IAM Client

Revision by: Sergio-Feliciano Mendoza-Barrera

February 3, 2022

IAM client in Java. The GitHub repository is here.

Table of content

- *Introduction*
- *Configure IAM-Client*
- *How to run IAM-Client service*
- *Functionalities*

Introduction

IAM-Client service motivation

IAM-Client service is an IAM's client side application used to generate IAM login url using the client certificates. and its main motivation is to ease the intgration process to IAM service.

Technologies

- `Java 8`
- `Spring boot`
- `Spring web REST`
- `Lombok`
- `Java key store`
- `Maven`

Configure IAM-Client

In order to use the service properly we need to fill some properties from application.properties file:

To build IAM URL, search for each key in the specified file and fill it with the proper value

In the properties file the client will find a group of properties start with `iam.request.url` this group will help the client build IAM request url. Some of the values already filled the client may keep it as it is.

property	key	value explain
host	`iam.request.url.host`	Check if the client is using the staging or production host.
client-id	`iam.request.url.client-id`	it is the *reference number* giving to the client.
redirect-uri	`iam.request.url.redirect-uri`	*redirect-uri is a static* please use the submitted redirect-uri to the NIC **without queries or extra path**.

NOTE: *for Production please use the following host:* `https://iam.elm.sa/authservice/authorize`

NOTE: *for Staging please use the following host:* `https://iambeta.elm.sa/authservice/authorize`

Another group will start with `jks.store` this group will help the client refereing to the key store

property	key	value explain
path	`jks.store.path`	the full system path to the key store .
pass	`jks.store.pass`	provide the password for the key store.
store-type	`jks.store.store-type`	specify the key store type. (e.g. JKS)

lastly the client is going to refer to the certificates, the last group of properties will start with `certificate.client`, which will help to fetch certificate from the configured key store.

property	key	value explain
private.alias	`certificate.client.private.alias`	refer to the private key alias that giving to the certificate once imported to the key store.
private.password	`certificate.client.private.password`	refer to the private key's password of the certificate.
	`certificate.client.public.alias`	refer to the public key alias that giving to the certificate once imported to the key store.
public.password	`certificate.client.public.password`	refer to the private key's password of the certificate *if exists*.

Once you fill the previous key value pairs you are ready to run the application

In the following section you will know how to generate IAM url and how to validate it.

How to run IAM-Client service

- Linux remote server
- TO-DO

Functionalities

IAM-Client service exposes the following rest endpoints:

Generate IAM url

The client can directly generate login url by hitting the rest endpoint:

GET `http://localhost:8088/url`

and it will return back login url as string and you may use it to test.
NOTE: *In order to access IAM servers the client server need to be configured in the NIC.*

Validate IAM url

The client may validate login url by hitting the rest endpoint:

```
POST http://localhost:8088/url

@RequestBody
{"url":"https://iambeta.elm.sa/authservice/authorize?..."}
```

and it will return back validation response with `HTTP_STATUS 200` if it is valid, and with `HTTP_STATUS 422` with error description if it is invalid login url.

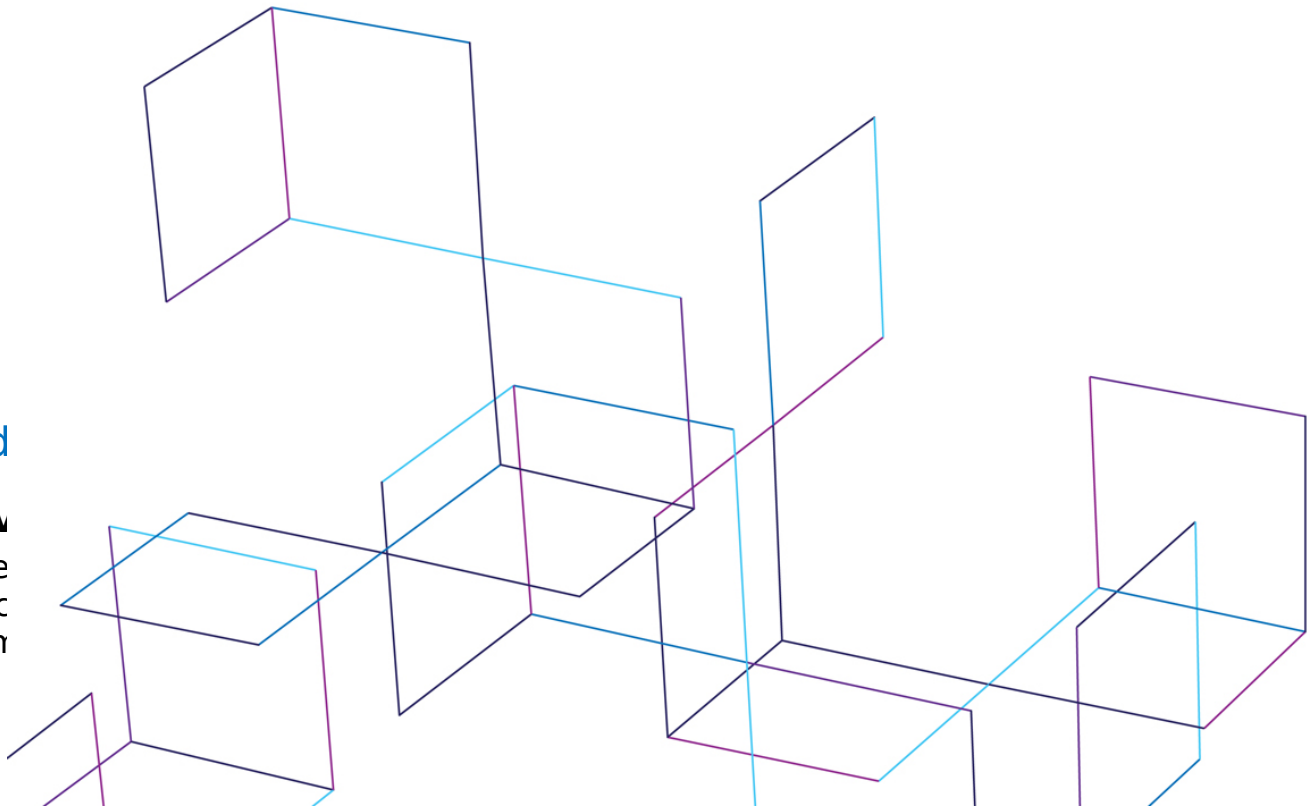
Extra documentation

The following pages are documents related to the IAM, NIC and ELM services.
EOF

"IAM"- Service Requirements & Specifications

Version 0.3

Date 28//01/2021



Introd

Overv

Service
transac
econom

the difficulty of managing and authenticating users online. User Registration and validation is a long and tedious phase, which consumes money and effort.

IAM comes to the picture to take away the burden of managing citizen and residents' digital identity. It is the Saudi National Identity Provider with solid way of identifying people online with unique digital identity. IAM has the ability to provide assurance to electronic service providers the identity of the individual seeking to obtain their services.

Definitions, Acronyms, and Abbreviations

Term	Definition
IAM	Identity and Access Management

Service Provider Zone

Customer General Information (need to be filled by customer)

Official name	**Mohammed ammar alahmed**
C.R. No	**1076516739**
Mobile number	**0563114446**
Email	**Mohdmedic1@gmail.com**

Service General Information (need to be filled by customer)

Service Provider Name	*to Be Filled By the Client* **Required** Ejarly
Service Name	*to Be Filled By the Client* **Required** *Ejarly*

Service Requirements& Specifications

Service Description/ Platform name	*to Be Filled By the Client* **Required** *Ejarly is a rental platform that allows community members to share their things with each other, and we enable property owners to benefit from their abandoned and untapped things by making money from them by renting them to other individuals, and we connect them with individuals who need things for short-term use or for individual use to rent them and save money. The platform connects and facilitates communication of the rental process between the two parties in a safe and reliable environment*
Technology	*to Be Filled By the Client* **Required** Javascript
Application server	*to Be Filled By the Client* **Required** PHP/Laravel/MYSQL
Current Authentication Scheme	*to Be Filled By the Client* **Required** Basic Authentication

Pre – production service profile and URLs details (need to be filled by customer)

Entity ID	***This URL will be unique identity of the Service Provider and it should be accessible through internet.*** *(Ex.* https://preprod.example.com) *to Be Filled By the Client* **Required** *https://preprod.ejarly.net*

Service Requirements& Specifications

Service Provider Callback	***This URL will be the OpenID Connect callback of the service provider and it should be accessible through internet.*** *(Ex.* https://preprod.example.com/auth/authorize/callback) *to Be Filled By the Client* **Required** *https://preprod.ejarly.net/auth/authorize/callback*
Service Provider Login URL	***Login URL to Receives and Processes the IAM authentication response.*** *(Ex.* https://preprod.example.com/_IAM/login) *to Be Filled By the Client* **Required** ***https://preprod.ejarly.com/_IAM/login***
Service Provider Logout URL	***Receives and Processes the IAM logout response simple/SLO request.*** *(Ex.* https://preprod.example.com/_IAM/logout) *to Be Filled By the Client* **Required** *https://preprod.ejarly.net/_IAM/logout*

Production service profile and URLs details (need to be filled by customer)

Entity ID	***This URL will be unique identity of the Service Provider and it should be accessible through internet.*** *(Ex.* https://example.com) *to Be Filled By the Client* **Required** *https://ejarly.net*

Service Requirements& Specifications

Service Provider Callback	***This URL will be the OpenID Connect callback of the service provider and it should be accessible through internet.*** *(Ex.* https:// example.com/auth/authorize/callback) *to Be Filled By the Client* **Required* *https:// ejarly.net/auth/authorize/callback*
Service Provider Login URL	***Login URL to Receives and Processes the IAM authentication response.*** *(Ex.* https://example.com/_IAM/login) *to Be Filled By the Client* **Required* ***https://ejarly.net/_IAM/login***
Service Provider Logout URL	***Receives and Processes the IAM logout response simple/SLO request.*** *(Ex.* https://example.com/_IAM/logout) *to Be Filled By the Client* **Required* *https://ejarly.net/_IAM/logout*

Custom User Attributes

Please list any not mentioned in the attribute list in the Appendix A

#	Attribute Name	Type	Description
1	Info1	String	This attribute is additional as per SP request
2	Info2	Date	This attribute is additional as per SP request
3			
4			

Appendices

Appendix A: User Attributes

#	Attribute Name	Type	Description
1	nationalId	String	This is the user identifier represented by the National Id (Resident Id) SAML2 NameID or http://iam.gov.sa/claims/userid
2	lang	Enum	For Language Consistency/Preferred Language of the user (AR/EN) http://iam.gov.sa/claims/lang
3	arabicName	String	Arabic Full Name

