**HTTPS**

****

Different teams which we can to secure a web service.

SSL- secure socket layer

TSL- Transport secure layer (More features added than SSL, Lot of certificates are created using TLS)

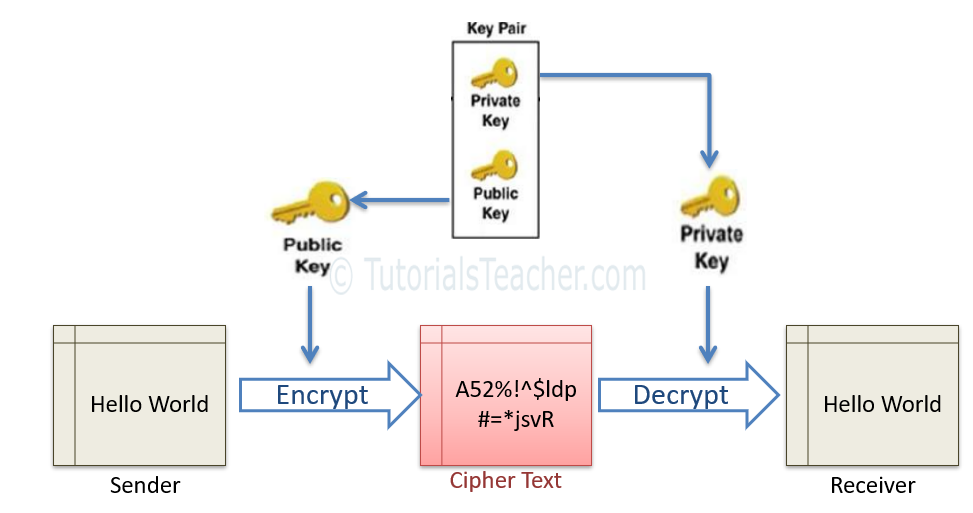
TrustStore and KeyStore

TrustStore - Certificate related to public information.

KeyStore- Certificate related to private information.

HPPTS (Hypertext transfer protocol secure) යනු HTTP හි secure version එක වන අතර එහි secure කියන එක ලගා කරගනු ලබනුයේ data encrypted කර transfer කිරීම මගින් වේ (secure communication between two parties by using the encryption mechanism it has).

**How Spring boot HTTPS works?**

****

HTTPS is used to make secure communication between two parties by using the encryption mechanism it has. In the case of SSL, it generates and HTTPS it generates an SSL certificate that contains the public and the private key, and other information which is needed to make it secure.

Now we will see how HTTPs works internally to secure the transfer of data over the network lets have a closer look at it.

HTTPS uses the encryption mechanism or protocol we can say to secure the communication, this protocol is known as TLS, which stands for Transport Layer Security. This protocol uses a key to transfer data among the two parties and hence encrypt the communication among them. Below we have the following type of key which are described as follows;

**1) Public key:** This type of key is public means available to everyone who wants to interact with the server in a secure way. We encrypt some information by using the public key and that information can only be decrypted by using the private key at another end.

**2) Private key:** This type of key is not at all public to everyone because it is controlled by the owner itself. This key we kept on the server to decrypt the public key information.

