Aufbau der CA:

Wir haben für unsere CA folgende Verzeichnisstruktur:

* CA
  + certs
  + newcerts
  + private
  + serial
  + database.txt
  + caconfig.cnf

certs = In diesen Order liegen die Zertifikate

newcerts = In diesem Ordner liegen die neu erstellten Zertifikate

private= In diesem Ordner liegt der Private Schlüssel der Root-CA

serial= Seriennummer für das nächste Zertifikat

database.txt = Übersicht der bereits erstellten Zertifikate

caconfig.cnf= die Konfigurationsdatei für OpenSSL

Erstellen des Root Zertifikates:

Erstellen eines Privaten Schlüssels für die Root-CA und des Root-CA Zertifikates, anhand der in der caconfig angegeben Einstellungen.

openssl req -new -x509 -config caconfig.cnf -keyform PEM -keyout -🡪 private/cakey.pem -outform PEM -out certs/cacert.pem

Das Root-Zertifikat liegt im .pem Format vor. Windows beispielsweise nimmt aber nur Zertifikate im DER Format an, sodass man das .pem Zertifikat konvertieren muss:

openssl x509 -in certs/cacert.pem -outform der -out certs/cacert.crt

Serverzertifikat erstellen:

Auf dem Server ist dafür folgende Verzeichnisstruktur:

* MySSL
  + Public
  + Private

Zuerst erstellen wird ein neuer Privater Schlüssel erzeugt:

openssl genrsa -out private/privkey.pem 4096

Im nächsten Schritt wird eine Zertifikatsanfrage gestellt, welche dann von der CA verarbeitet werden wird. Die Angaben müssen dann korrekt mit der Signatur der Root-CA matchen:

openssl req -new -key private/privkey.pem -out cert-req.pem

Die Zertifikatsanfrage dann in das CA Verzeichnis legen, um es dann durch die CA zertifizieren zu lassen:

openssl ca -config caconfig.cnf -in cert-req.pem

Das neue Zertifikat liegt dann im Order newcerts. Die database.txt wird aktualisiert und der Zähler in serial hochgezählt.

Um nun noch das Zertifikat incl. Privatem Schlüssel in eine PKCS Datei zu konvertieren, kopieren wir das erstelle Server Zertifikat in den MYSSL Ordner und führen folgenden Befehl aus:

openssl pkcs12 -export -in Zertifikatname.pem –inkey privkey.pem -out PKCS12\_Datei.p12 -name "My Certificate"

Dateiverweise:

CA:

<https://www.dropbox.com/s/i4ofpzd7d5nc0nx/caconfig.cnf>  
<https://www.dropbox.com/s/nc9xrkhvfx6wz2u/cacert.pem>  
<https://www.dropbox.com/s/cmlajkb1p3p4zpl/cakey.pem>

Server:

<https://www.dropbox.com/s/62djlsh3fdxxdj6/privkey.pem>  
<https://www.dropbox.com/s/gtwcshfzlvgmtit/MYCert.pem>  
<https://www.dropbox.com/s/8vuys4jw5vpjxvs/pkcs.p12>  
<https://www.dropbox.com/s/y0hc3l4x38a9zhe/keystore.jks>