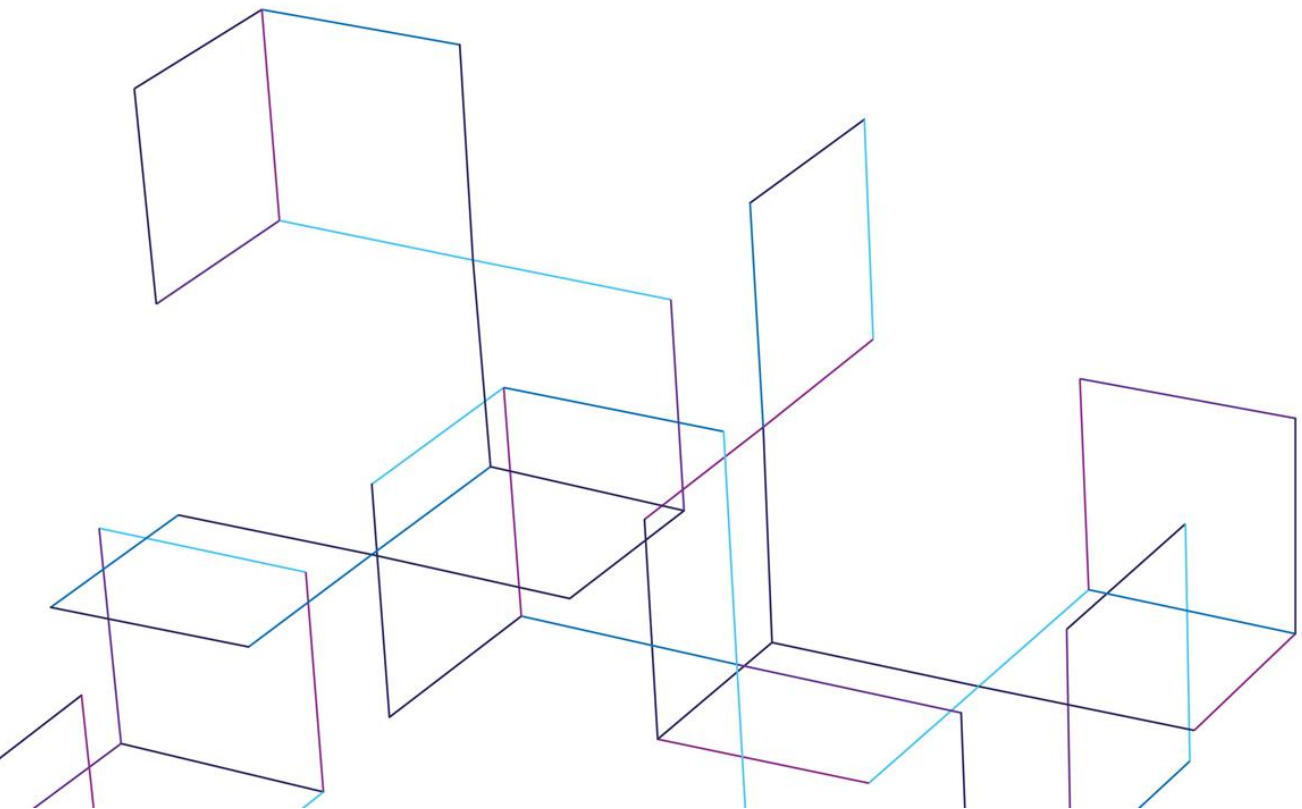
			http://iam.gov.sa/claims/arabicName
4	englishName	String	English Full Name http://iam.gov.sa/claims/englishName
5	dobHijri	Date	Date Of Birth Hijri Example: 1487/06/12 http://iam.gov.sa/claims/dobHijri
6	dob	Date	Date Of Birth Gregorian Example: Tue Feb 30 03:00:00 AST 1987 http://iam.gov.sa/claims/dob
7	arabicNationality	String	Arabic Nationality http://iam.gov.sa/claims/arabicNationality
8	nationality	String	English Nationality http://iam.gov.sa/claims/nationality
9	nationalityCode	String	Nationality code, list of codes are in the Annex D. http://iam.gov.sa/claims/nationalityCode
10	gender	Enum	Male/Female http://iam.gov.sa/claims/gender
11	arabicFirstName	String	Arabic First Name http://iam.gov.sa/claims/arabicFirstName
12	englishFirstName	String	English First Name http://iam.gov.sa/claims/englishFirstName
13	arabicFamilyName	String	Arabic Family Name http://iam.gov.sa/claims/arabicFamilyName
14	englishFamilyName	String	English Family Name http://iam.gov.sa/claims/englishFamilyName
15	arabicFatherName	String	Arabic Father Name http://iam.gov.sa/claims/arabicFatherName
16	englishFatherName	String	English Father Name http://iam.gov.sa/claims/englishFatherName
17	arabicGrandFatherName	String	Arabic Grand Father Name http://iam.gov.sa/claims/arabicGrandFatherName
18	englishGrandFatherName	String	English Grand Father Name http://iam.gov.sa/claims/englishGrandFatherName
19	assuranceLevel	String (Optional)	Level of Assurance according to the authentication sequence and the status of the user registration http://iam.gov.sa/claims/assuranceLevel
20	cardIssueDateGregorian	Date	Gregorian Saudi Identity Card Issue Date or Iqama Issue Date Example: Tue Jan 20 03:00:00 AST 2015 http://iam.gov.sa/claims/cardIssueDateGregorian
21	cardIssueDateHijri	Date	Hijri Saudi Identity Card Issue Date or Iqama Issue Date Example: 1436/09/29 http://iam.gov.sa/claims/cardIssueDateHijri
22	IssueLocationAr	String	Card Issue Location, Example: Riyadh http://iam.gov.sa/claims/issueLocationAr
23	IssueLocationEn	String	Card Issue Location, Example: الرياض http://iam.gov.sa/claims/IssueLocationEn
24	iqamaExpiryDateHijri	Date	Hijri Iqama Expiration Date Example: 1436/09/29 http://iam.gov.sa/claims/iqamaExpirationDateH
25	iqamaExpiryDateGrego	Date	Gregorian Iqama Expiration Date

	rian		Example: 2017/09/29 http://iam.gov.sa/claims/iqamaExpirationDateH
26	idExpiryDateHijri	Date	Hijri Id Expiration Date Example: 1436/09/29 http://iam.gov.sa/claims/idExpirationDateH
27	idExpiryDateGregorian	Date	Gregorian Id Expiration Date Example: 2017/09/29 http://iam.gov.sa/claims/idExpirationDateH
28	versionNumber	String	Version of the identity Example: “2” http://iam.gov.sa/claims/versionNumber

IAM Authentication Service Integration Guide

Version 0.5

Date 27/04/2021



Content

Introduction 3

OpenID Connect Authentication Business Scenario 3

OpenID Connect Authentication Service Integration 6

IAM Service Provider Certificate Issuance Procedure 10

Code Sample 10

Document Writers

Written By	Date	Issue	Note
Firas Moalla	07/03/2019	01	Author
Firas Moalla	08/04/2019	02	Updated Integration Requirements and Provided Certificate Integration Guide
Firas Moalla	21/05/2019	03	Staging and Production Environment
Firas Moalla	01/07/2019	04	Linux Certificate CSR and max_age configuration
Firas Moalla	Add Date	05	IAM Productino URL
	Add Date	06	
	Add Date	07	
	Add Date	08	
	Add Date	09	
	Add Date	10	
	Add Date	11	
	Add Date	12	
	Add Date	13	

1. Introduction

1.1 Purpose

The purpose of this document is to provided a high level specification for IAM authentication system through partner platform using a OpenID Connect model as the main driving protocol between all involved parties. The partner is responsible for the integration process with the service provider under IAM management governance.

1.2 Scope

The scope of this document is the technical integration between the Service Provider and Elm for IAM authentication services.

1.3 Definitions

- IAM: the entity that delivers end-user identification.
- Service Provider: the entity that delivers the online service.
- Elm: the broker between IAM and the service provider.
- End-User: Saudi Arabian citizen or resident.

2. Authentication Service Business Scenario

The business scenario of IAM authentication is summarized in Figure.1. This business scenario is a high level communication description involving IAM System, Service Provider, Em, and the End-User.

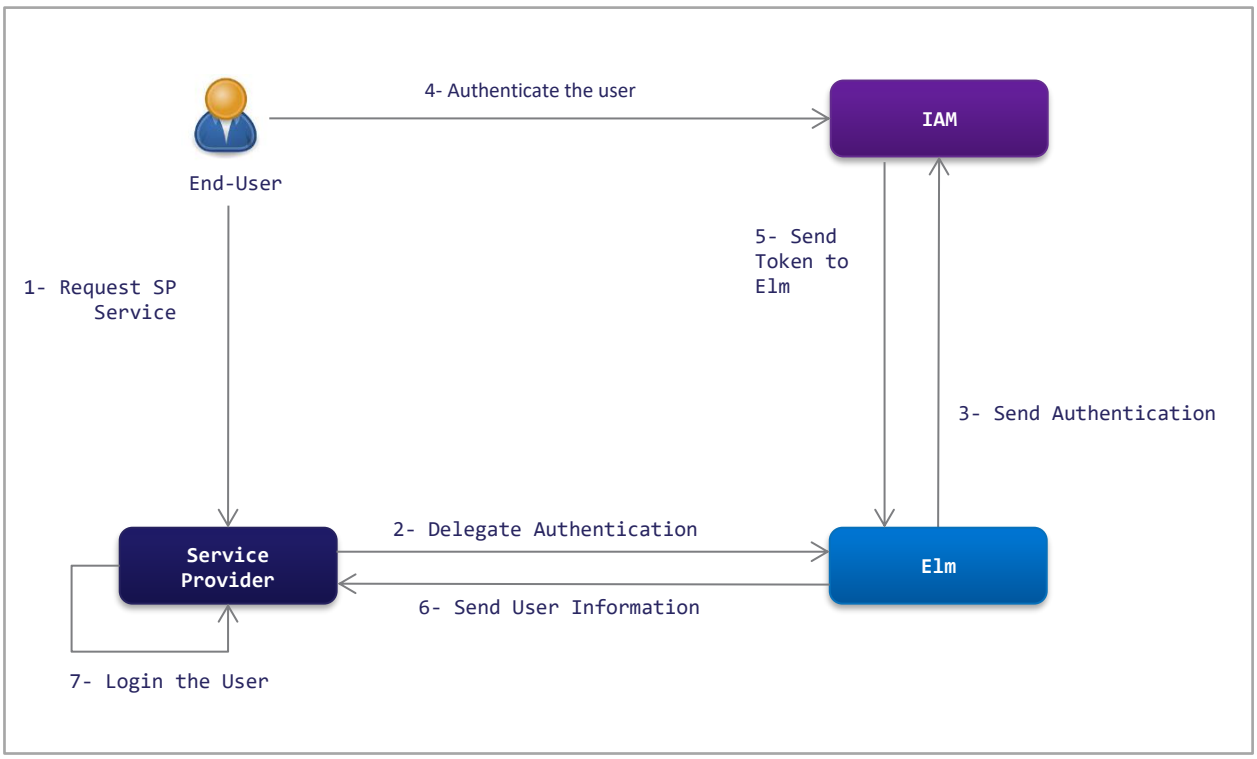


Figure.1: IAM Authentication Business Scenario

The authentication scenario that corresponds to Figure.1 is as follows:

1. The End-User requests to login to the Service Provider.
2. The Service Provider builds an OpenID Connect authentication request and signs the OpenID Connect request with his certificate key, and then sends the signed OpenID Connect request to Elm through End-User browser redirection.

   Note that the Service Provider certificate is issued from IAM PKI Infrastructure.
3. Elm validates the OpenID Connect request of the Service Provider and redirects the user to IAM login page for authentication.
4. IAM authenticates the End-User according to the Service Provider policy.
5. IAM sends the token to Elm.
6. Elm sends a signed user token (id_token) to the Service Provider through End-User browser redirection.
7. The Service Provider receives and validates the user token (id_token) and extracts user information for authentication.

The user token includes the information in Table.1 below.

#	Attribute Name	Type	Description
1	nationalId	String	This is the user identifier represented by the National Id (Resident Id)
2	lang	Enum	For Language Consistency/Preferred Language of the user. Example: AR/EN
3	arabicName	String	Arabic Full Name
4	englishName	String	English Full Name
5	dobHijri	Date	Date Of Birth Hijri. Example: 1487/06/12
6	dob	Date	Date Of Birth Gregorian. Example: Tue Feb 30 03:00:00 AST 1987
7	arabicNationality	String	Arabic Nationality
8	nationality	String	English Nationality
9	nationalityCode	String	Nationality code, list of codes are in Appendex A
10	gender	Enum	Male/Female
11	arabicFirstName	String	Arabic First Name
12	englishFirstName	String	English First Name
13	arabicFamilyName	String	Arabic Family Name
14	englishFamilyName	String	English Family Name
15	arabicFatherName	String	Arabic Father Name
16	englishFatherName	String	English Father Name
17	arabicGrandFatherName	String	Arabic Grand Father Name
18	englishGrandFatherName	String	English Grand Father Name
19	assuranceLevel	String (Optional)	Level of Assurance according to the authentication sequence and the status of the user registration
20	cardIssueDateGregorian	Date	Gregorian Saudi Identity Card Issue Date or Iqama Issue Date Example: Tue Jan 20 03:00:00 AST 2015
21	cardIssueDateHijri	Date	Hijri Saudi Identity Card Issue Date or Iqama Issue Date Example: 1436/09/29
22	IssueLocationAr	String	Card Issue Location, Example: Riyadh
23	IssueLocationEn	String	Card Issue Location, Example: الرياض
24	iqamaExpiryDateHijri	Date	Hijri Iqama Expiration Date Example: 1436/09/29
25	iqamaExpiryDateGregorian	Date	Gregorian Iqama Expiration Date Example: 2017/09/29
26	idExpiryDateHijri	Date	Hijri Id Expiration Date Example: 1436/09/29
27	idExpiryDateGregorian	Date	Gregorian Id Expiration Date Example: 2017/09/29
28	versionNumber	String	Version of the identity Example: “2”

3. OpenID Connect Authentication Service Integration

IAM Authentication uses OpenID Connect to authenticate users; The authentication service is provided to the Service Provider with two functionalities: the login service and the logout service. Login and logout services are based on End-User browser redirection. Therefore, there is no back-to-back integration channel between the Service Provider and Elm which facilitates the integration process. What follows explains the login and logout services from a technical perspective, respectively.

3.1 OpenID Connect Authentication: Login Service

The login service uses OpenID Connect protocol as follows:

1) Authentication Request - from Service Provider to Elm

   After the user clicks on the login button in the Service Provider's web page, the service provider builds the following OpenID Connect Authentication request and sends it to Elm.

https://iam.elm.sa/authservice/authorize?
scope=openid
&response_type= id_token
&response_mode=form_post
&client_id=<CLIENT_ID> (provided by Elm)
&redirect_uri=https://www.service_provider.com/callback
&nonce=GUID_RANDOM (example: b55224f7-e83d-4250-aa4a-451d32666e59)
&ui_locales=ar
&prompt=login
&max_age=timestamp (the current time in seconds using local Saudi Arabia time)
&state=<Signed Message Signature>

The query string must be in the same order as above. Note that max_age represents the current time in seconds when the OPIC request was generated; The generated time of max_age (in seconds) must match local Saudi Arabia time. Moreover, the state must be signed by the private key of the services provider. The OpenID Connect Authentication request to be signed must be as follows:

https://iam.elm.sa/authservice/authorize?
scope=openid
&response_type= id_token
&response_mode=form_post
&client_id=<CLIENT_ID>
&redirect_uri=https://www.service_provider.com/callback
&nonce=GUID_RANDOM
&ui_locales=ar
&prompt=login
&max_age=timestamp

The state serves as a proof that the service provider has the private key. The state also serves as a reference that maps user requests from the Service Provider to responses from Elm. Therefore, the Service Provider must store the state, for example in a session management system, to link OpenID Connect responses to requests. The Service Provider must store the state in a hashed format (SHA256).

The URL https://iam.elm.sa/authservice/authorize is a production URL. For staging, please use the following URL replacing the query string accordingly.

https://iambeta.elm.sa/authservice/authorize

2) Authentication Response - from Elm to Service Provider

After user authentication is successfully completed Elm sends the following response to the Service Provider.

POST /callback HTTP/1.1
Host: https://iam.elm.sa
Content-Type: application/x-www-form-urlencoded
id_token=header.claims.signature
&state=<Hashed State>

The state is the hash of the request state using SHA256. The Service Provider must map the state in the request to the state in the response to serve the user and for verification. The id_token is a JWT token with three parts: header, claims, and signature. A sample of the

id_token is listed below. The Service Provider must validate the state and the id_token; The id_token validation must include:

- Header: the header will be {“typ”:”JWT”, “alg”:”RS256”} , RSA signature is used.
- Signature: the digital signature must be validated using Elm’s certificate. Furthermore, the configured certificate fields should be validated.
- Expiry: the JWT is not expired.
- Claims: the expected claims of the user.

{
"sub": "1010101010",
"acr": "2",
"nbf": 1517317320,
"iss": "https://www.iam.gov.sa/authservice",
"iat": 1517317470,
"exp": 1517317620,
"dob": "Wed Dec 19 03:00:00 AST 1979",
"aud": "https://www.serviceprovider.com/cb",
"jti": "e1db40ad-cae6-4f2e-929f-822c8b2a1e92",
"englishFamilyName": "Abu Saleh ",
"lang": "en",
"arabicFamilyName": "...",
"nationalityCode": "1",
"cardIssueDateGregorian": "Sat Oct 25 03:00:00 AST 2008",
"englishFirstName": "... ",
"gender": "Male",
"englishFatherName": "Faisal ",
"arabicNationality": "السعودية",
"dobHijri": "1400/01/30",
"arabicGrandFatherName": "- ",
"arabicName": " ...
"englishGrandFatherName": "- ",
"arabicFirstName": "...",
"preferredLang": “en”,
"cardIssueDateHijri": "1429/10/25",
"englishName": "...",

"nationality": "Saudi ",
"arabicFatherName": "... ",
"issueLocationAr": "جوازات القصيم ",
"issueLocationEn": "Riyadh Passports"
}

3.2 OpenID Connect Authentication: Logout Service

The Service Provider requests the logout service through IAM. Further details will be provided on demand.

4. IAM Service Provider Certificate Issuance Procedure

IAM Services uses mutual authentication through PKI based certificates for security; The Service Provider identifies Elm through digital signature and the Service Provider identifies Elm through digital signature.

The following steps describes how to generate a CSR that is required for IAM Service by Elm. The Service Provider must generate the CSR following the steps below and then share the CSR with Elm. The CSR generation steps are explained for Linux and Windows System, respectively. Some of these steps are based on NIC guides.

4.1 Linux: IAM Service Provider Certificate Issuance Procedure

Here are the Linux instructions using OpenSSL to generate a CSR for IAM Service Provider:

1. Generate a private key:
   - openssl genrsa -des3 -out iamtest.spname.key 2048

   enter the password and save it somewhere safe for later use
2. Generate an CSR:
   - openssl req -new -key iamtest.spname.key -out iamtest.spname.key.csr

   enter the certificate details. IMPORTANT: for Common Name, please enter the Reference Number that was given to you by Elm
3. openssl pkcs12 -export -out certificate.pfx -inkey iamtest.spname.key -in referenceNumber.cer -certfile InfraCAPP.cer

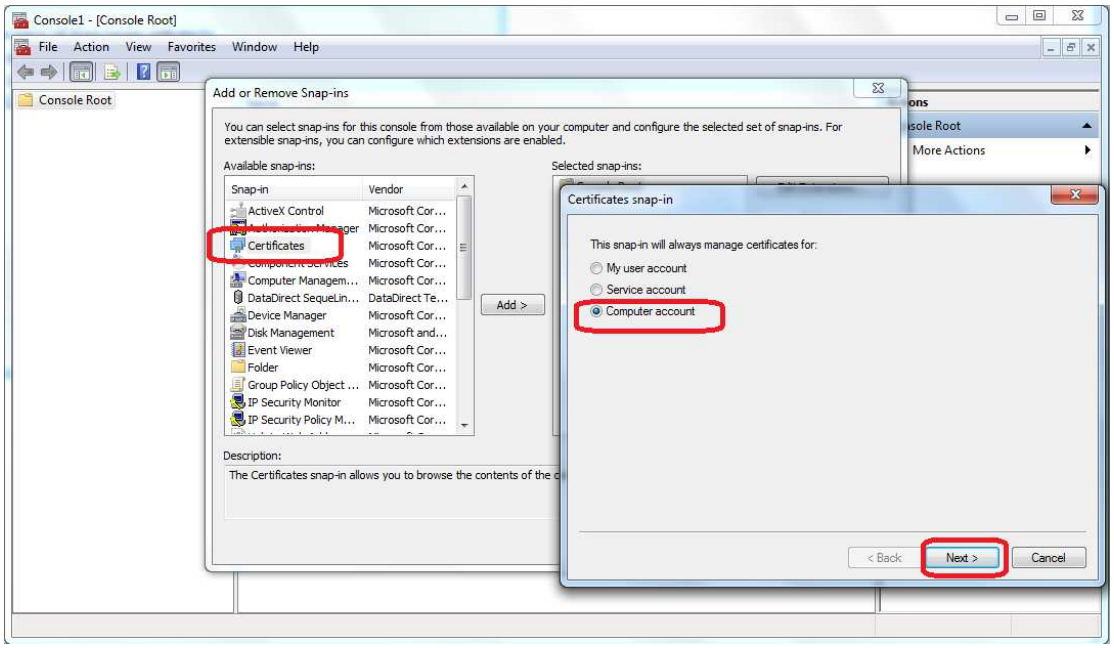
referenceNumber.cer is the certificated that is signed by NIC and InfraCAPP.cer is the intermediate NIC certificate. Both certificates referenceNumber.cer and InraCAPP.cer will be shared with you by Elm

The command above will generate a certificate.pfx file. This file contains the private and public key signed by NIC. Please use the private key of certificate.pfx to sign the OpenID Connect requests that are going out from your system to IAM.