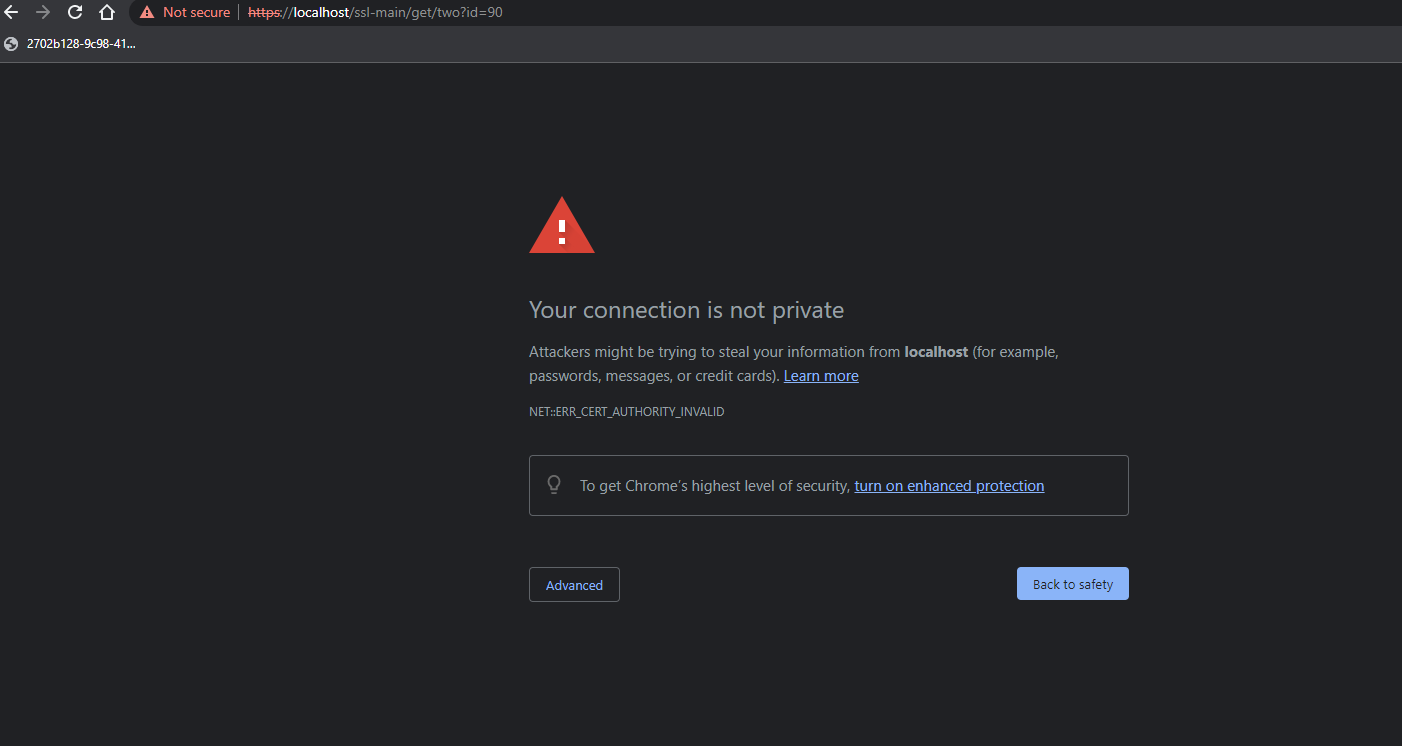
Now we will see one basic flow for HTTP and HTTPS and underrated it step by step let’s get started,

1) HTTPS is a secure layer between the HTTP and the TCP, and this layer is nothing but the TLS which stands for Transport Layer socket, HTTPS runs on port 443.

2) If we talk about HTTP then it does not really use secure communication for data transfer. There is no layer like TLS in between.

