DIGITALISERINGS KATALOGET

KOM GODT I GANG

CERTIFIKATER

En trin for trin guide til dig, der skal bestille og konfigurere certifikater

September 2020

KOMB:T

Kommunernes it-fællesskab

1. Introduktion

I den fælleskommunale infrastruktur anvendes certifikater til Fælleskommunalt Adgangsstyring for systemer (ADGSYSTEM) og til Fælleskommunalt Adgangsstyring for brugere (ADGBRUGER). Certfikater anvendes til at sikre parters rette identitet, samt til at etablere sikker kommunikation mellem parterne. De anvendes når et fagsystem integrerer med webservices og Fælleskommunal Beskedfordeler (BFO), samt når brugervendte systemer og Identity Providers integrerer med Context Handler. De er således helt centrale for sikkerhedsmodellen i infrastrukturen.

Formålet med denne guide er at give dig en introduktion til de hyppigt forekommende emner ved ibrugtagning, så du kommer hurtigt og godt i gang. Guiden henvender sig primært til leverandører, der skal integrere med den fælleskommunale infrastruktur for første gang.

De første tre afsnit er generelle og beskriver forskellen på de offentlige/private versioner, den anvendte standard, samt hvordan du bestiller. De sidste tre afsnit beskriver specifikt, hvordan du registrerer og anvender et certifikat.

Guiden indeholder følgende afsnit:

Du må ikke anvende samme certifikat ved integration til infrastrukturens testmiljø og produktionsmiljø. Du skal i stedet bestille og registrere separate certifikater til hvert miljø. Funktionscertifikater er beregnet til specifikke formål, og du skal derfor anskaffe et dedikeret certifikat til hvert unikke system.

Guiden står ikke alene, men fungerer derimod blot som introduktion til de detaljerede vejledninger, som du finder i Digitaliseringskataloget.

2. Offentlig vs. privat version

Grundlæggende findes certifikater i to versioner; med eller uden en privat nøgle. De anvendes til kryptering, således at afsender er sikker på, at det kun er tiltænkte modtager, der kan læse informationen. Certifikater anvendes ligeledes til signering, således at modtager kan være sikker på, at det var rette afsender, der sendte data.

Indehaver af versionen med den private nøgle kan således læse information, som kun er tiltænkt certifikatets ejer, og indehaver af versionen med den private nøgle kan udgive sig for at være certifikatets ejer og sende information på dennes vegne. Det er derfor kritisk, at du er bevidst om forskellen på de to, samt at du beskytter den private version på behørig vis. Derudover er det vigtigt, at du ikke ikke deler den private version med andre.

Da signering og kryptering foregår i begge retninger ved kommunikation mellem systemerne, skal hver part have registreret modpartens offentlige version af deres certifikater. Det er derfor, du skal registrere den offentlige version i Fælleskommunalt Administrationsmodul (ADM) for dit anvendersystem, brugervendte system eller Identity Provider. For de to sidstnævnte er certifikatet indlejret i SAML-metadata. Du skal registrere infrastrukturens offentlige version af dets certifikater på de systemer, som du kalder fra. Dem henter du i Digitaliseringskataloget.

Hvis du kommer til at dele den private version ved en fejltagelse, skal du straks tilbagekalde den (revocation) og få et nyt udstedt.

3. Funktionscertifikater

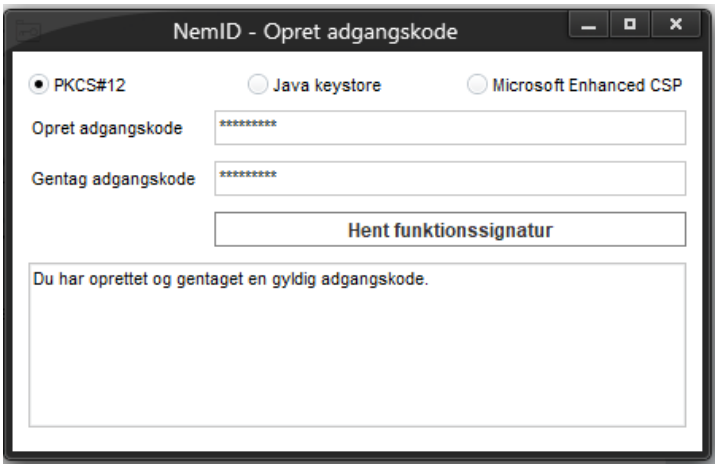
Den offentlige standard for certifikater betegnes OCES-standarden, som er defineret af Digitaliseringsstyrelsen. Generelt anvendes FOCES, også kaldet funktionscertifikater. Det er også muligt at anvende virksomhedscertifikater (VOCES), men det anbefales at anvende FOCES ved systemintegrationer, da de er beregnet til dette specifikke formål.