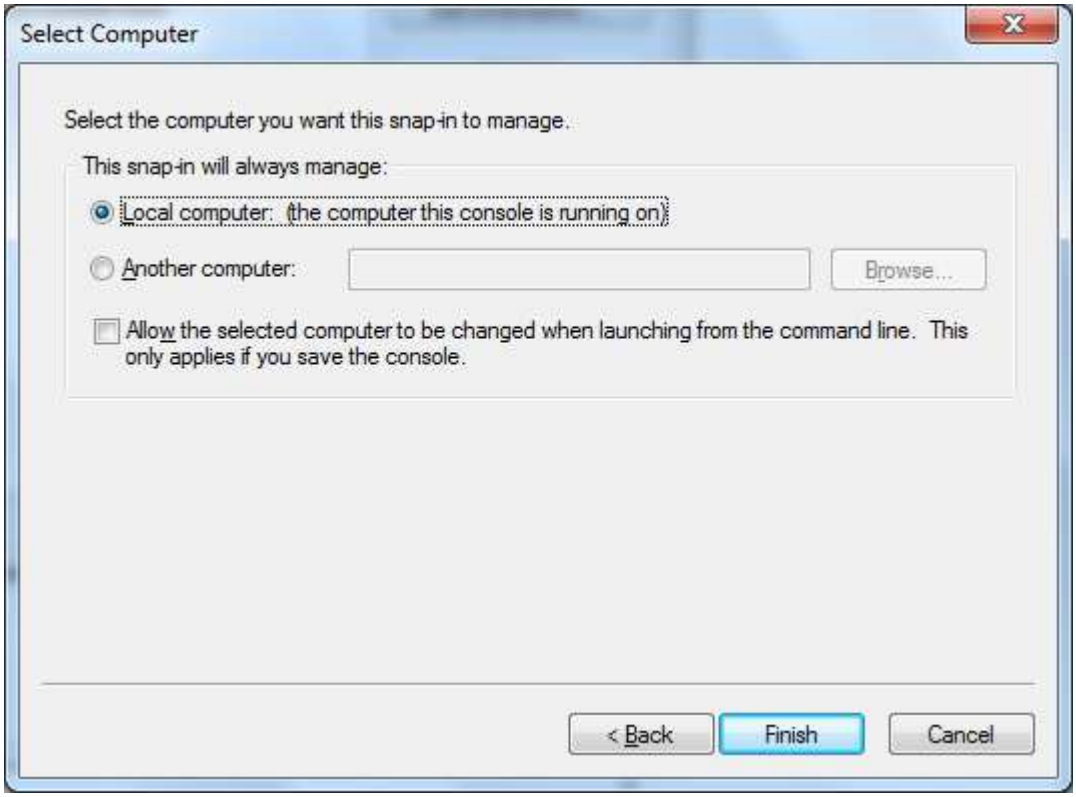
4.2 Windows: IAM Service Provider Certificate Issuance Procedure

4.2.1 Generate the Certificate Signing Request

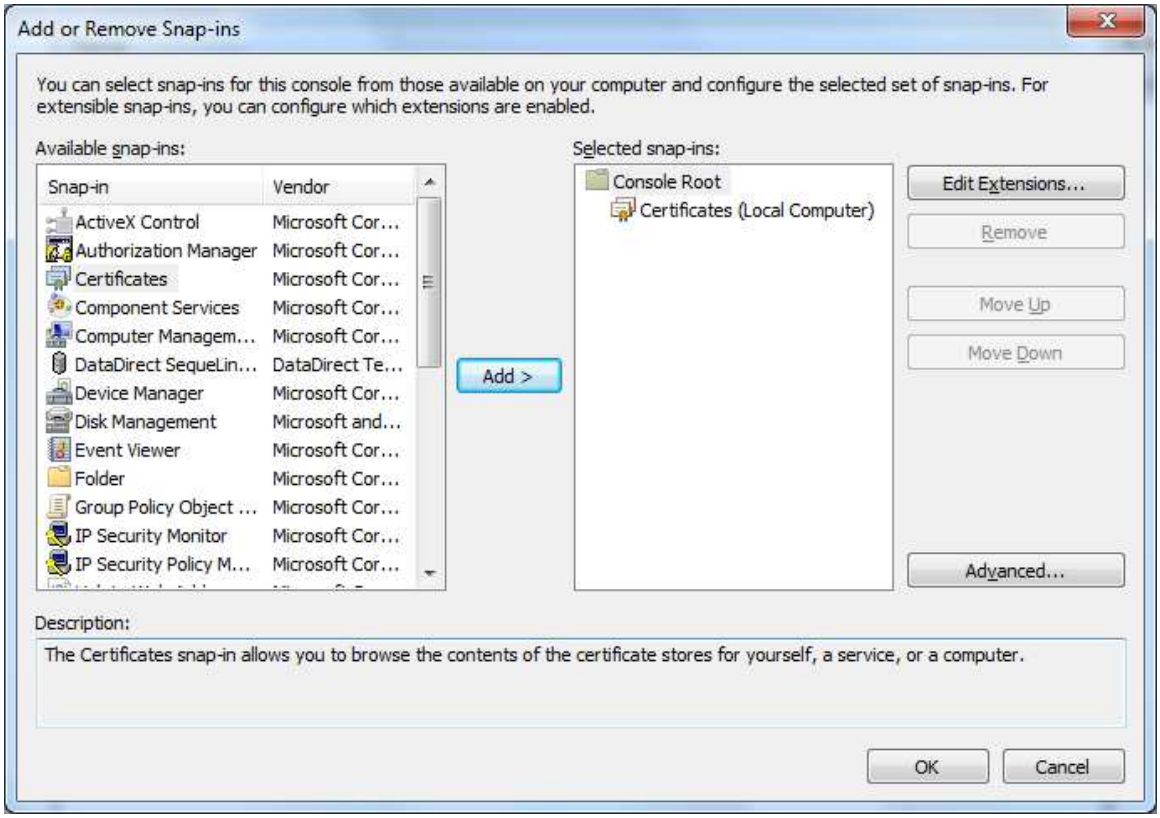
Type certmgr.msc in the search bar on Windows and run as administrator. If you do not see "Certificates – Local Machine" then type mmc in the search bar and run as administrator. Navigate to File > Add/Remove Snap-In; Select Certificates and click on Add button then select Computer Account and Click Next as shown in the Figure below.



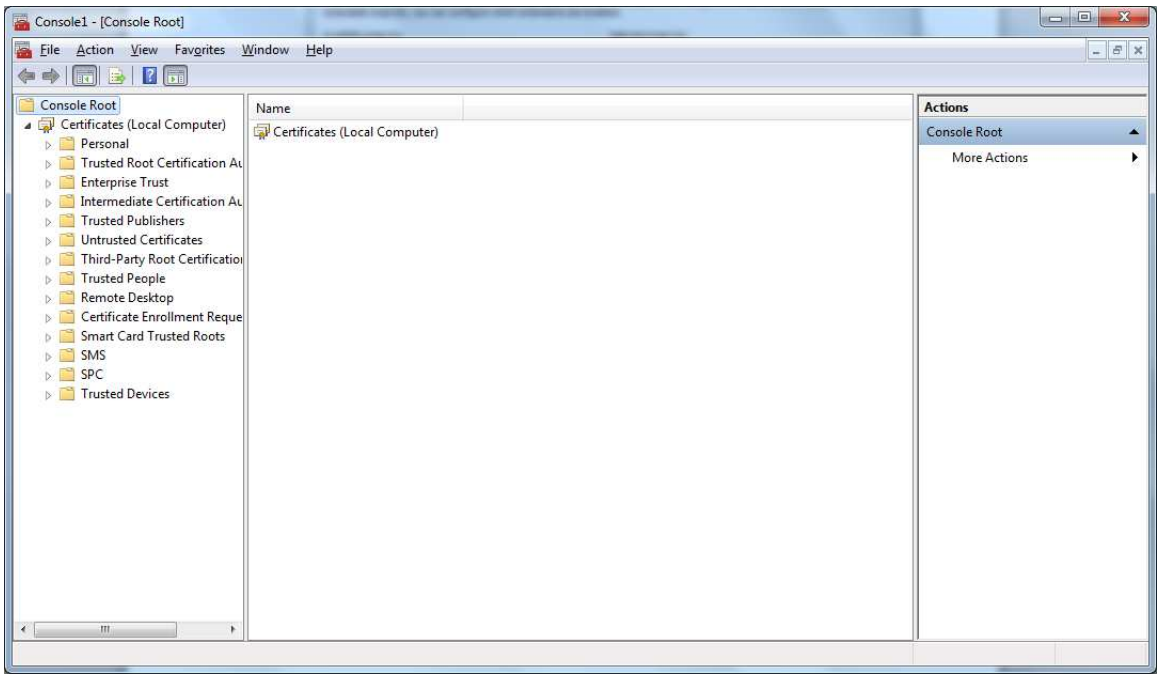
If you are asked to select a computer then choose "Local Computer" and click on Finish as shown below.