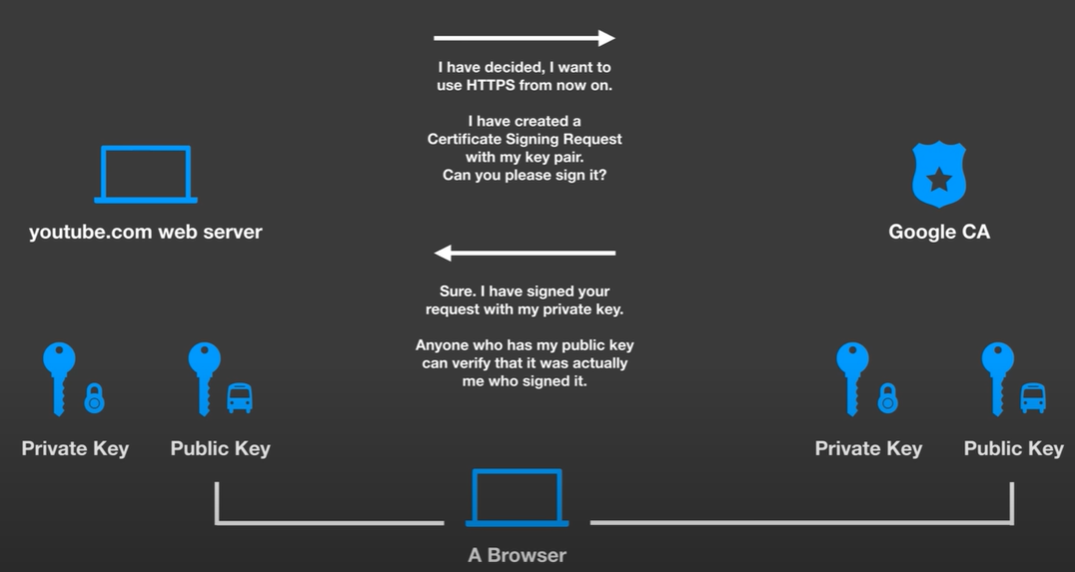


Browser එකෙන් HTTPS request එකක් යැවුවම server එකෙන් public key එක සහ certificate එක browser එකට දෙනු ලබයි.Browser එකෙන් එකම certificate එක validate කරනු ලබනවාi එහිදී, Browser එකත් එක්කම by default ම (list of trusted certificate තියනවා) authorized certificate issuers ලගේ certificate තියාගෙන සිටින අතර ඒවා use කරලා තමා server එකෙන් දෙන certificate එක valid certificate එකක්ද නැද්ද බලනේ.issuer certificate එක දෙද්දී එකෙත් private key and [public key දෙකක් ඇති අතරa public key එක use කරලා server එකෙන් දුන්න certificate එක අදාළ certificate issue කල company එකෙන් දුන්න එකක්ද යන්න validate කරලා බලනු ලබයි.නමුත් self-sign certificate වගේ එවගේදී ඊට අදාළව browser එකේ මුකුත් නැතිනිසා server එකන් provide කරන certificate එක validate කරගන්න browser එකට බැරි නිසා bowser එකෙන් warning එකක් දෙනු ලබයි.

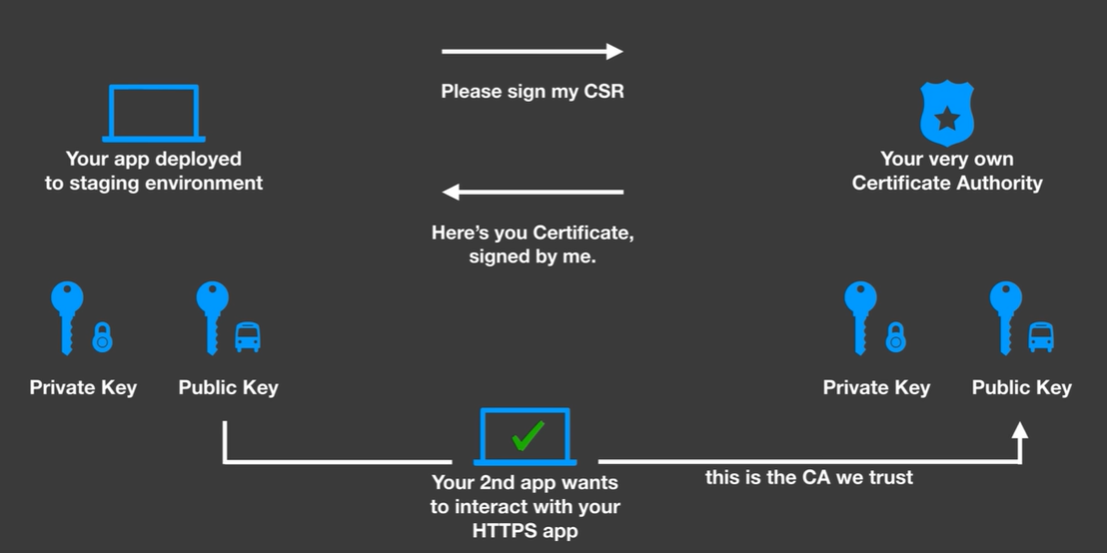


ඊට පස්සේ වන process එක වනුයේ browser එකෙන් වෙනම key එකක් generate කරල (Session key) එම key එක certificate එකත් එක්ක දුන්න public key එකෙන් encrypt කරලා server එකට දෙනු ලබයි.මෙය decrypt කර ගන්න පුලුවන් වනුයේ server එකේ ඇති private key එකෙන් පමණක් වේ.server එකෙන් එම key එක decrypt කරලා තබා ගනී.පසුව එම session key එක use කර data encrypt decrypt කරමින් data secure ලෙස transfer කරගනු ලබයි.