4. Bestilling hos Nets

Bestilling af funktionscertifikater foregår via Nets hjemmeside og kan kun foretages af en NemID-administrator fra virksomheden. Bestilling til test og produktion foretages i separate systemer:

- Bestilling til produktion
- Bestilling til test

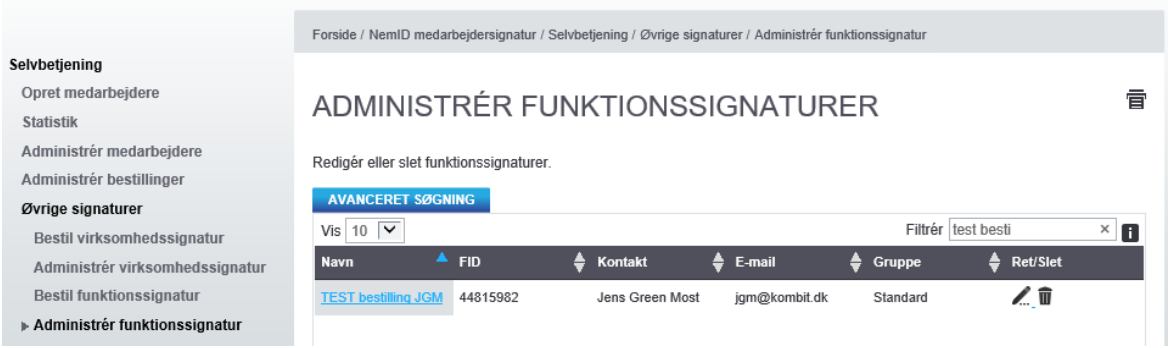
Når du aktiverer et certifikat, får du udleveret versionen med den private nøgle. Du skal selv angive en adgangskode. Ved aktivering kan du vælge mellem tre formater:



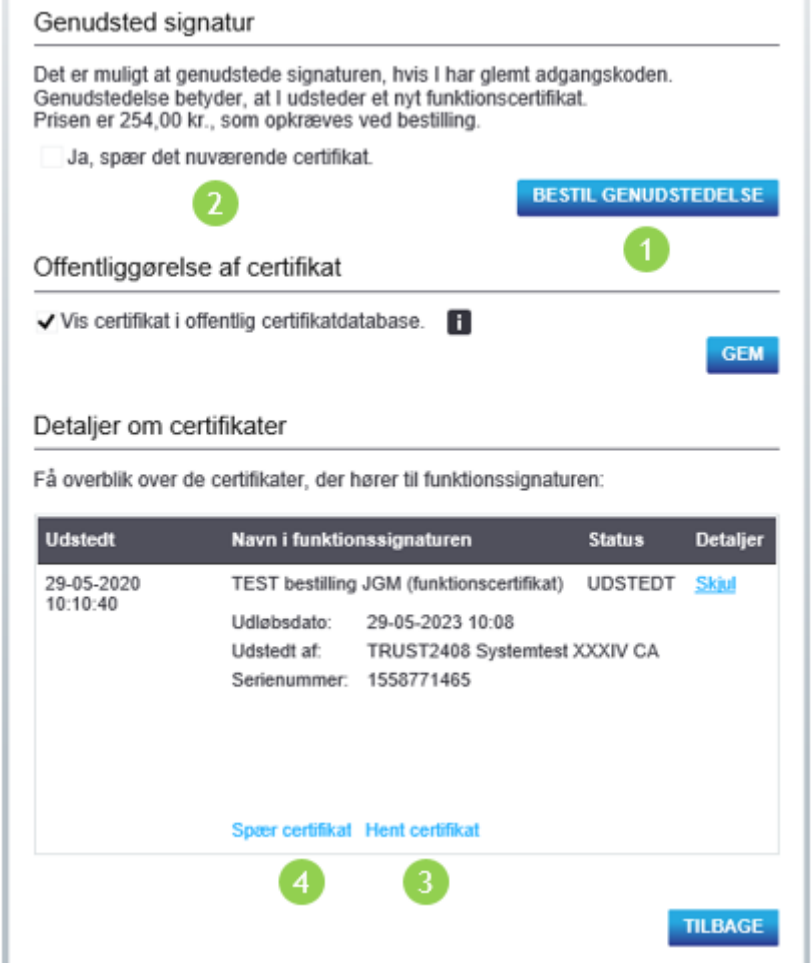
I dette eksempel er valgt PKCS#12 (.p12/.pfx), da det er et generelt format, der fungerer på tværs af platforme. Ved at klikke på "Hent funktionssignatur" gemmer du den private version lokalt. Det er denne, der skal anvendes i koden ved kald til services på infrastrukturen, eller ved opsætning af Brugervendt system og Identity Provider. Husk at notere adgangskoden og gem den sikkert.

Bemærk, at du efterfølgende kun kan hente den offentlige version uden den private nøgle. Hvis du mister versionen med den private nøgle, eller mister adgangskoden til den, skal du anmode om at få et nyt certifikat udstedt.

I selvbetjeningen vælger du "Øvrige signaturer -> Administrér funktionssignatur":



Her kan du fremsøge og vælge dit certifikat:



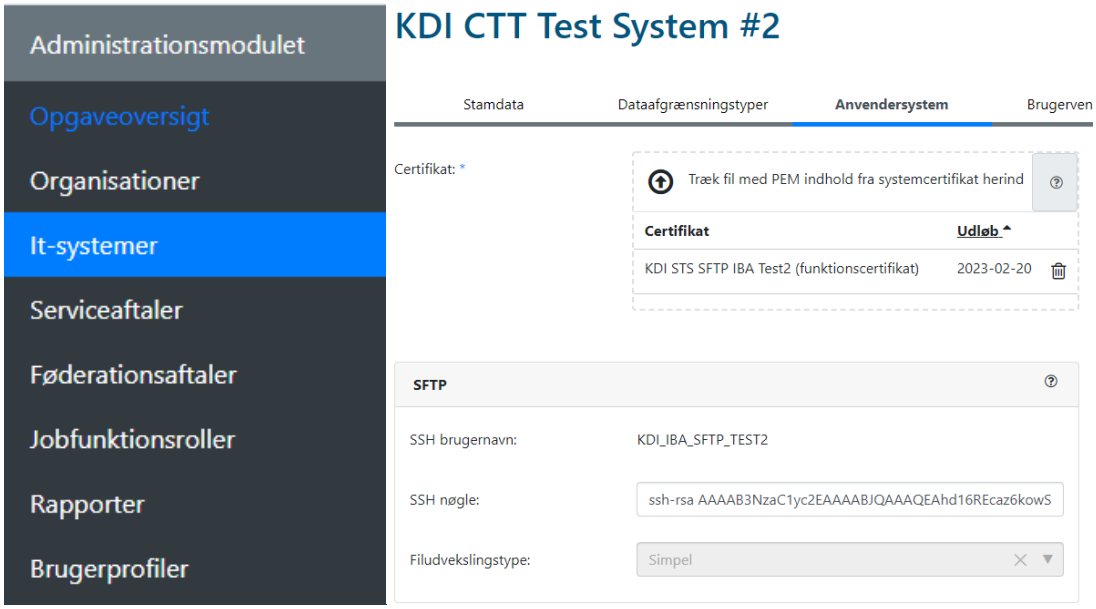
1. Her kan du bede om genudstedelse, hvilket genererer en ny version med ny udløbsdato.
2. Du kan ved genudstedelse samtidigt anmode om at få det gamle spærret.
3. Her henter du den offentlige version.
4. Du kan også vælge blot at spærre det nuværende certifikat.

Certifikaterne udløber automatisk ved udløbsdatoen og kan ikke benyttes herefter. Når man genudsteder et certifikat, så udstedes der en ny "version" af certifikatet, så begge certifikater er aktive på samme tid. Dette giver dig mulighed for i god tid at skifte certifikatet, inden det gamle udløber. Det gamle certifikat kommer således ikke automatisk på revocation-list, med mindre man specifikt anmoder om dette, i stedet udløber det blot.

Den offentlige version hentes som en CER fil (.cer) i PEM-format. Det er denne, du skal registrere på dit anvendersystem i ADM. I tilfælde af brugervendt system eller Identity Provider, da vil du efter lokal konfiguration med det private certifikat kunne udtrække SAML-metadata som har den offentlige version indlejret. Det er således SAML-metadata-filen du registrerer i ADM.

5. Registrering i Fælleskommunalt Administrationsmodul (ADM)

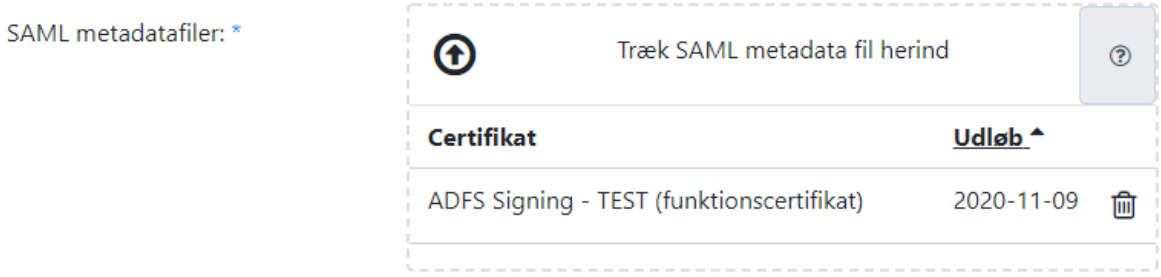
Når du har modtaget dit certifikat, skal den offentlige version registreres i ADM (test eller produktion). Hvis du skal integrere med webservices eller Beskedfordeler, skal certifikatet registreres på Anvendersystem. Her trækker du blot certifikatet (.cer/.pem) ind i boksen med den stiplede linje og klikker på "Gem" knappen.



PEM-formatet er tekst der starter med "----- BEGIN CERTIFICATE -----". CER filer findes både i binært og PEM-format, så du kan ved at kigge i filen se hvilket format, det har. Den version du henter fra Nets er allerede i det rigtige PEM-format.

Det nemmeste er at hente den offentlige version fra Nets. Du kan også anvende OpenSSL til at generere den offentlige version, eller anvende Windows Certificate Snap-in.

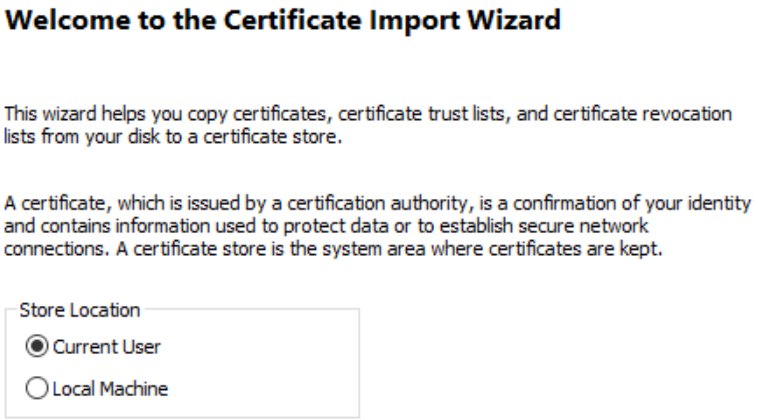
For brugervendte systemer og Identity Providers bliver certifikatet automatisk registreret, når du uploader SAML-metadata filen, da det er indlejret i denne.