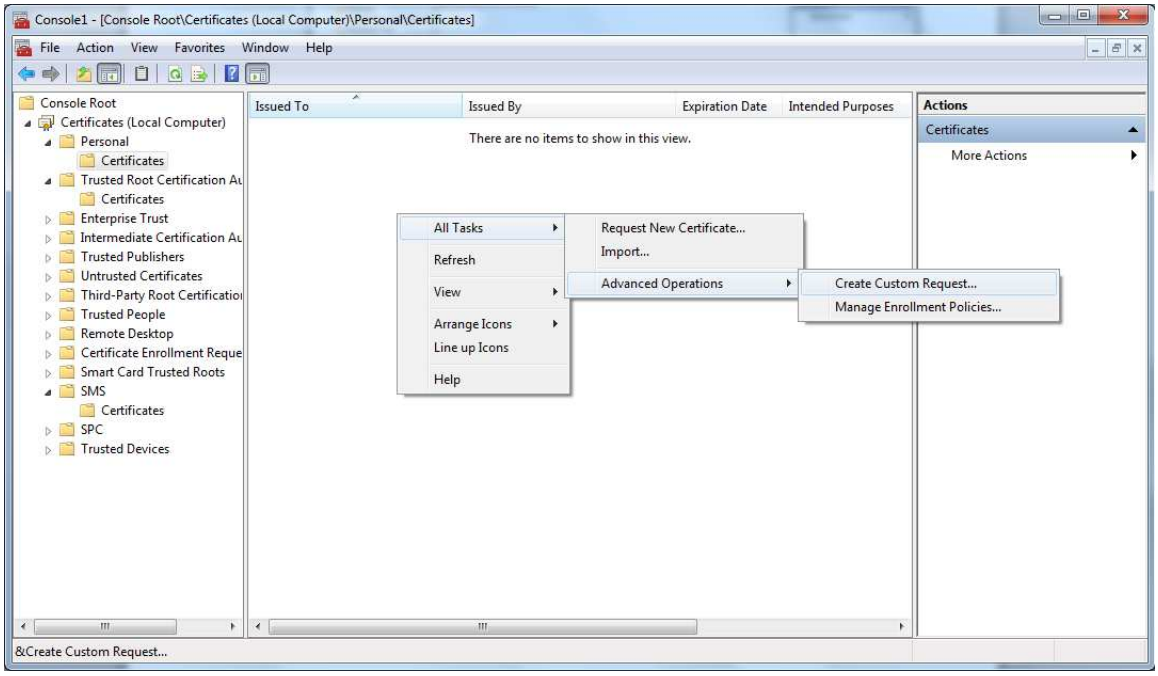
Confirm your changes by clicking Ok.



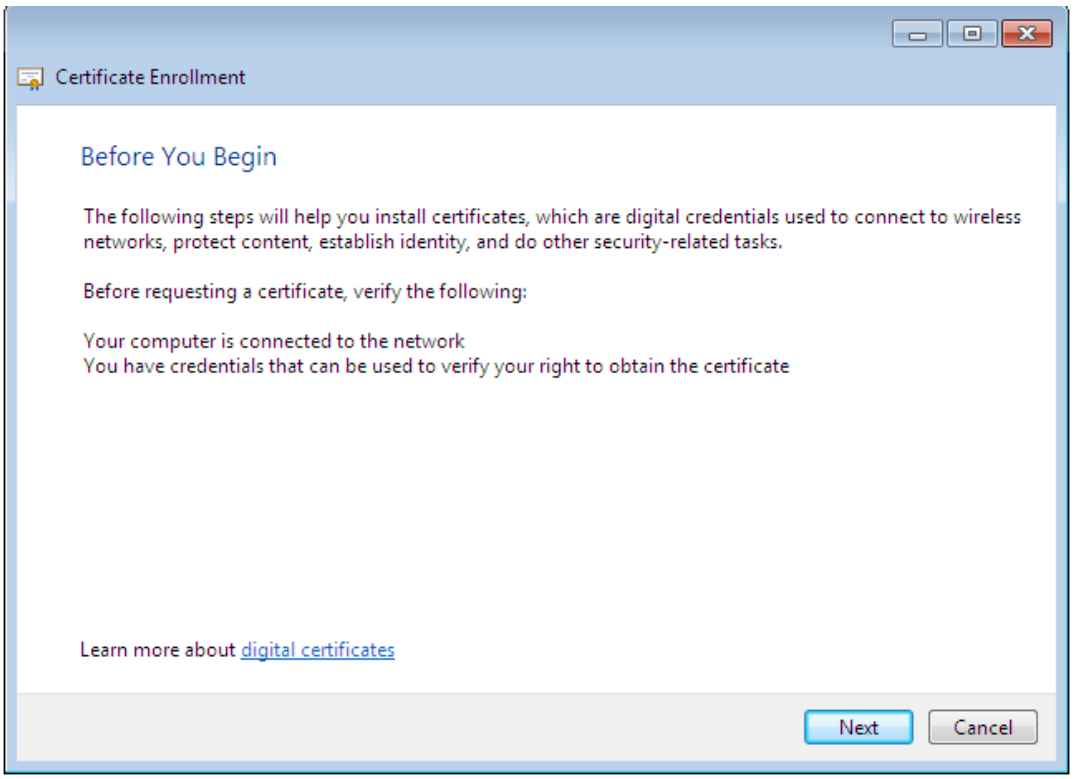
The Certificate Manager will appear as shown below.



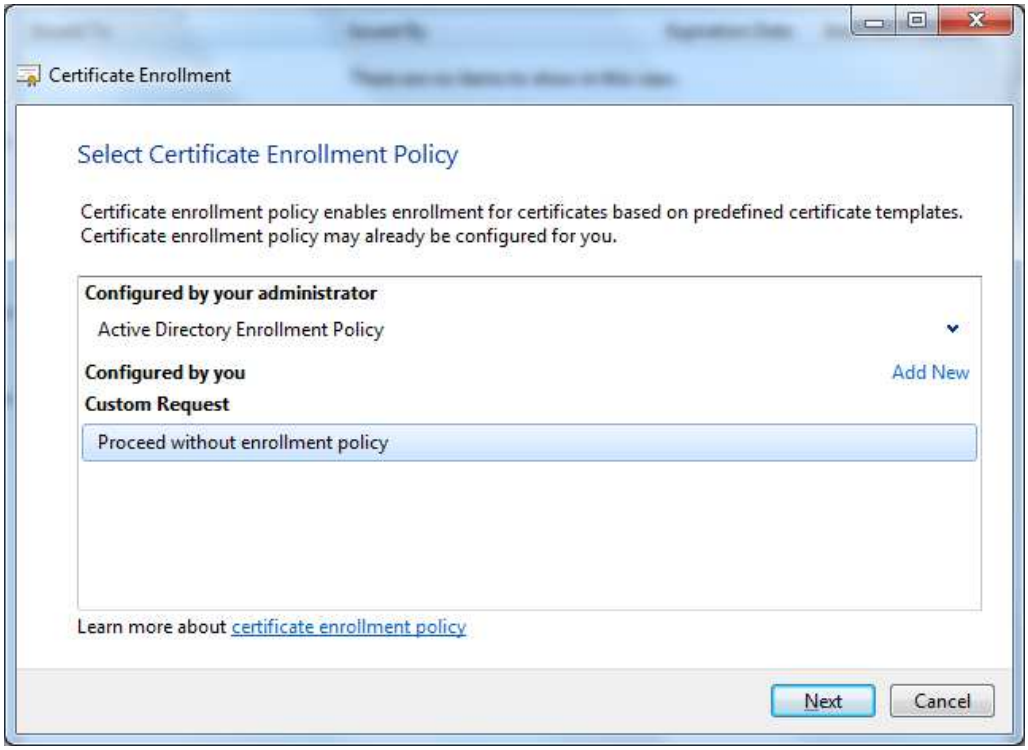
Navigate to Personal and then right click on the right panel, select All Tasks > Advanced Operations > Create Custom Request as show below.



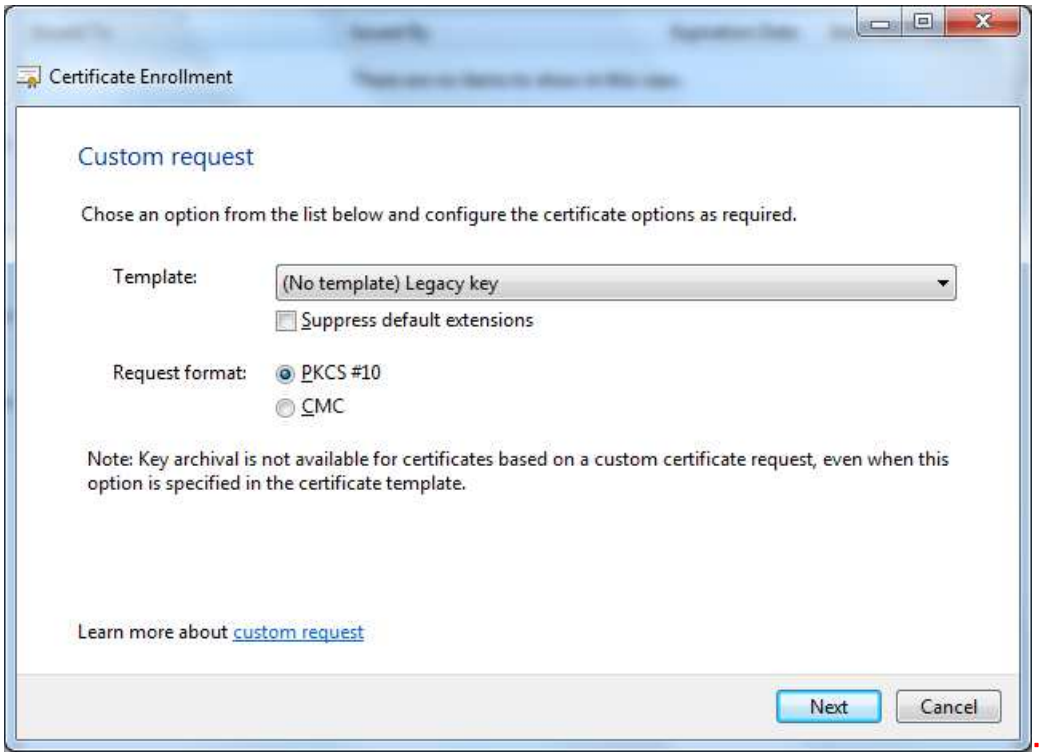
Certificate Enrollment will be prompted as shown below. Click on Next.



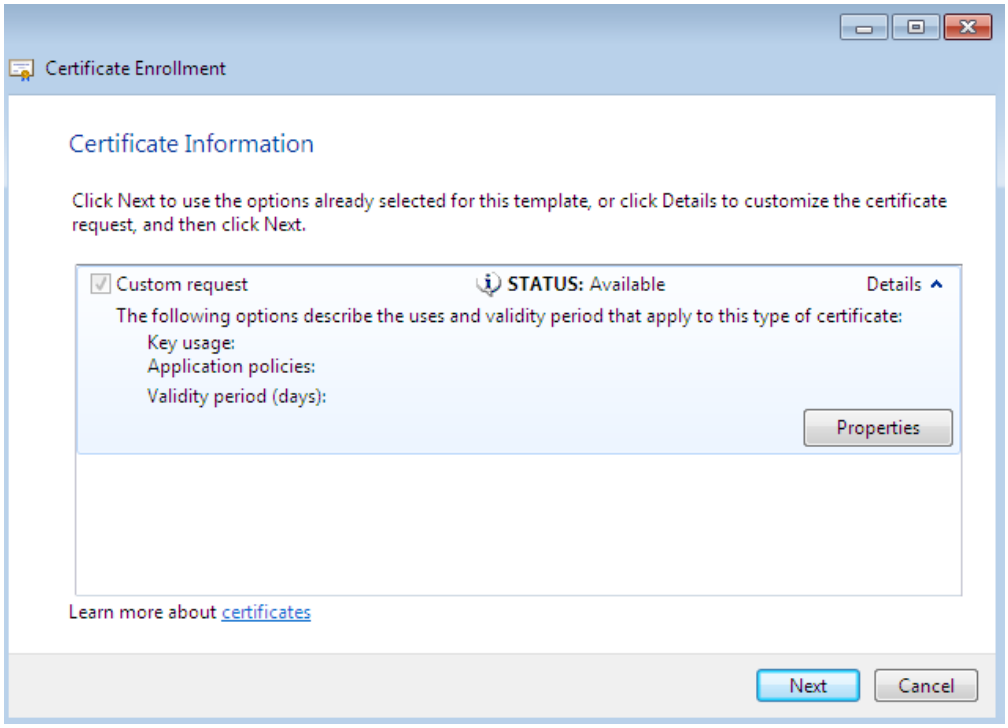
Choose the “Proceed without enrollment policy” then click on Next as shown below.