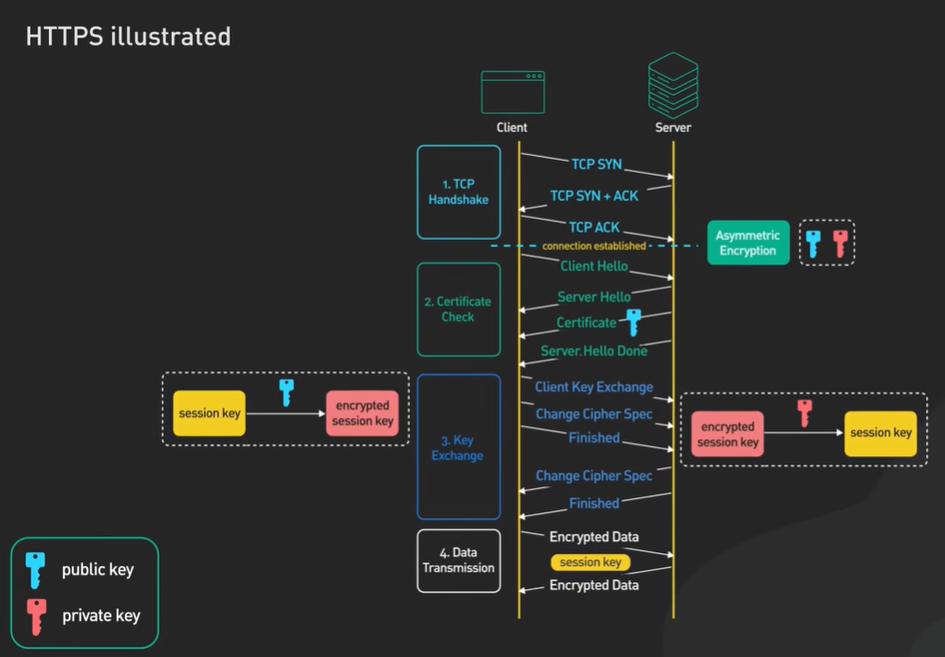
**Server එක certificate එක generate කරගන්නා ආකාරය.**

