoSuchAlgorithmException e) ...{ e.printStackTrace(). } } 2. SHA加密，与MD5相似的用法，只是两者的算法不同。protected byte[] encrypt(byte[] obj) ...{ try ...{ MessageDigest sha = MessageDigest.getInstance("SHA"). sha.0update(obj). return sha.digest(). } catch (NoSuchAlgorithmException e) ...{ e.printStackTrace(). } } 3. RSA加密，RAS加密允许解密。常用于文本内容的加密。 import java.security.KeyPair. import java.security.KeyPairGenerator. import java.security.interfaces.RSAPrivateKey. import java.security.interfaces.RSAPublicKey. import javax.crypto.Cipher. /** *//** * RSAEncrypt * * @author maqujun * @see */ public class RSAEncrypt ...{ /** *//** * Main method for RSAEncrypt. * @param args */ public static void main(String[] args) ...{ try ...{ RSAEncrypt encrypt = new RSAEncrypt(). String encryptText = "encryptText". KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA"). keyPairGen.initialize(1024). KeyPair keyPair = keyPairGen.generateKeyPair(). // Generate keys RSAPrivateKey privateKey = (RSAPrivateKey) keyPair.getPrivate(). RSAPublicKey publicKey = (RSAPublicKey) keyPair.getPublic().

byte[] e = encrypt.encrypt(publicKey, encryptText.getBytes()). byte[] de = encrypt.decrypt(privateKey,e). System.out.println(encrypt.bytesToString(e)). System.out.println(encrypt.bytesToString(de)). } catch (Exception e) ...{ e.printStackTrace(). } } /** *//** * Change byte array to String. * @return byte[] */ protected String bytesToString(byte[] encrytpByte) ...{ String result = "". for (Byte bytes : encrytpByte) ...{ result = (char) bytes.intValue(). } return result. } /** *//** * Encrypt String. * @return byte[] */ protected byte[] encrypt(RSAPublicKey publicKey, byte[] obj) ...{ if (publicKey != null) ...{ try ...{ Cipher cipher = Cipher.getInstance("RSA"). cipher.init(Cipher.ENCRYPT_MODE, publicKey). return cipher.doFinal(obj). } catch (Exception e) ...{ e.printStackTrace(). } } return null. } /** *//** * Basic decrypt method * @return byte[] */ protected byte[] decrypt(RSAPrivateKey privateKey, byte[] obj) ...{ if (privateKey != null) ...{ try ...{ Cipher cipher = Cipher.getInstance("RSA"). cipher.init(Cipher.DECRYPT_MODE, privateKey). return cipher.doFinal(obj). } catch (Exception e) ...{ e.printStackTrace(). } } return null. } }