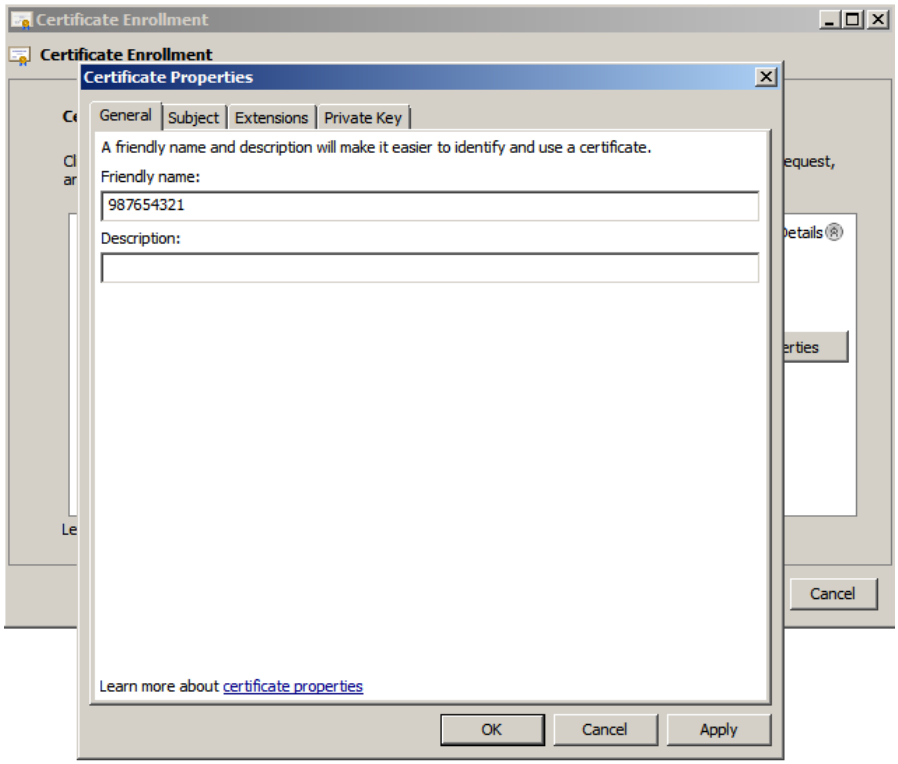
Afterwards, choose “Legacy Key” as a custom request template and leave other options as is as shown below. **Note: Do not select CNG key**



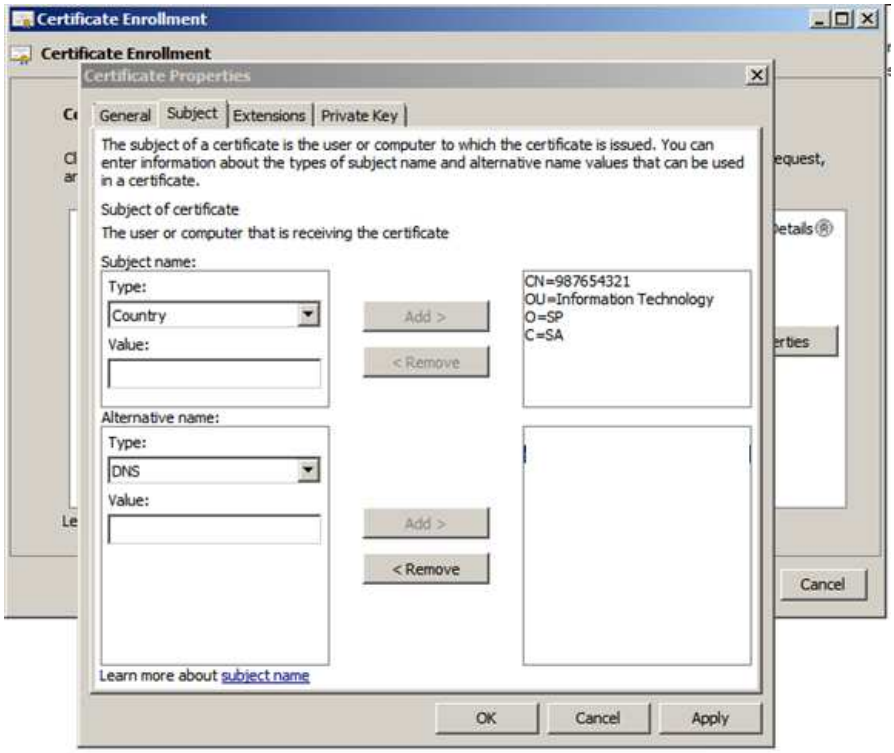
Click on Properties of the Custom request - see below.



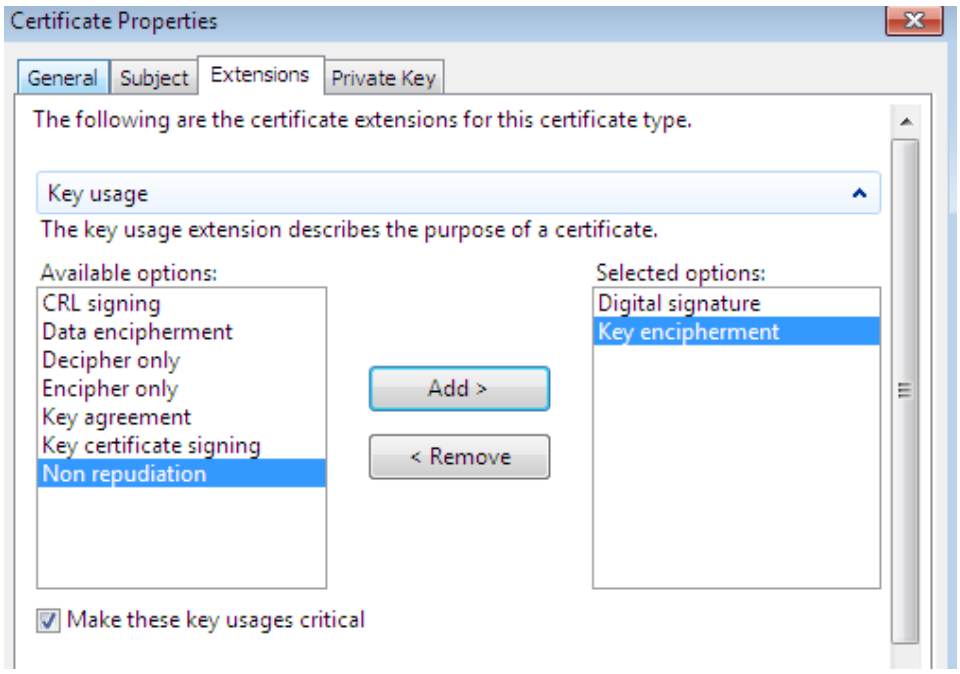
Select the General tab and type in the Reference Number provided by Elm team.



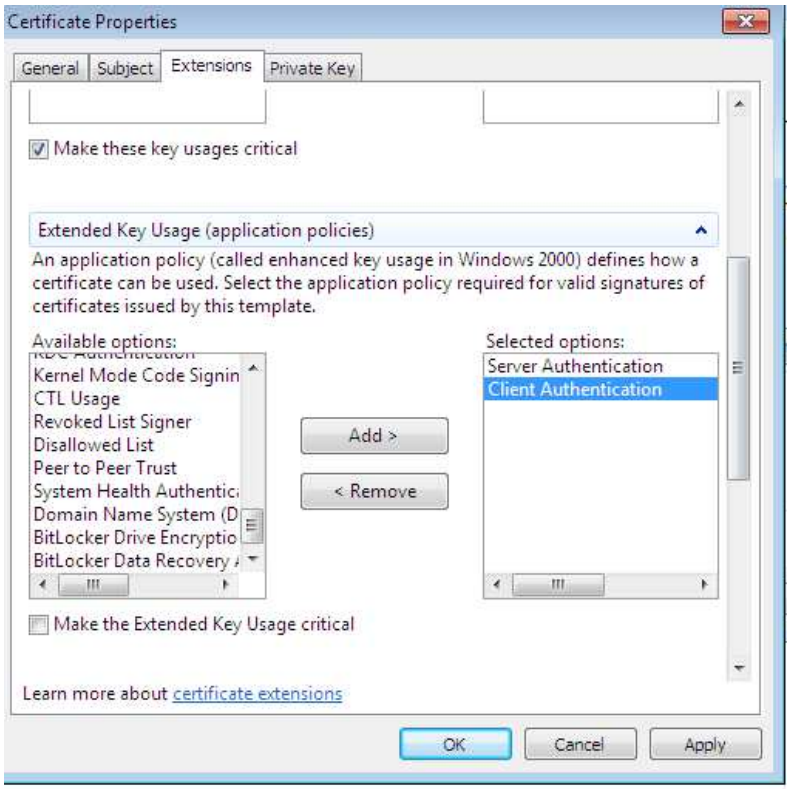
Navigate to the Subject tab and fill in the following information (the CN=<Reference Number> is enough).



Navigate to the Extension tab and select Key Usage. Under Key Usage, select “Digital Signature” and “Key encipherment” and add them using Add button as shown below.