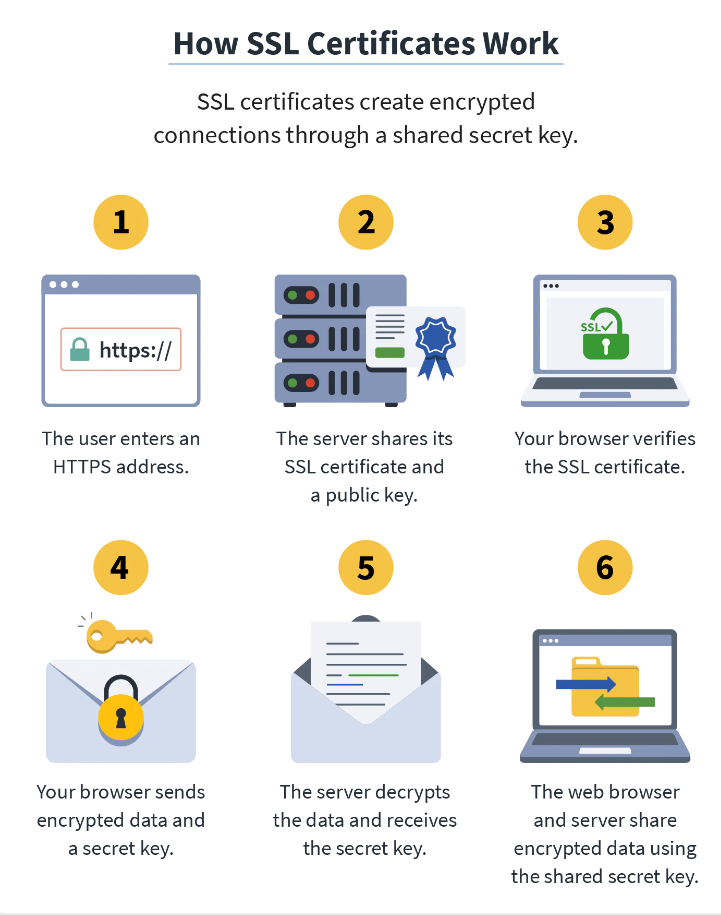


**Server - server communicate කරගන්නා ආකාරය**



**SSL hand shake එක කරගන්නා full process එක.**





**Generate a certificate and add to project path and create configuration.**

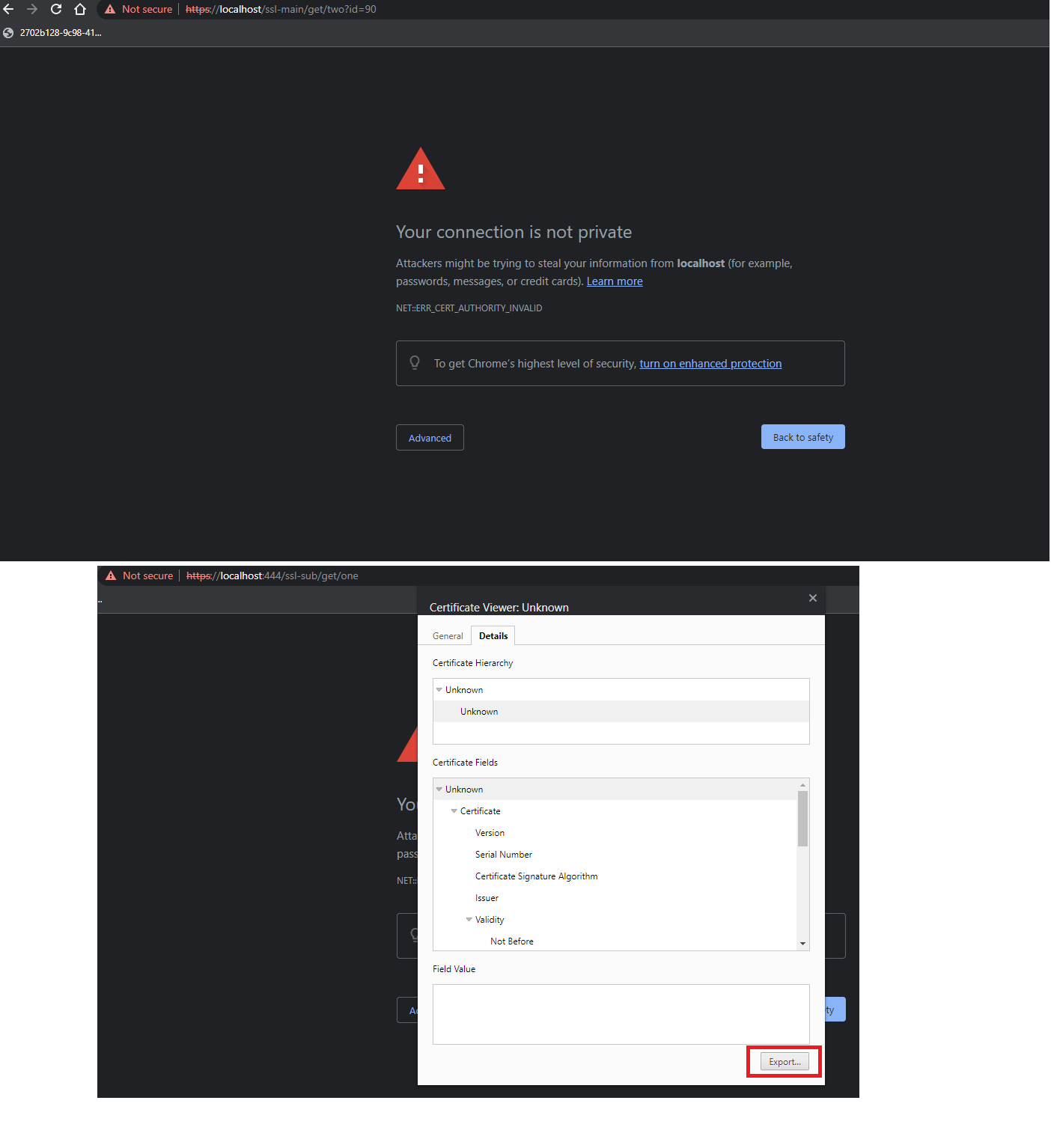
Solving Browser showing warning.