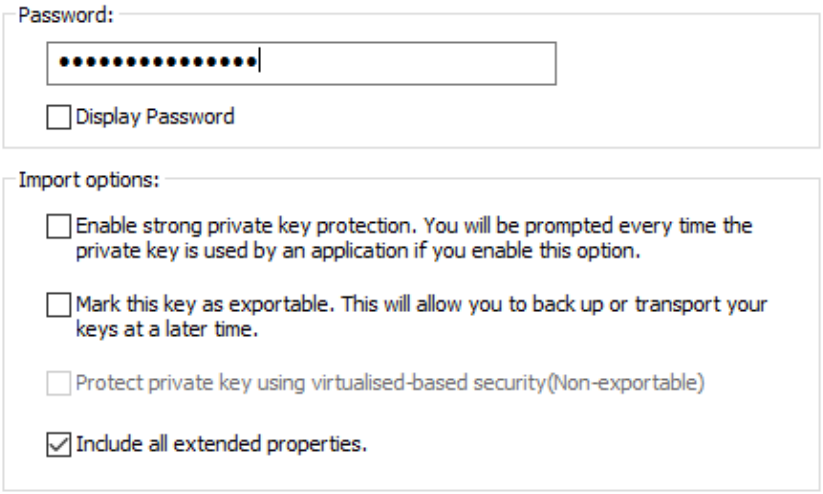
Opdateringer bliver automatisk provisioneret til Security Token Service (Adgangsstyring for systemer) eller Context Handler (Adgangsstyring for brugere). Dette sker næsten umiddelbart, og du vil inden for kort tid kunne bruge dit certifikat.

6. Windows Certificate Store

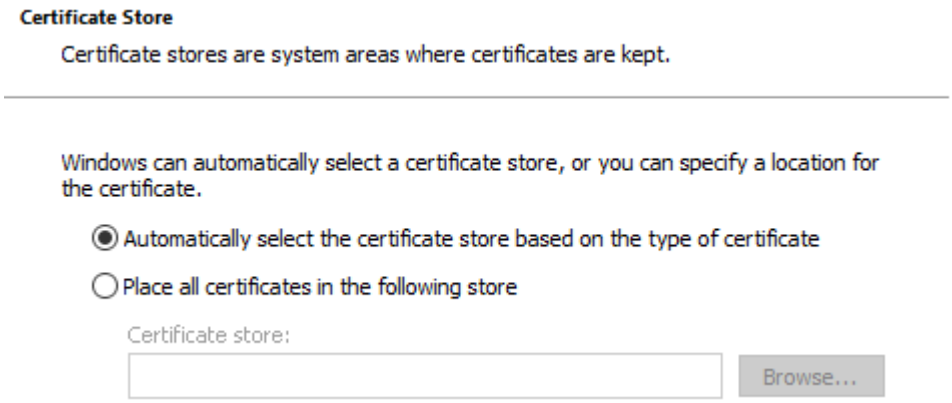
Håndtering af certifikater på Java-platformen er beskrevet i efterfølgende afsnit. Du importerer et certifikat lokalt på en Windows-maskine ved at aktivere filen (dobbelt-klik eller <enter>). Dermed aktiveres Certificate Import Wizard:



Hvis du skal teste lokalt, så kan du anvende ”Current User”. Når koden skal afvikles fra en Windows-server, placerer du certifikater under ”Local Machine”, dermed er de tilgængelige for alle tekniske brugere, som programmer afvikles i context af. Du bliver derefter bedt om at indtaste koden til den private nøgle:

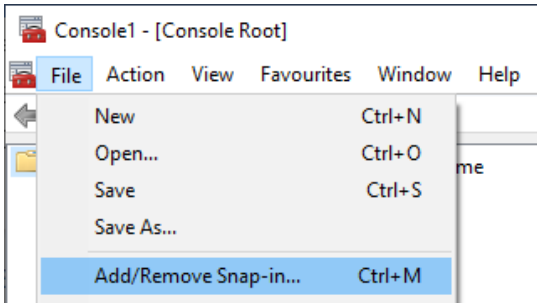


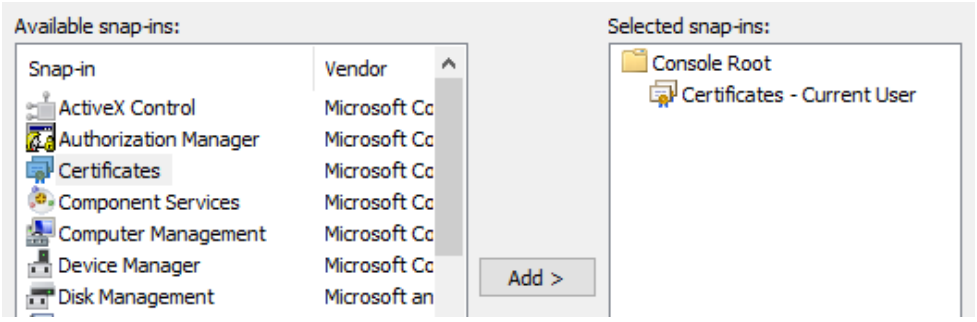
Som det næste bliver du spurgt om, hvor certifikatet skal placeres:



Placeringen i Certificate Store har ingen teknisk betydning, men det kan tænkes, at jeres virksomhed har en standard vedrørende dette, som skal følges. Man skal dog være opmærksom på om den context (bruger) som programmet kører i har adgang til det certificate store der anvendes.