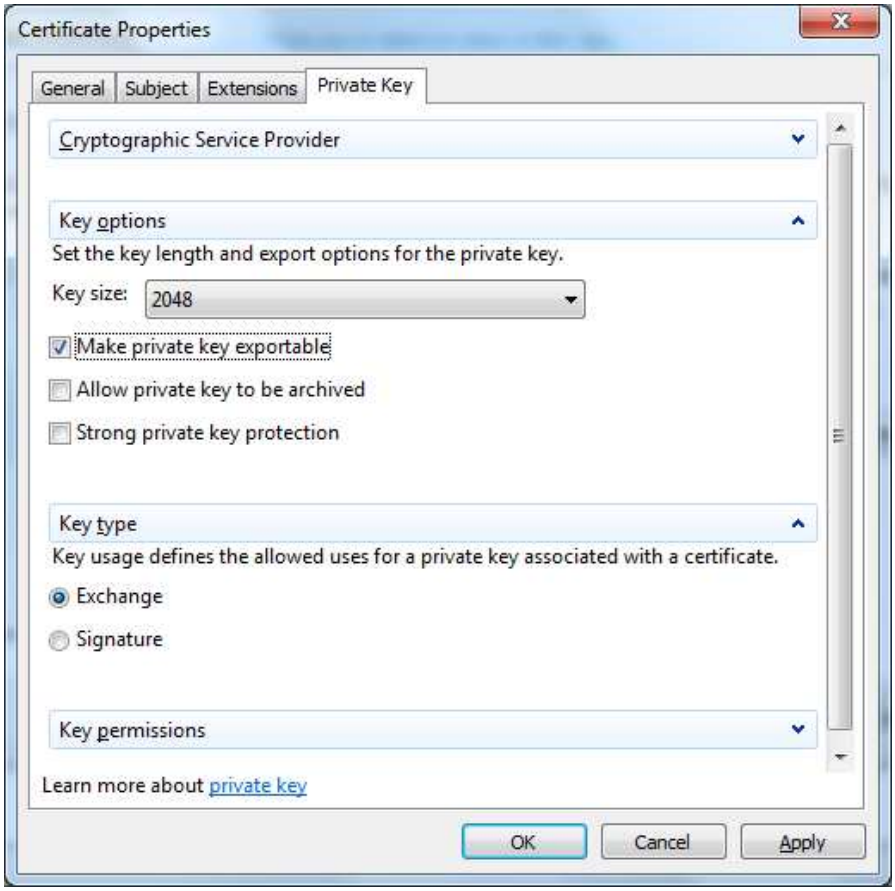
Under Extensions tab select Extended Key Usage. Under Extended Key Usage select “Server Authentication” and “Client Authentication” using the Add button.



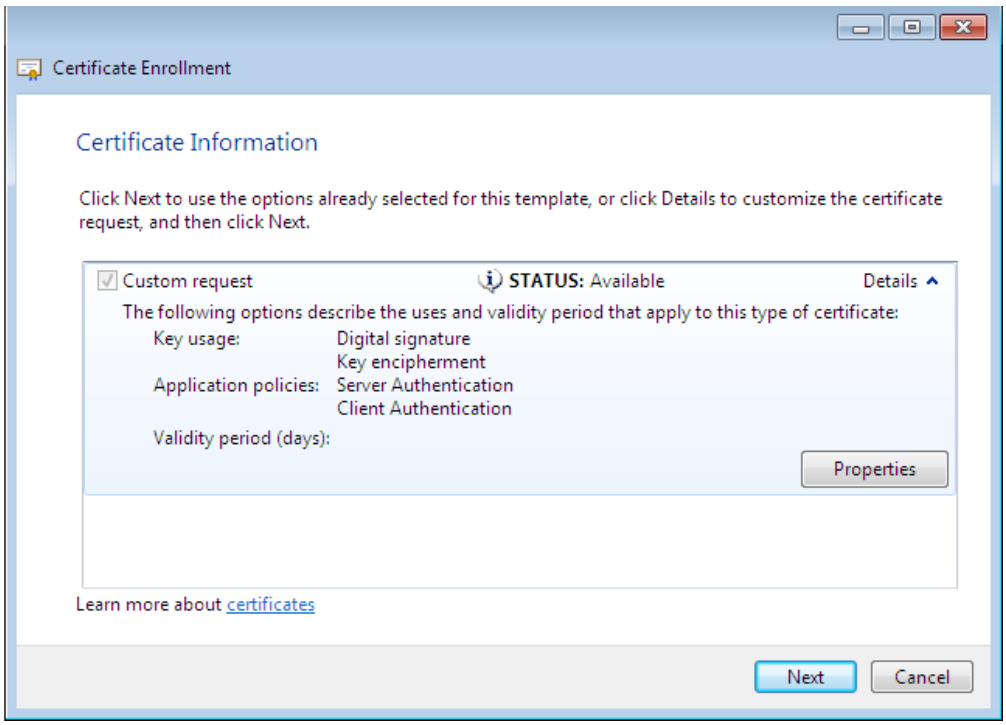
Navigate to the Private Key tab and choose the key options and hashing algorithm as shown below. Ensure that:

- Key size is 2048.
- “Make private key exportable” is checked.
- Key type is set to Exchange.

Note that when Key type is set to Exchange the key size will automatically turn into 1024. Please change the key size back to 2048.



Confirm and verify the certificate properties and click on Next.



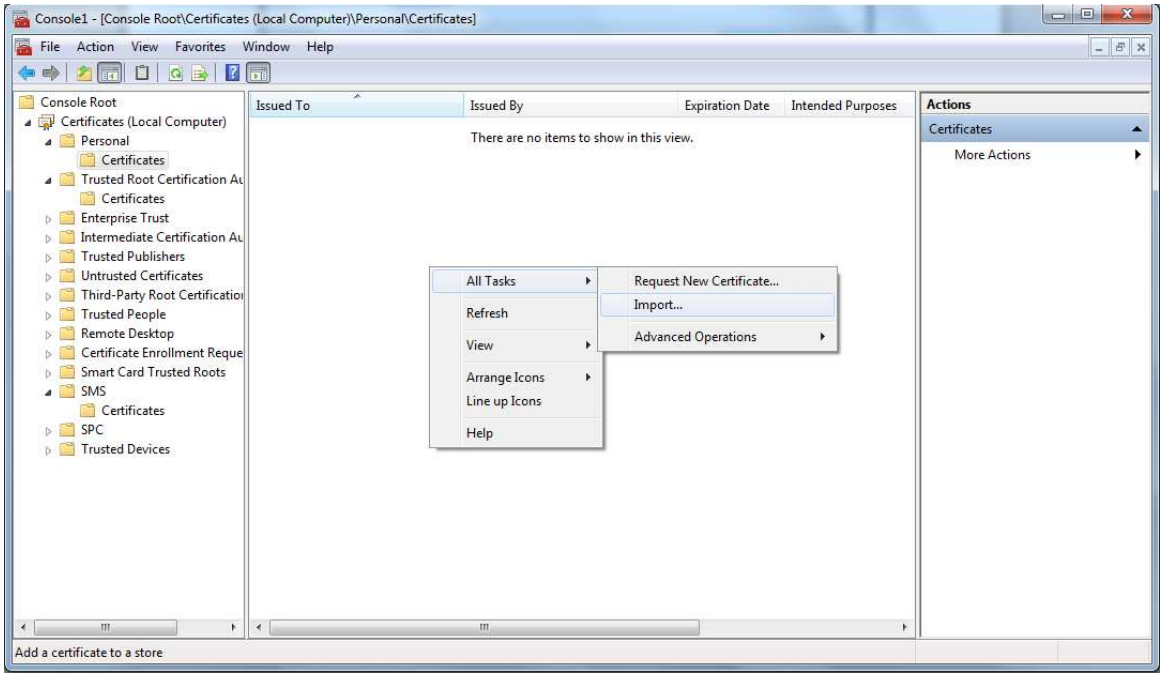
A prompt will be displayed to save the request; Save the request somewhere safe in your desk as Base64 and name it <Common Name>.csr (i.e. <Reference Number>.csr).

Send the generated CSR to Elm team.

4.2 Import the signed Certificate from Elm

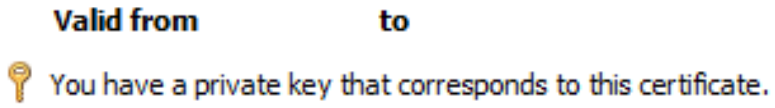
The CSR will be signed by IAM team and Elm will send the certificate bac to you. To import the certificate, open up the Certificate Manager with admin privilege under Local Computer).

Navigate to Certificate > Personal, then right click on the right panel and click on All tasks > Import as shown below.



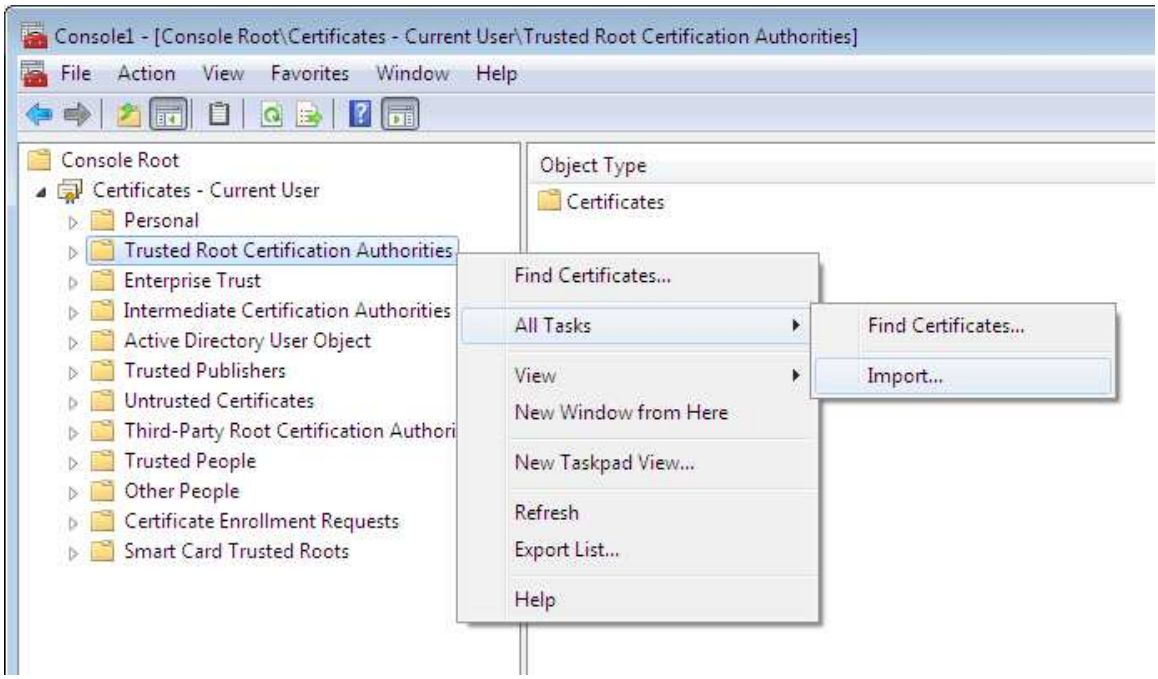
Selected the certificate that you have received from Elm and validate it to import the certificate. After the certificate is imported make sure that a private key is associated with the certificate. The certificate icon should look similar to what follows.