Run the application, then browser shows that warning. That means that browser cannot authorized self-sign certificate. To solve that export certificate and then install OR download the certificate from the browser and then install. After that close and re run the browser. Problem will be solved.

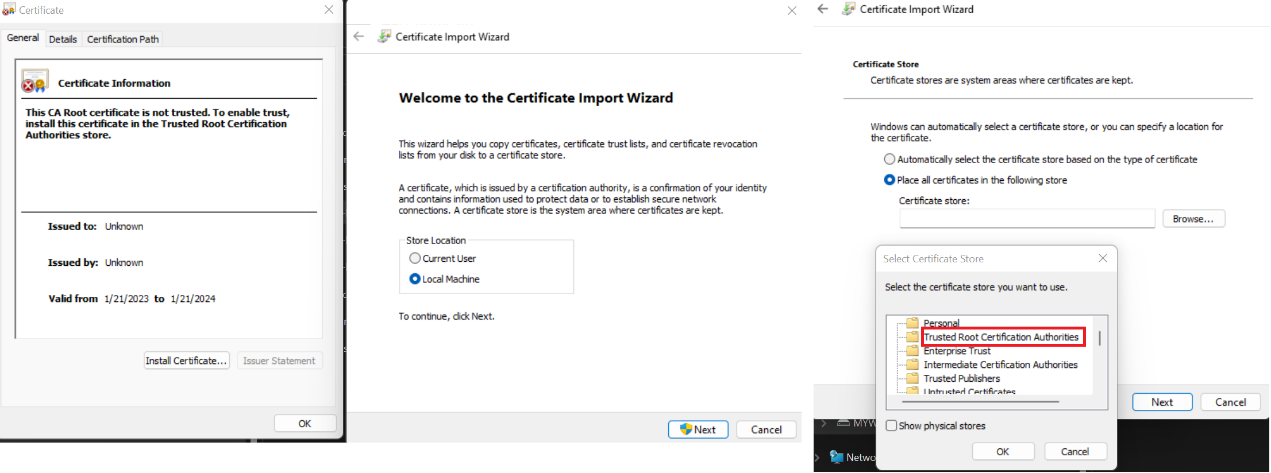
01.Export certificate

keytool -export -alias main-app-keystore-one -storepass 123456789 -file main-app-keystore-one.crt -keystore main-app-keystore-one.p12 -storetype pkcs12

02.Download the certificate

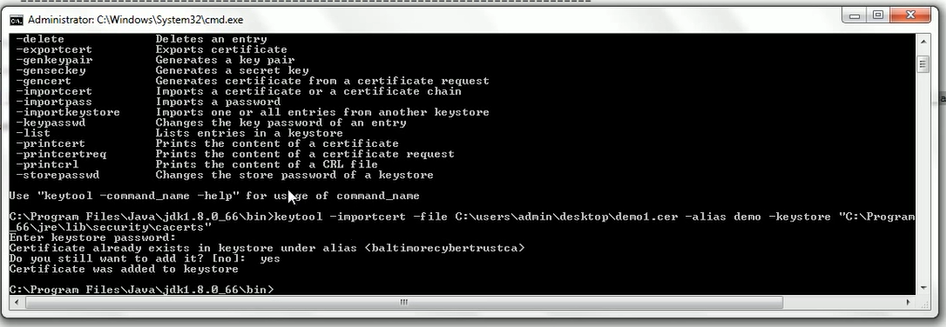


03.Installing certificate.

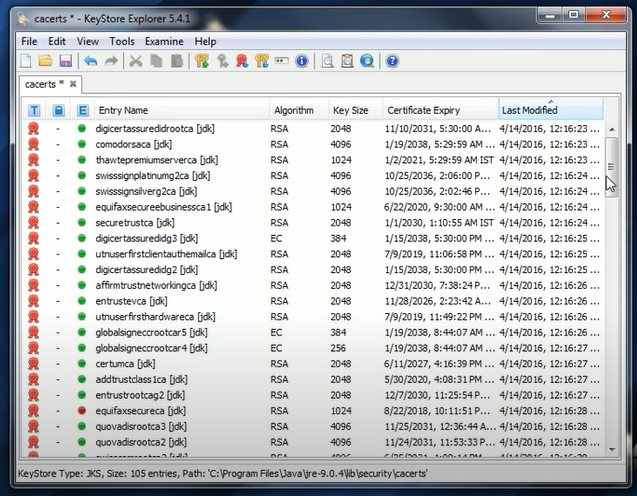


Server server communication එකක්නම් කරන්නේ මෙවිට certificate needs to be add to JRE certificate. මෙය ආකාර දෙකකින් කරගත හැක.එනම් Using CMD (through import ) සහ KeyStore Explorer use කිරීම හරහා.

01.Using CMD (through import )



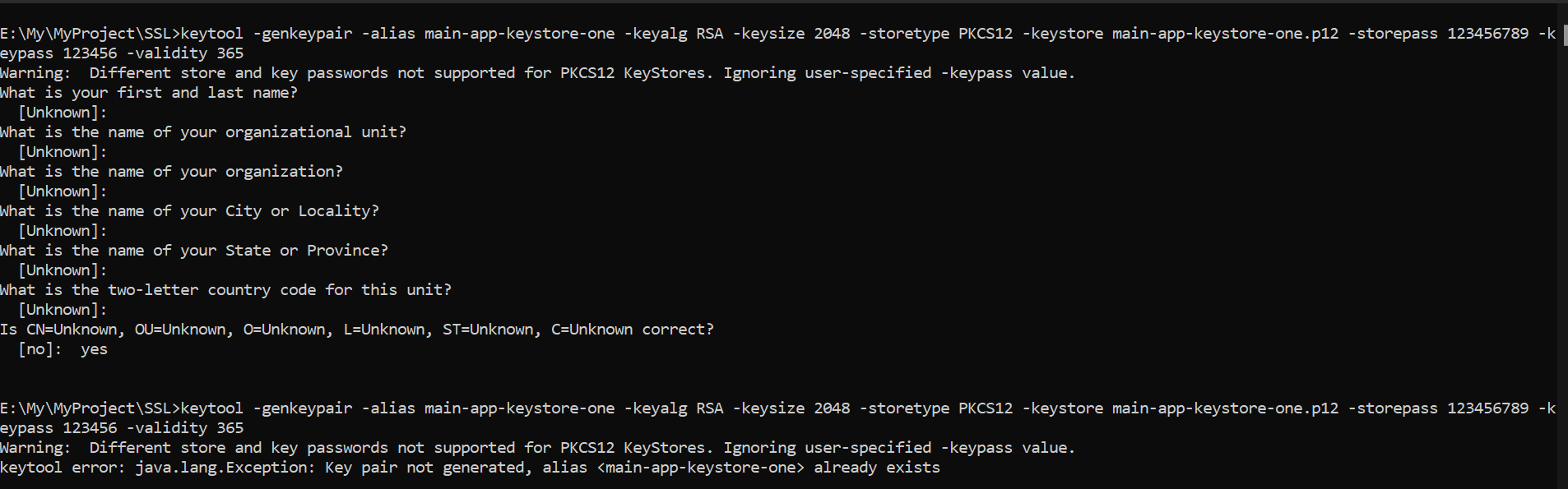
02.KeyStore Explorer



**Project creation**

Create a spring boot project by adding web dependencies

Then create keypair



Add it to project class path

Add bellow configuration and run the project.