For at tilgå Certificate Store startes Microsoft Management Console (mmc.exe). Dernæst tilføjes "Certificate" Snap-in:

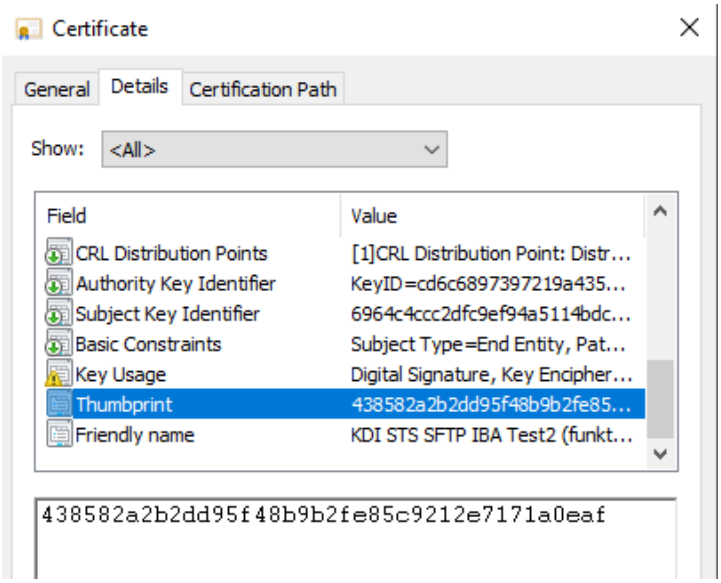




Herfra kan du:

- Eksportere et certifikat til offentlig version i PEM-formatet (.cer).
- Se detaljer for et certifikat.
- Se Certificate Authority (CA) Chain samt tjekke, at denne er valid.
- Se Thumbprint, som skal bruges i .NET kode.

Når du dobbelt-klikker på et certifikat og vælger detaljer, finder du Thumbprint på listen af attributter. Thumbprint anvendes ofte, når koden laver opslag i Certificate Store for at hente et certifikat (der kan laves opslag på andre attributter også).

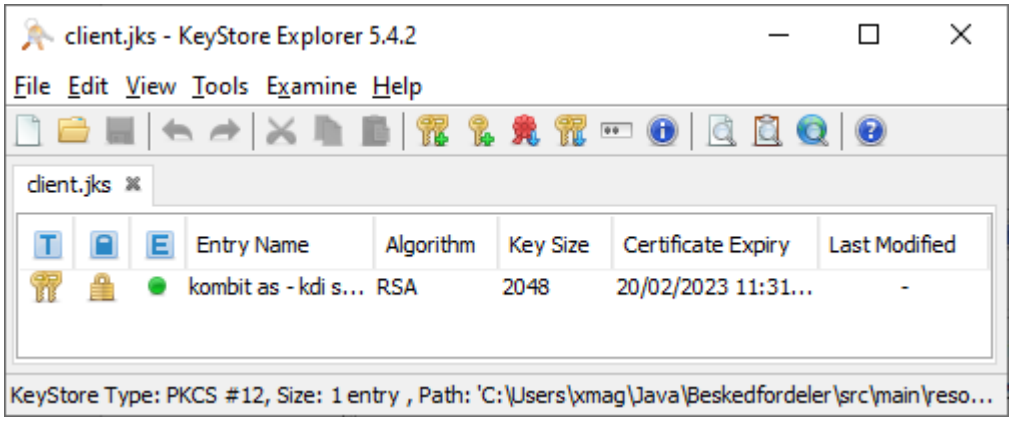


Som beskrevet tidligere skal du skal hente og registrere infrastrukturens certifikater på den maskine koden afvikles fra, på samme måde som dit eget certifikat. Der findes separate certifikater til Security Token Service (Adgangsstyring for systemer), Context Handler (Adgangsstyring for brugere), webservices og de fælleskommunale støttesystemer. Du behøver selvfølgelig kun at registrere de certifikater, der tilhører komponenter, du skal integrere med.

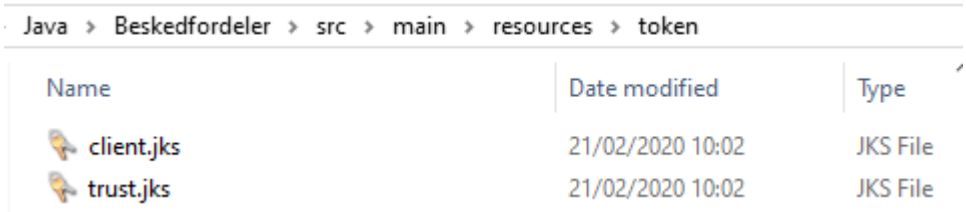
Eksempler på anvendelse af certifikater ved kald til webservices findes i .NET client for Serviceplatformens DemoService.

7. Java Keystore

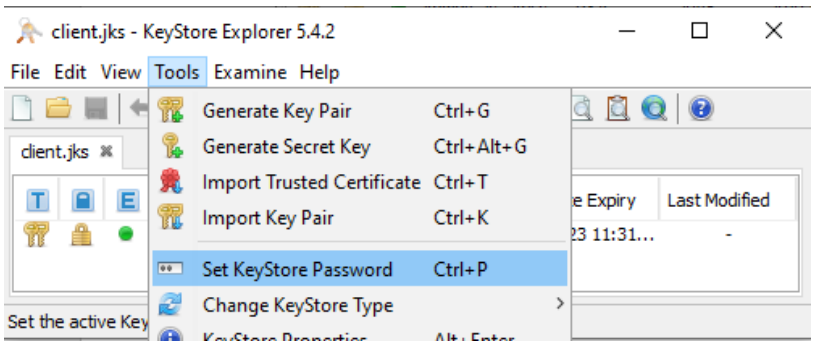
Til håndtering af certifikater på Java-platformen anbefales https://keystore-explorer.org/. Certifikater gemmes i Java KeyStore filer (.jks).