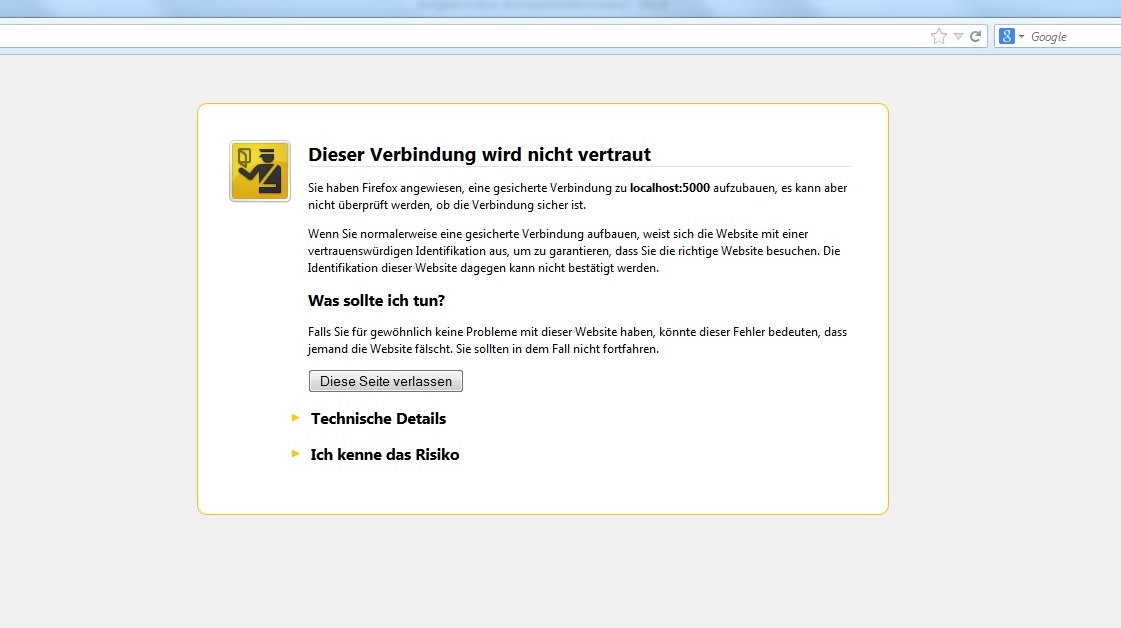
Webserver

Java Keystore Datei erstellen:

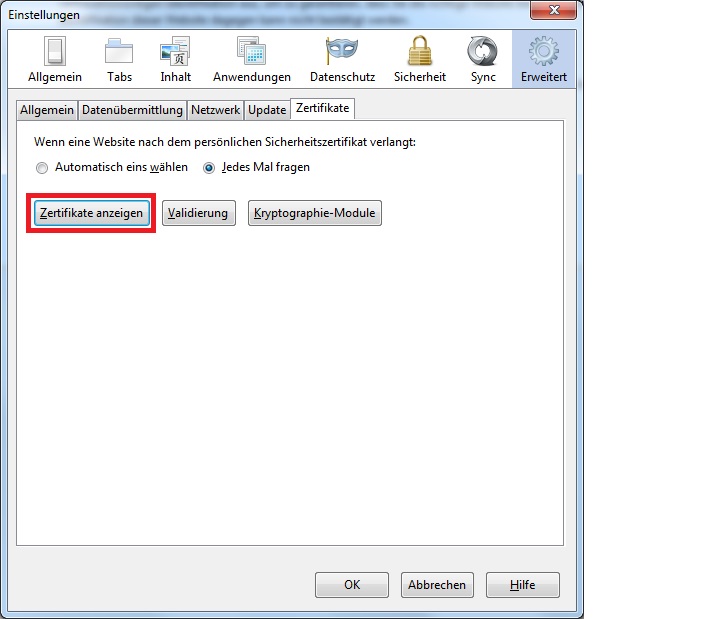
keytool -importkeystore -destkeystore \*pfad\*keystore.jks

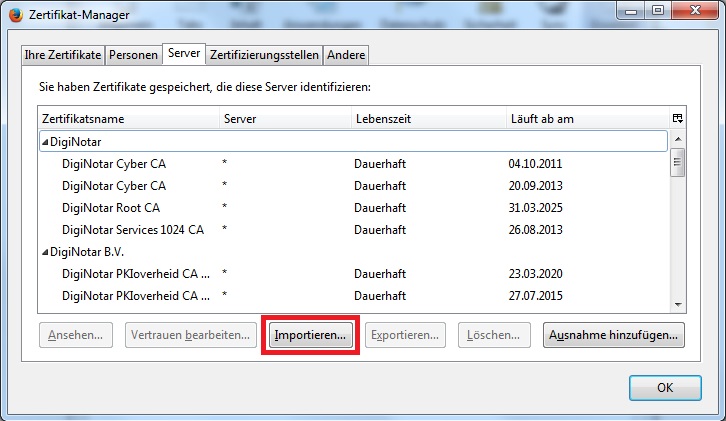
-srcstoretype PKCS12 -srckeystore \*pfad\*mypkcs12.p12

Nachdem der Webserver auf HTTPS umgestellt wurde, erscheint zuerst folgende Meldung beim aufruf der Webseite:



Dies kommt daher, weil der Browser Firefox noch nicht unser Root-CA Zertifikat kennt, und deshalb vertraut er diesem ausgestellten Zertifikat nicht. Um das zu beheben fügen wir unser Root-CA Zertifikat, in Firefox ein:





Nach dem Importieren muss noch das Vertrauen des Zertifikates Bearbeitet werden:

