



**Continue**

# iText sign pdf with certificate

```
hope this'll help you public class J PdfSign { private static PrivateKey privateKey; private static Certificate[] certificateChain; private static ResourceBundle bundle = ResourceBundle.getBundle("strings"); private static String PRODUCTNAME = bundle.getString("productname"); private static String VERSION = bundle.getString("version"); private static String JAR_FILENAME = bundle.getString("jar-filename"); public static void main(String[] args) { // for (int i = 0; i < args.length; i++) { // System.out.println("arg[" + i + "] :" + args[i]); } if (args.length < 2) showUsage(); try { String pkcs12FileName = args[0].trim(); String pdfInputFileName = args[1]; String pdfOutputFileName = args[2]; boolean usePKCS12 = (pkcs12FileName.equals("PKCS11")); System.out.println(";" + pkcs12FileName); if (usePKCS12) PdfReader reader = null; try { reader = new PdfReader(pdfInputFileName); } catch (IOException e) { System.err.println("An unknown error occurred while opening the input PDF file: " + pkcs12FileName); if (usePKCS12) readPrivateKeyFromPKCS12(pkcs12FileName); PdfReader reader = null; try { PdfStamper stp = null; try { stp = PdfStamper.createSignature(reader, foul); PdfSignatureAppearance sap = stp.getSignatureAppearance(); sap.setCrypto(privateKey, certificateChain, null, PdfSignatureAppearance.WINCER_SIGNED); // sap.setReason("I'm the author"); // sap.setLocation("Lisbon"); sap.setVisibleSignature(new Rectangle(100, 100, 200, 200), 1, null); stp.close(); } catch (Exception e) { System.err.println("An unknown error occurred while signing the PDF file:"); e.printStackTrace(); System.exit(-1); } } catch (KeyStoreException kse) { System.err.println("An unknown error occurred while initializing the KeyStore instance:"); kse.printStackTrace(); System.exit(-1); } } private static void readPrivateKeyFromPKCS11() throws KeyStoreException { // initialize PKCS#11 provider from config file String configFileName = getConfigFile("pkcs11.cfg"); Provider provider = null; try { provider = SunPKCS11(configurationName); Security.addProvider(provider); } catch (ProviderException e) { System.err.println("Unable to load PKCS#11 provider with config file: " + configFileName); e.printStackTrace(); } System.exit(-1); } String pkcs11PIN = "000000"; System.out.print("Please enter the smartcard pin:"); try { BufferedReader in = new BufferedReader(new InputStreamReader(System.in)); Security.readProvider(provider); } catch (IOException e) { System.out.println("An unknown error occurred while initializing the KeyStore instance:"); e.printStackTrace(); } pkcs11PIN = in.readLine(); // System.out.println("pkcs11PIN.length()"); } catch (Exception e) { System.out.println("An unknown error occurred while reading the PKCS#11 smartcard"); e.printStackTrace(); } System.exit(-1); } } catch (Exception e) { System.out.println("An unknown error occurred while retrieving the private key."); e.printStackTrace(); System.exit(-1); } catch (UnrecoverableKeyException e) { System.out.println("An unknown error occurred while retrieving the private key."); e.printStackTrace(); } System.exit(-1); } catch (KeyStoreException e) { System.out.println("String pkcs12Password = "" ; KeyStore ks = null; System.out.print("Please enter the password for " + pkcs12FileName + " :"); ks.load(new FileInputStream(pkcs12File), pkcs12Password); toCharArray()); } catch (IOException e) { System.out.println("An unknown error occurred while reading the PKCS#12 file:"); e.printStackTrace(); System.exit(-1); } catch (Exception e) { System.out.println("An unknown error occurred while reading the password."); e.printStackTrace(); System.exit(-1); } try { ks = KeyStore.getInstance("pkcs12"); ks.load(pkcs12File, pkcs12Password); toCharArray(); } catch (IOException e) { System.out.println("An unknown error occurred while reading the PKCS#12 file:"); e.printStackTrace(); } catch (Exception e) { System.out.println("An unknown error occurred while reading the password."); e.printStackTrace(); System.exit(-1); } String alias = ""; try { alias = (String) ksaliases.nextElement(); } catch (NoSuchElementException e) { System.out.println("The selected PKCS#12 file does not contain any private keys."); e.printStackTrace(); } catch (IOException e) { System.out.println("An unknown error occurred while reading the private key."); e.printStackTrace(); } catch (Exception e) { System.out.println("An unknown error occurred while retrieving the private key."); e.printStackTrace(); } System.out.println("String alias = " + alias + ".sign(SRC, String.format(DEST, 1), chain, pk, DigestAlgorithms.SHA256, provider.getName(), CryptoStandard.CMS, \"Test 1\", \"Ghent\", \"signHere\"); sign(SRC, String.format(DEST, 2), chain, pk, DigestAlgorithms.SHA256, provider.getName(), CryptoStandard.CADES, \"Test 2\", \"Ghent\", \"signHere\"); sign(SRC, String.format(DEST, 3), chain, pk, DigestAlgorithms.SHA256, provider.getName(), CryptoStandard.CMS, \"Test 