Eksempler på anvendelse af certifikater ved kald til infrastrukturen findes i demo-kode til Beskedfordeler (i dokumentationspakken) samt Java client for Serviceplatformens DemoService. De anvender begge .jks filer, der kan genanvendes, blot man udskifter relevante certifikater, men du kan med fordel lave to nye tomme keystores og kalde dem fx client.jks og trust.jks. Husk at skifte referencer til dem i koden.



Filen *client.jks* indeholder det private certifikat, der bruges ved kald til infrastrukturen. Dette skal du erstatte med dit eget. Vigtigt: Et keystore kan indeholde flere certifikater, men client.jks må kun indeholde et certifikat for dit anvendersystem. Det skal have samme adgangskode som selve certifikatet. Vælg ”Tools” i menu og dernæst ”Set KeyStore Password”:

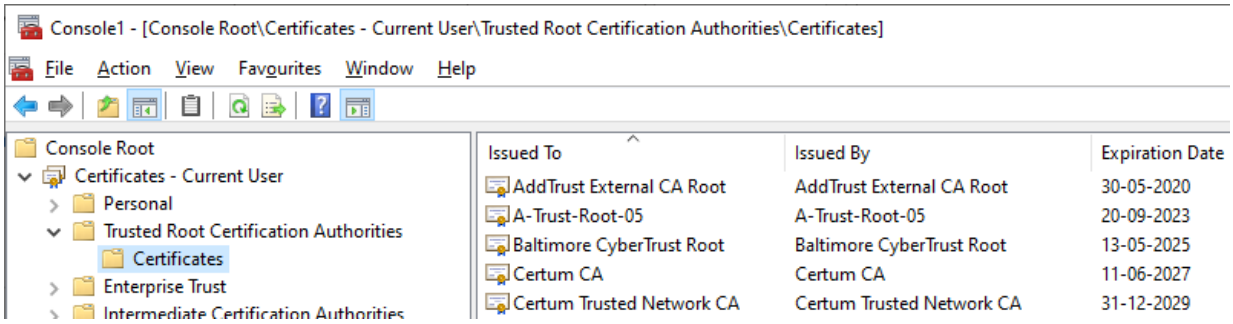


Filen *trust.jks* indeholder infrastrukturens certifikater for det eksterne testmiljø, og disse bør være gyldige, hvis du har hentet seneste version af demo-koden. Hvis de er udløbet, kan du hente seneste version fra infrastrukturens certifikater. Dette Keystore har ikke behov for at få sat en adgangskode, da det kun indeholder offentlige certifikater. Det er blot manglende CA du skal importere her, da certifikater anvendt til signing og krypteret forbindelse medsendes i svar fra endpoint.

8. Certification Authorities

For at et certifikat skal fungere korrekt, skal certification authorities (CA), der siger god for det, også være registreret.

Hvis kode afvikles i .NET skal CA være registreret i *Trusted Root Certification Authorities* (Current User eller Local Computer afhængigt af den context programmet kører i):



Hvis kode afvikles i Java skal CA være registreret i \lib\security\cacerts som er en underfolder til dit Java Runtime Environment. Du kan se indholdet og vedligeholde certifikater i dette med *keytool*:

```
..\jdk1.8.0_261\jre\lib\security>keytool -list -keystore cacerts
```

Eller alternativt kan du tilføje manglende CA til et Trust Keystore som du henviser til fra koden, som i eksempelkoden til webservices og Beskedfordeler.

Hvis dit Trust Store er af nyere dato, da indeholder det sandsynligvis allerede de nødvendige Root og Intermediate certifikater som indgår i Certification Chain for de certifikater der anvendes. Så det er ikke altid nødvendigt at tilføje CA til Trust Store. Det kan være nødvendigt hvis der anvendes self-signed certifikater, eller hvis Trust Store ikke er af nyere dato på den maskine koden afvikles fra.

Selve signing og HTTPS certifikater skal ikke tilføjes Trust Store! Disse returneres af endpoint du kalder. Det er blot deres CA der kan mangle.

Hvis et CA i certification path ikke er registreret på maskinen (Windows) vil det være markeret med et udråbstegn udfor pågældende certifikat under Certification Path. Følgende er et konstrueret eksempel der viser, hvordan du identificerer problemet og fikser det.