Double click on the certificate icon and you should see the following.



4.3 Import the MOI Root CA V2 or PP

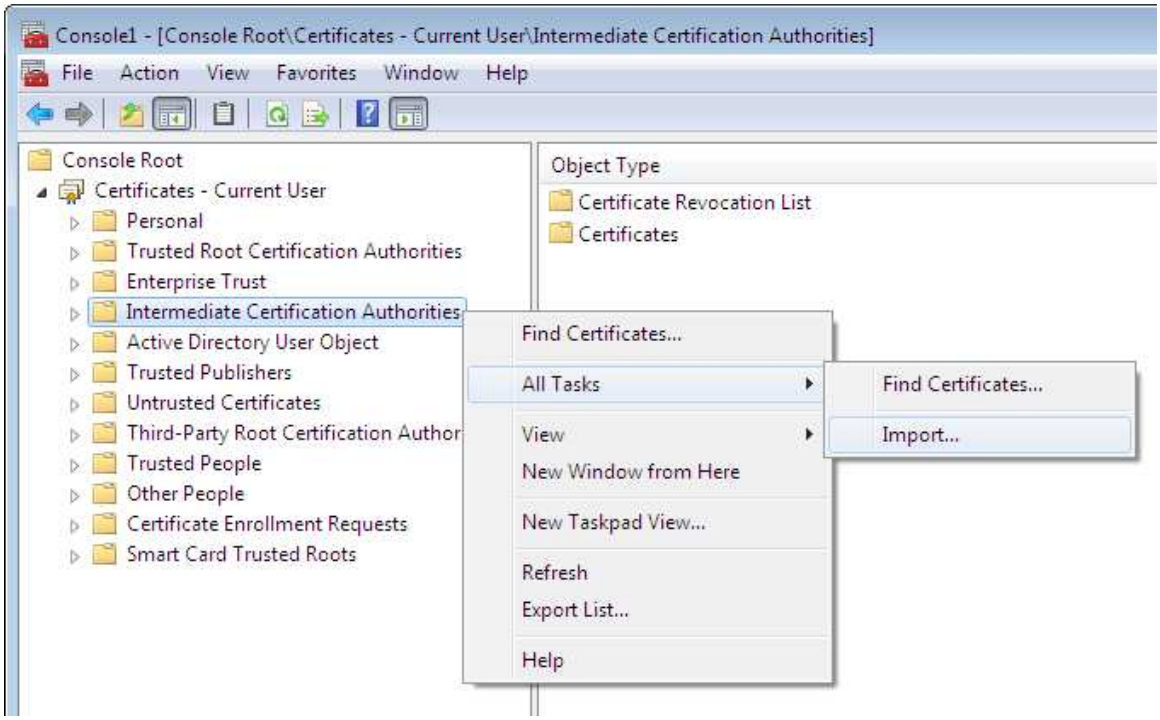
Elm team will share the Root certificate with you. From the Certificate Manager navigate to Certificates > Trusted Root Certification Authorities. Right click on Trusted Root Certification Authorities and click on All Tasks > Import then Import the MOI Root CA V2 into the Trusted Root Certification Authorities as shown below.



You will be asked to select the Root that you have received from Elm.

4.4 Import the Infra CA V2 or PP

Elm team will share the Infra certificate with you. From the Certificate Manager navigate to Certificates > Intermediate Certification Authorities. Right click on Intermediate Certification Authorities and click on All Tasks > Import then Import the Infra CA V2 into the Intermediate Certification Authorities as shown below.



You will be asked to select the Infra certificate that you have received from Elm.

Instructions for generating a CSR for the NIC

You may use the following commands to generate a CSR for the NIC on Linux. **Be sure to change the CN** to match what the NIC gives you (replace "**12345678**" below when generating the file "nic-csr.conf" – the very first command).

```
linuxprompt$ echo -e "[ req ]\nprompt = no\ndistinguished_name = dn\nreq_extensions = reqexts\n\n[ dn ]\nCN =
12345678\n\n[ reqexts ]\nkeyUsage = digitalSignature, keyEncipherment\nextendedKeyUsage = clientAuth" > nic-
csr.conf

linuxprompt$ more nic-csr.conf
[ req ]
prompt = no
distinguished_name = dn
req_extensions = reqexts

[ dn ]
CN = 12345678

[ reqexts ]
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth

linuxprompt$ openssl req -config nic-csr.conf -new -newkey rsa:2048 -nodes -keyout privatekey.key -out
certreq.csr
Generating a RSA private key
..............................+++++
..................+++++
writing new private key to 'privatekey.key'
-----

linuxprompt$ ls -l privatekey.key certreq.csr
-rw-r--r-- 1 root root  960 Mar 20 14:01 certreq.csr
-rw-r--r-- 1 root root 1708 Mar 20 14:01 privatekey.key

linuxprompt$ file privatekey.key certreq.csr
privatekey.key: ASCII text
certreq.csr:    PEM certificate request

linuxprompt$ openssl req -in certreq.csr -text
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: CN=12345678
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c1:a5:13:0e:38:7b:7c:ac:8e:8e:ee:23:b2:01:
                    64:e0:4d:ff:78:b4:f3:b0:24:35:6c:d5:74:9b:79:
                    eb:95:98:7b:26:d4:1d:5a:e3:66:32:8c:e0:7c:18:
                    7a:d2:13:ac:ef:a3:b9:a0:94:04:b1:f2:92:46:1a:
                    d2:9e:d7:fc:c3:95:bd:e1:e0:26:db:8f:06:2f:26:
                    b2:38:c3:e9:55:d2:bd:d7:d3:58:fd:b7:cd:10:dc:
                    8a:b1:da:34:04:27:cc:d7:47:35:4c:1a:f4:fd:03:
                    3d:20:4d:c2:1d:1e:55:0f:b8:b6:d4:ce:50:cf:37:
                    ad:74:07:bc:45:a6:44:81:4a:36:0d:4e:3c:4a:c7:
                    0d:c7:2c:3d:a0:b3:1c:c6:41:da:bd:4a:99:73:04:
                    63:89:2f:e0:d3:62:04:70:73:82:e5:a0:dd:70:41:
                    5f:f7:84:25:68:d1:88:2b:13:ff:8b:d9:c1:81:14:
                    61:81:21:f2:81:f2:8b:a5:75:ea:86:1b:5e:3a:55:
                    bb:a7:fd:35:26:b5:d2:fb:80:cb:af:c6:a0:2f:05:
                    4f:b9:27:13:9c:24:b4:dd:0c:6d:dd:d9:de:32:27:
                    a3:00:10:ef:97:2d:7b:e3:f7:d9:71:b3:f7:82:89:
                    31:46:c9:d6:7c:e6:82:65:ea:cc:8b:ba:64:58:c2:
                    26:9d
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
```

```
                TLS Web Client Authentication
    Signature Algorithm: sha256WithRSAEncryption
         69:ef:6f:18:32:ac:a3:bc:f5:80:51:52:4b:cb:9b:59:c8:dc:
         21:24:ca:2f:1d:09:ac:47:a3:41:87:67:b9:48:97:c3:5f:14:
         85:71:69:f9:76:c0:3d:7f:a0:dd:67:0f:09:2c:52:ea:82:e0:
         ef:42:ff:fa:1e:ef:6d:8e:66:d6:b3:a1:d7:59:d3:a2:bf:8d:
         6f:f6:b6:14:a7:ed:69:2c:ef:7f:4c:a5:45:74:b2:9a:87:0e:
         75:d7:a9:76:cb:30:2e:97:50:d1:a7:2b:38:ed:8e:8b:c4:fd:
         d3:bd:ef:0b:d8:01:99:be:86:e2:cc:ea:36:60:0c:b9:cb:8a:
         75:e0:23:1a:1f:42:a2:70:05:25:26:62:81:d1:c6:e6:d3:b2:
         60:20:07:f1:7e:de:69:ff:91:76:04:9d:81:8e:7a:3a:d0:07:
         62:6b:11:94:b0:09:9b:dd:cf:fa:0e:38:99:c2:bd:a1:9b:fa:
         d7:80:de:cd:c4:09:65:fb:4a:4e:cb:29:4a:89:cc:ee:7e:84:
         1a:f8:30:6f:9b:36:c3:80:b1:29:fa:f7:64:85:c1:9a:4a:d2:
         a6:db:dc:cc:22:77:0f:98:5b:8f:6e:57:b5:2d:4a:4b:11:3f:
         2a:86:18:bf:b1:7b:36:cf:73:ed:20:73:0c:c8:3e:0c:06:46:
         55:6e:b1:1a
-----BEGIN CERTIFICATE REQUEST-----
MIICizCCAXMCAQAwEzERMA8GA1UEAwwIMjUyNjA3OTIwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQDBpRMOOHt8rI6O7iOyAWTgTf94tPOwJDVs1XSbeeuV
mHsm1B1a42YyjOB8GHrSE6zvo7mglASx8pJGGtKe1/zDlb3h4CbbjwYvJrI4w+lV
0r3X01j9t80Q3Iqx2jQEJ8zXRzVMGvT9Az0gTcIdHlUPuLbUzlDPN610B7xFpkSB
SjYNTjxKxw3HLD2gsxzGQdq9SplzBGOJL+DTYgRwc4LloN1wQV/3hCVo0YgrE/+L
2cGBFGGBIfKB8ouldeqGG146Vbun/TUmtdL7gMuvxqAvBU+5JxOcJLTdDG3d2d4y
J6MAEO+XLXvj99lxs/eCiTFGydZ85oJl6syLumRYwiadAgMBAAGgMzAxBgkqhkiG
9w0BCQ4xJDAiMAsGA1UdDwQEAwIFoDATBgNVHSUEDDAKBggrBgEFBQcDAjANBgkq
hkiG9w0BAQsFAAOCAQEAae9vGDKso7z1gFFSS8ubWcjcISTKLx0JrEejQYdnuUiX
w18UhXFp+XbAPX+g3WcPCSxS6oLg70L/+h7vbY5m1rOh11nTor+Nb/a2FKftaSzv
f0ylRXSymocOddepdsswLpdQ0acrOO2Oi8T9073vC9gBmb6G4szqNmAMucuKdeAj
Gh9ConAFJSZigdHG5tOyYCAH8X7eaf+RdgSdgY56OtAHYmsRlLAJm93P+g44mcK9
oZv614DezcQJZftKTsspSonM7n6EGvgwb5s2w4CxKfr3ZIXBmkrSptvczCJ3D5hb
j25XtS1KSxE/KoYYv7F7Ns9z7SBzDMg+DAZGVW6xGg==
-----END CERTIFICATE REQUEST-----
```