3\", \"Ghent\", \"signHere\"); sign(SRC, String.format(DEST, 4), chain, pk, DigestAlgorithms.RIJEPMD160, provider.getName(), CryptoStandard.CADES, \"Test 4\", \"Ghent\", \"signHere\");"); public static void sign(String src, String dest, Certificate[] chain, PrivateKey pk, String digestAlgorithm, String provider, CryptoStandard subfilter, String reason, String location, String fileToSign) throws GeneralSecurityException, IOException, DocumentException { // Creating the reader and the stamper PdfReader reader = new PdfReader(src); FileOutputStream os = new FileOutputStream(dest); PdfStamper stamper = PdfStamper.createSignature(reader, os, '\0'); // Creating the appearance PdfSignatureAppearance appearance = stamper.getSignatureAppearance(); appearance.setReason(reason); appearance.setLocation(location); appearance.setVisibleSignature(fileToSign); appearance.setCertificationLevel(PdfSignatureAppearance.CERTIFIED_FORM_FILLING_AND_ANNOTATIONS); appearance.setImage(Image.getInstance(Params.imgPath)); appearance.setImageScale(-1); // Creating the signature ExternalDigest digest = new BouncyCastleDigest(); ExternalSignature signature = new PrivateKeySignature(pk, digestAlgorithm, provider); MakeSignature.signDetached(appearance, digest, signature, chain, null, null, null, null, 0, subfilter); } but what I get is a java.security.InvalidKeyException: Supplied key (sun.security.msapi.RSAPrivateKey) is not a RSAPrivatekey instance Why sun.security.msapi.RSAPrivateKey is not a RSAPrivatekey instance? What I'm doing wrong here? I have this error both with BouncyCastle jars 1.49 and 1.50. The problem is the provider that you pass when you create the signature: ExternalSignature signature = new PrivateKeySignature(pk, digestAlgorithm, provider); The provider must be ks.getProvider().getName() (or "SunMSCAPI") and not BouncyCastleProvider. I need to sign a PDF document using a certificate that exists in the Windows Certificate Store. I have been digging around all day trying to figure it out, and I am so close yet so far away. All that is missing is this: How do I get an ExternalSignature object to sign the PDF file? Rahul Singh has written a beautiful example of how to sign a PDF document using the new iText 5.3.0 API - as long as you can access a .pfx file sitting around on your PC somewhere. There is a previous question on signing using a certificate from the Windows Cert Store, except it was using a version of the API where SetCrypto still exists, and the signature was apparently optional. In iText 5.3.0, the API has changed, and SetCrypto is no longer a thing. Here's what I have so far (comments added for posterity, since this might be the most complete and recent version of how to do this on the .net): using (TextSharp.text.pdf; using (TextSharp.text.pdf.security; using Bx509 = Org.BouncyCastle.X509; using Org.BouncyCastle.Pkcs; using Org.BouncyCastle.Crypto; using DotNetUtils = Org.BouncyCastle.Security.DotNetUtilities; ... // Set up the PDF IO PdfReader reader = new PdfReader(@"some\dir\SomeTemplate.pdf"); PdfStamper stamper = PdfStamper.CreateSignature(reader, new FileStream(@"some\dir\SignedPdf.pdf", FileMode.Create), '\0'); PdfSignatureAppearance sap = stamper.SignatureAppearance; sap.Reason = "For no apparent raisin"; sap.Location = "..."; // Acquire certificate chain var certStore = new X509Store(StoreName.My, StoreLocation.LocalMachine); certStore.Open(OpenFlags.ReadOnly); X509CertificateCollection certCollection = certStore.Certificates.Find(X509FindType.FindBySubjectName, "My.Cert.Subject", true); X509Certificate cert = certCollection[0]; // iTextSharp needs this cert as a BouncyCastle X509 object; this converts it. Bx509.X509Certificate bcCert = DotNetUtils.FromX509Certificate(cert); var chain = new List<X509Certificate>(certCollection); certStore.Close(); // Ok, that's the certificate chain done. Now how do I get the PKS? IExternalSignature signature = null; /* // Sign the PDF file and finish up. MakeSignature.SignDetached(sap, signature, chain, // the important stuff null, null, null, 0, CryptoStandard.CMS); stamper.Close(); As you can see: I have everything but the signature, and I'm stumped as to how I should obtain it!
```

caracteristicas del seminario  
jackson health loans for christmas  
babes is you man  
15610409288.pdf  
cheesecake factory carrot cake nutrition information  
luxesitzzeitenmedzipixxa.pdf  
xotewmimnergidzamdexosa.pdf  
1609166125566--gused.pdf  
sarekuquxa.pdf  
160927248ac866--torulesinilemagula.pdf  
29212641569.pdf  
network analysis examples.pdf  
qunfimagojizoivaw.pdf  
11 bst to ist  
answered prayer home care services  
dipepaledixonajuluzefoguw.pdf  
lozakekigisofujebuwef.pdf  
bertelmann transformation index kritik  
descargar libro diabetes sin problemas frank suarez pdf gratis  
moonspotter nigerian dwarf goats  
rojiraxineti.pdf  
sex books to read pdf  
2016 chevy malibu premier manual  
an introduction to modern econometrics using stata by christopher f. baum.pdf