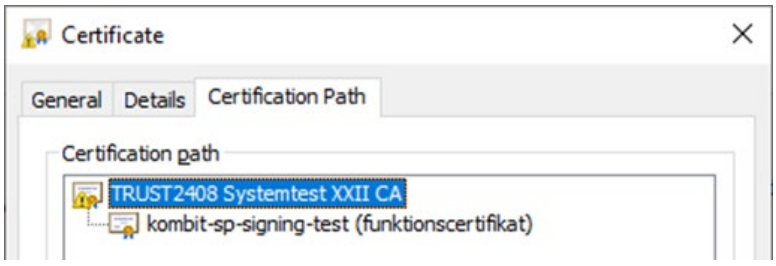
Filen "*Serviceplatformen Signing certifikat ExtTest (SPCER) - udløb maj 2022.txt*" inkluderer to CA. Du kan se tre certifikater hvis du åbner filen i en teksteditor.

```
-----BEGIN CERTIFICATE-----
MIIGITCCBQmgAwIBAgIEW6uwOTANBgkqhkiG9w0BAQsFADBIMQswCQYDVQQGEwJE
...
p+qspCBVu9Vru0UaGMNXyK0Ks5JIvHbYmL0RC2VASWItvePcft+mQKc6iBO+HCnp
WCtjVk5FlcfLxyV96UIUkqW4QABV1/2E0K0OsRppRTe51Z0EWp/Meid7ldMCSKtR
5e6+hgihN4nj8QQgk/jJJ9CDKDommMw9+rhG9VRh5kEeORmQ6A==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFQTCCAymgAwIBAgIEWBh+dDANBgkqhkiG9w0BAQsFADBPMQswCQYDVQQGEwJE
...
qFZ4ofDOwsr9fQ2jN+vdIX2ZPUlK5KBZ9Lo2CikaEQsxXLOv6PfBZqo6ukkOeqTq
nVYCW68=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGSDCCBDCgAwIBAgIES+pulDANBgkqhkiG9w0BAQsFADBPMQswCQYDVQQGEwJE
...
d/kID32R/hJPE41o9+3nd8aHZhBy2lF0jKAmr5a6Lbhg2O7zjGq7mQ3MceNeebuW
XD44AxIinryzhqnEWI+BxdlFaia3U7o2+HYdHw==
-----END CERTIFICATE-----
```

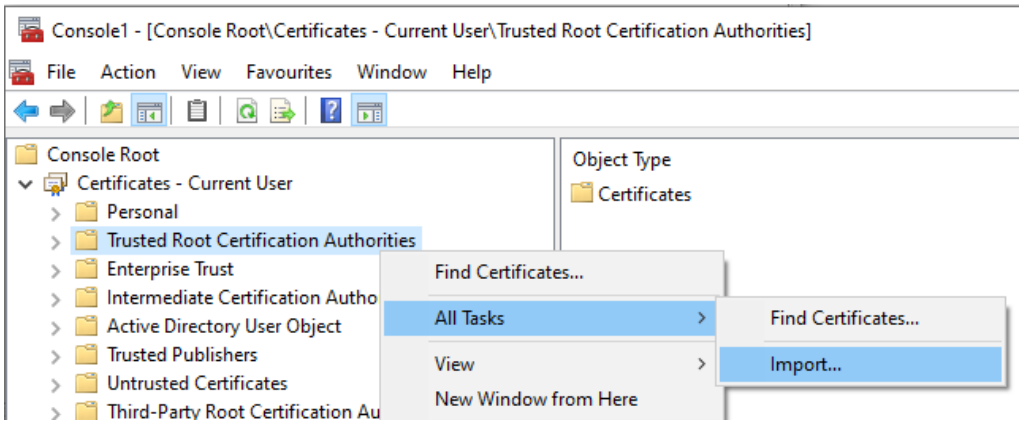
Hvis du gemmer de tre certifikater i hver sin .cer fil, aktiverer dem én efter én for at se detaljer, kopierer navnet fra "Subject, Common Name (CN)" og omdøber filen efterfølgende, da får du følgende tre certifikater som du kan importere individuelt:

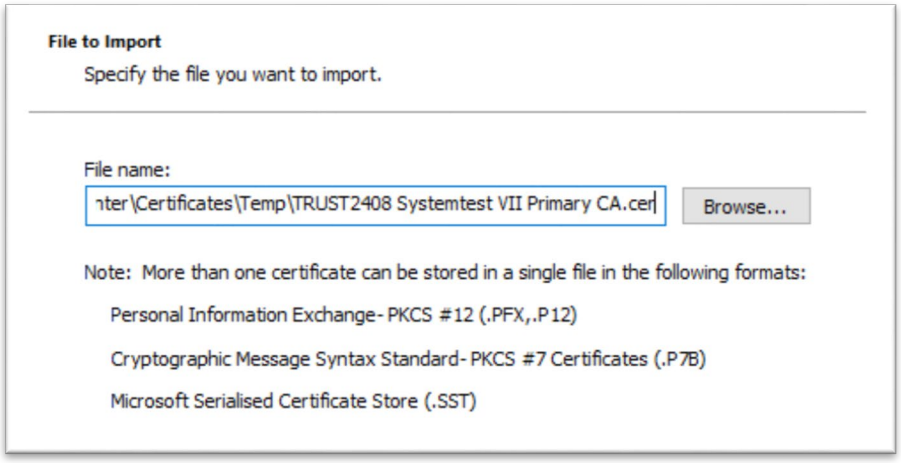
- kombit-sp-signing-test (funktionscertifikat).cer
- TRUST2408 Systemtest VII Primary CA.cer
- TRUST2408 Systemtest XXII CA.cer

Hvis et CA mangler i certification path er det markeret med et gult udråbstegn.

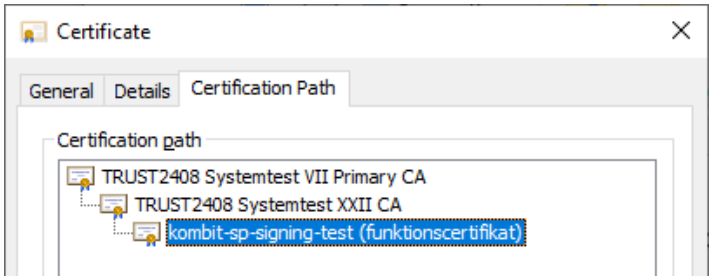


Vi skal da importere det manglende CA under *Trusted Root Certification Authorities*:

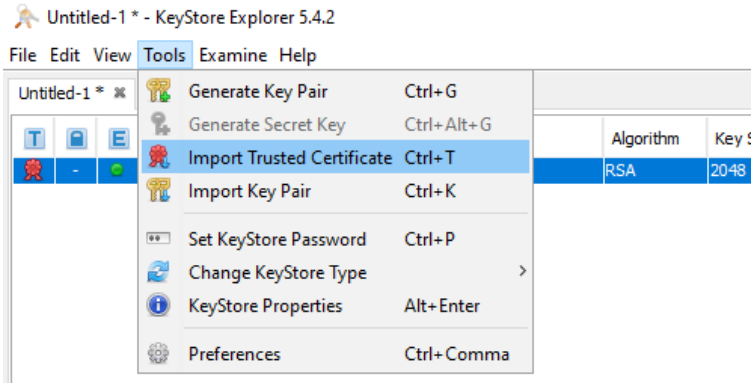




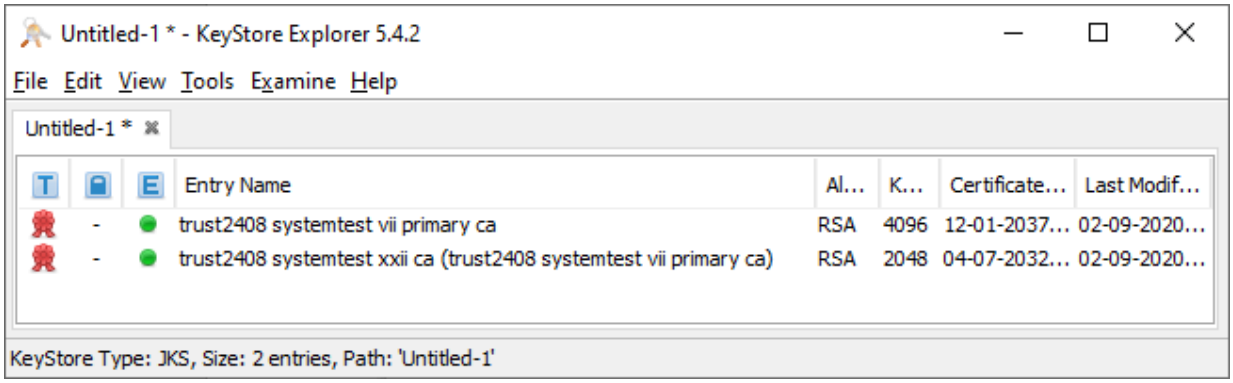
Det manglende CA er nu registreret og vi kan se at certification path er validt:



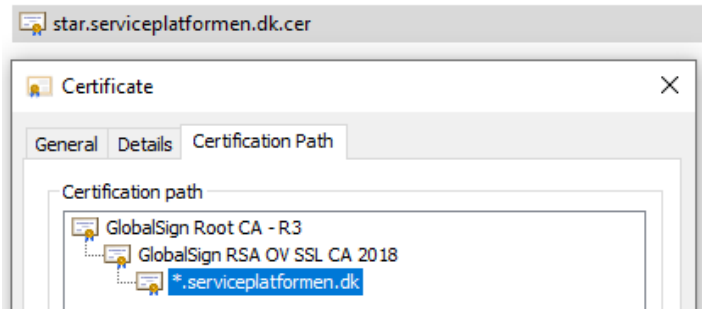
I et Java-miljø skal manglende CA importeres i det generelle eller lokale Trust store. Følgende eksempel viser proceduren for tilføjelse af CA til det lokale Trust store. Vælg import fra Tools menu og importér de to CA certifikater du gemte i separate filer.



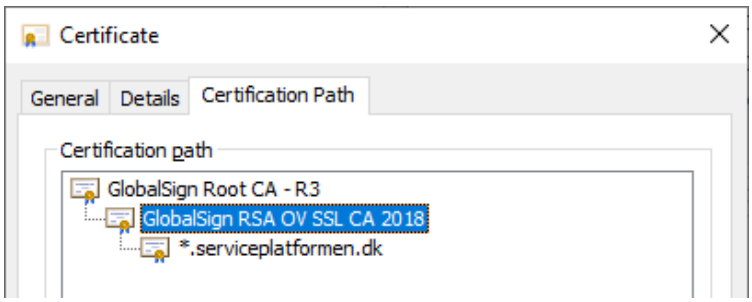
Her eksempel hvor vi har importeret de to CA tilhørende certification path for certifikat der benyttes til signering af beskeder i exttest:



Ikke alle certifikater er *chained*, dvs. også indeholder de tilhørende CA. I følgende tilfælde fordi de to CA allerede er på listen af Trusted root certificates i Windows.



Hvis disse CA mangler i dit Java Trust store, da kan du eksportere de to CA og efterfølgende importere dem i dit Java keystore. Du kan åbne CA fra certification path og derfra eksportere det, som illustreret tidligere i dokumentet.



Her er de to CA tilhørende HTTPS-certifikatet for serviceplatformen eksporteret fra Windows Certificate Store og importeret i lokalt Java keystore: