MyCodeSigner Thales Integration

Code Signing Solution Integration Guide

Document Number: MyCodesigner-Integration-Guide-v1

Release Date: February 2020

Contents

1. Preface

This document is intended to customers for MyCodesigner application integration with SafeNet Network HSM, and also covers the necessary information to install Luna Client, configure and integrate MyCodesigner application with SafeNet Network HSM.

1.1 Scope

This document outlines the steps to integrate MyCodesigner application with SafeNet HSM. SafeNet HSM is used to secure the keys for code signing.

1.2 Gemalto Rebranding

In early 2015, Gemalto completed its acquisition of SafeNet, Inc. As part of the process of rationalizing the product portfolios between the two organizations, the Luna name has been removed from the SafeNet HSM product line, with the SafeNet name being retained. As a result, the product names for SafeNet HSMs have changed as follows:

Old product name	New product name
Luna SA HSM	SafeNet Network HSM
Luna PCI-E HSM	SafeNet PCI-E HSM
Luna G5 HSM	SafeNet USB HSM
Luna Client	SafeNet HSM Client

NOTE: These branding changes apply to the documentation only. The SafeNet HSM software and utilities continue to use the old names.

1.3 Support Contacts

If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or Gemalto Customer Support. Gemalto Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Gemalto and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact Method	Contact Information
Address	Gemalto 4690 Millennium Drive

	Belcamp, Maryland 21017, USA	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Technical Support Customer Portal	https://serviceportal.safenet-inc.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.	
Encryption Consulting Support Contact Details	Fazel Ahsan, Cyber Security Consultant, Encryption Consulting LLC, email: fazelahsan@encryptionconsulting.com, Tel: +1 469 815 4136, 130 N Preston Rd, Prosper TX 75078.	
	Puneet Singh, President, Encryption Consulting LLC, email: puneetsingh@encryptionconsulting.com, Tel: +1 469 815 4136, Cell: +1 469 400 7592, 130 N Preston Rd, Prosper TX 75078.	
	Prabal Kanti Sarkar, Software Engineer, Encryption Consulting LLC, email: Prabal@encryptionconsulting.com, Tel: +1 469 815 4136, 130 N Preston Rd, Prosper TX 75078.	

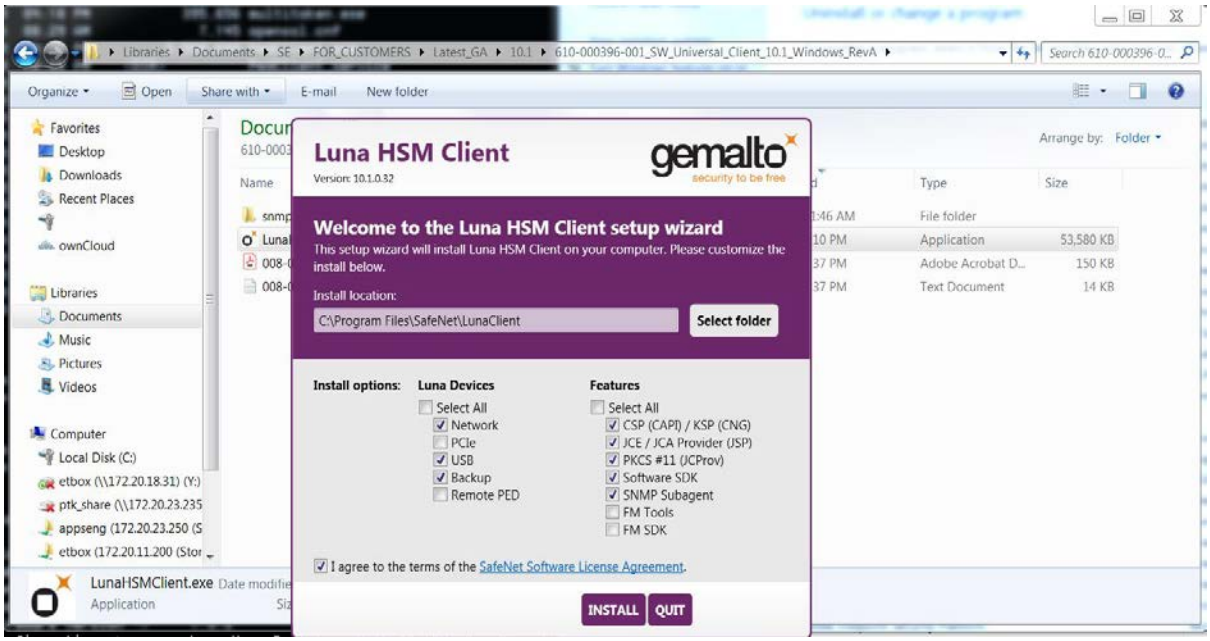
2. THALES – ELAB SETUP

- The SafeNet eLab environment is provided for the use of Thales customers and partners so that you can become familiar with the operation and functionality of SafeNet Luna Network HSM, with respect to testing application connectivity, development and integration. Since the eLab is a shared environment, you will not be able to experience the complete HSM appliance administration and management interface and capabilities, as this is being managed by the Thales eLab team, hence you have limited access to the roles described below, and you won't be allowed to run certain commands like 'tamper/decommission the HSM'.
- Feel free to test the your assigned HSM partition as you like, however, please be aware that at any time there may be more than one customer evaluating the appliance via the eLab units.
- The eLab email address is not for technical support. If you have any questions regarding the operation or capabilities of the network HSM product please contact your local Thales Systems/Sales Engineer.
- Questions specific to eLab status or feedback on the eLab program can be sent to eLab.request@gemalto.com.
- The purpose of this document is to setup and configure an HSM client to establish a secure connection with the HSM server running in the eLab environment.

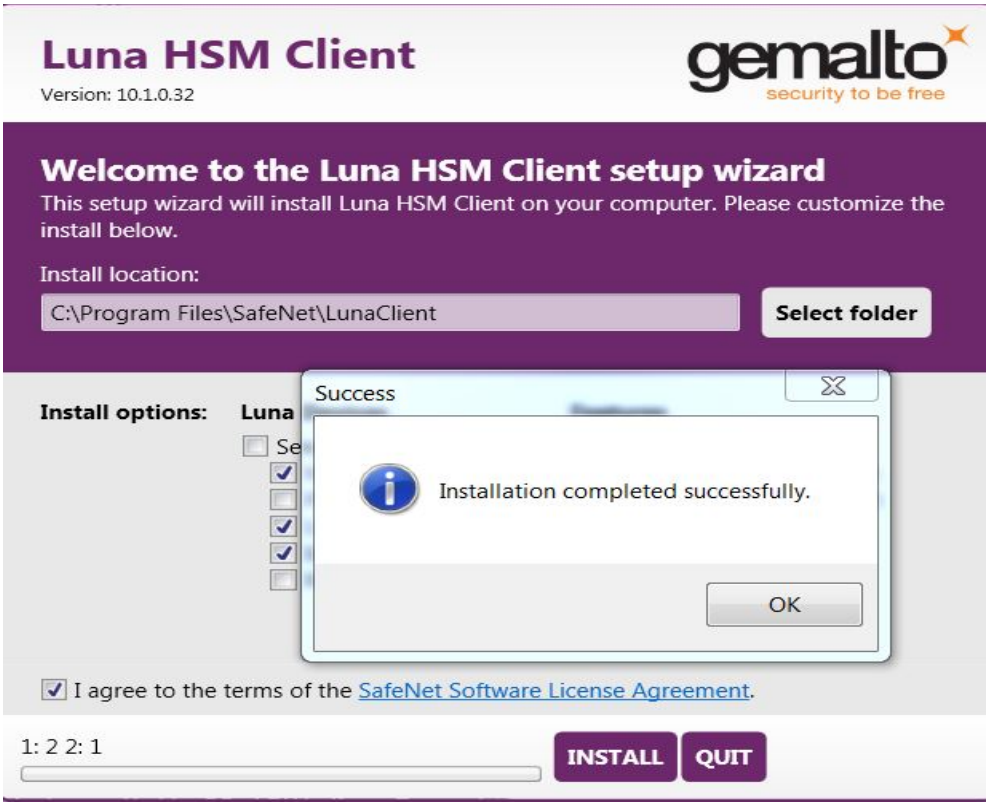
3. HSM CLIENT SETUP AND CONFIGURATION

3.1 Download and Install the Luna HSM 10.1 Client (i.e. SDK)

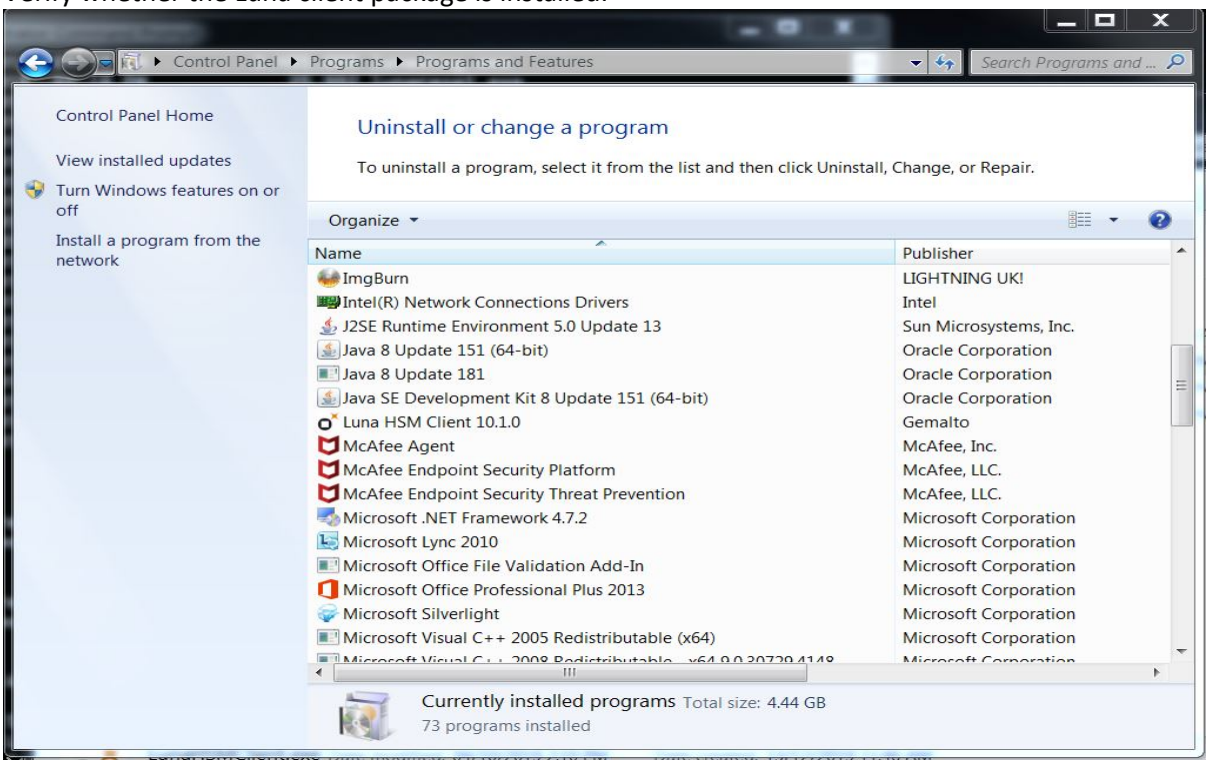
- Navigate to the folder where the Luna HSM 10.1 client was downloaded, extract and right click the LunaHSMClient.exe application and select Run as administrator.



When you see the following on your screen, the installation is complete.



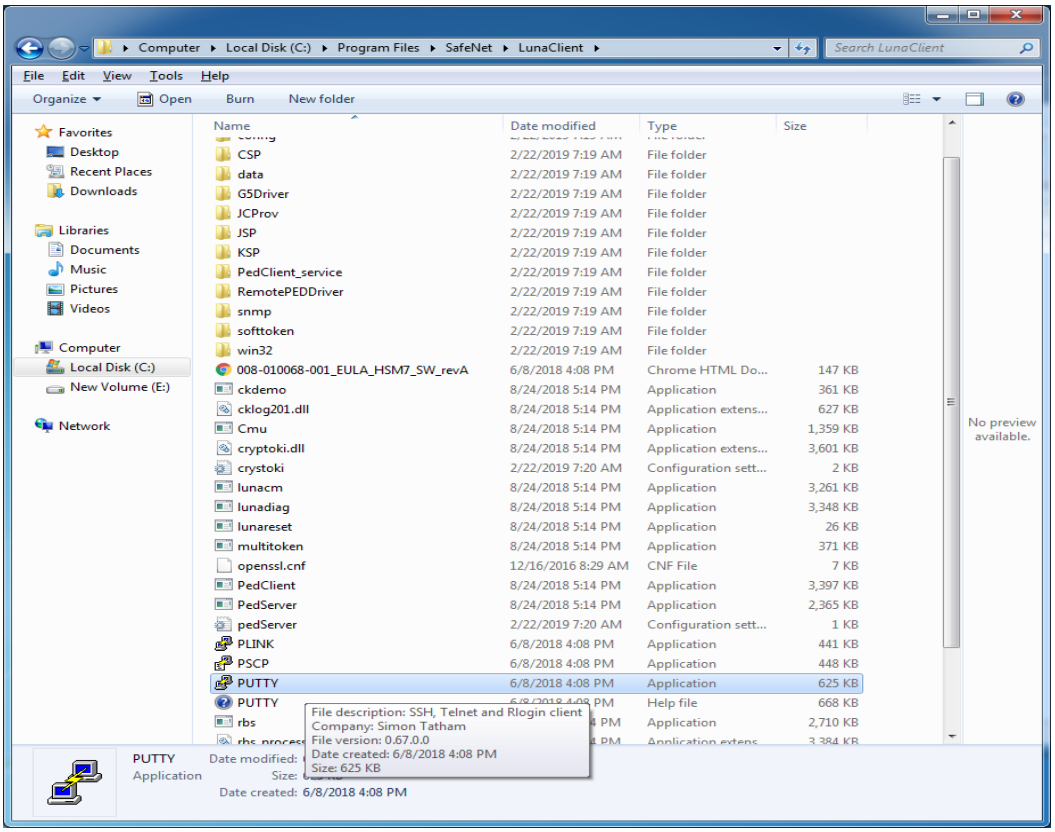
Verify whether the Luna client package is installed.



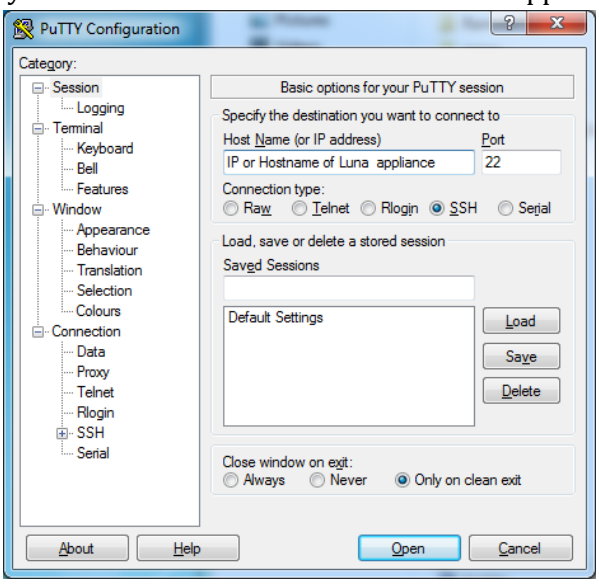
3.2 SSH into the eLab HSM appliance

- Check status of the HSM
- Check the state of the HSM partition assigned to you.

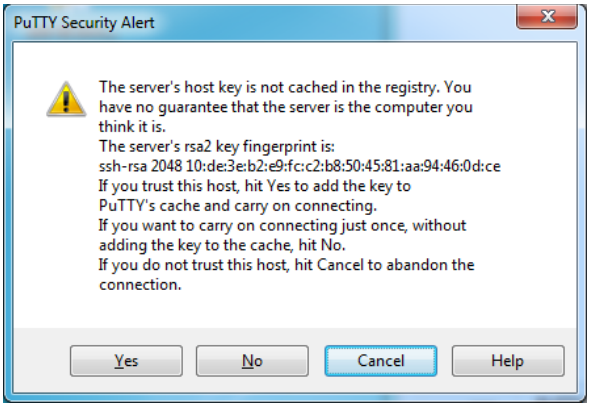
- Browse to the Luna client install directory



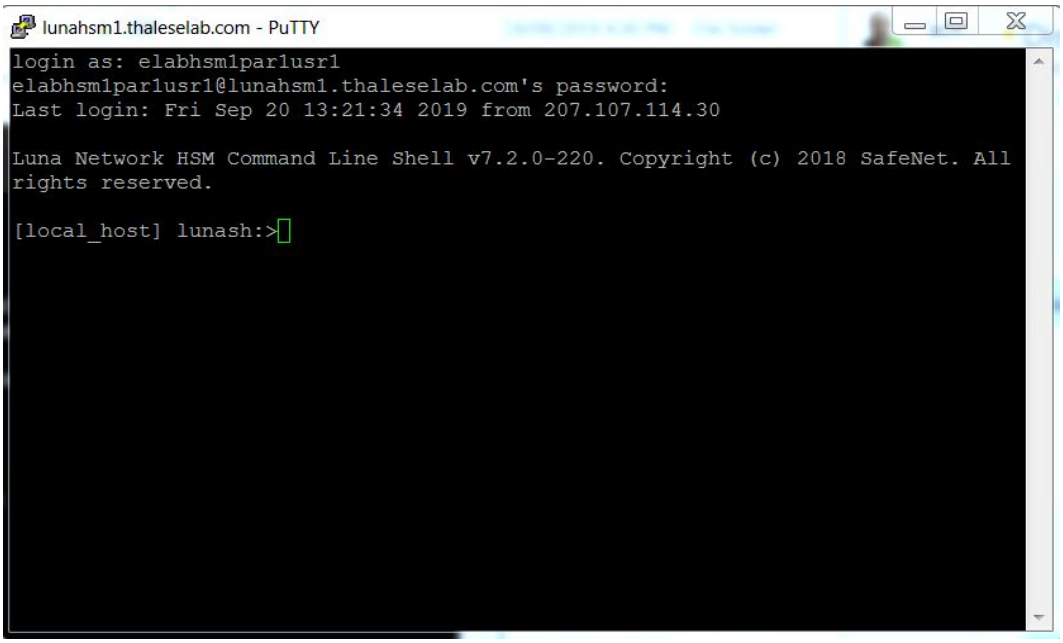
- Launch 'Putty'. Enter the IP or Hostname of the Luna appliance, click 'open'



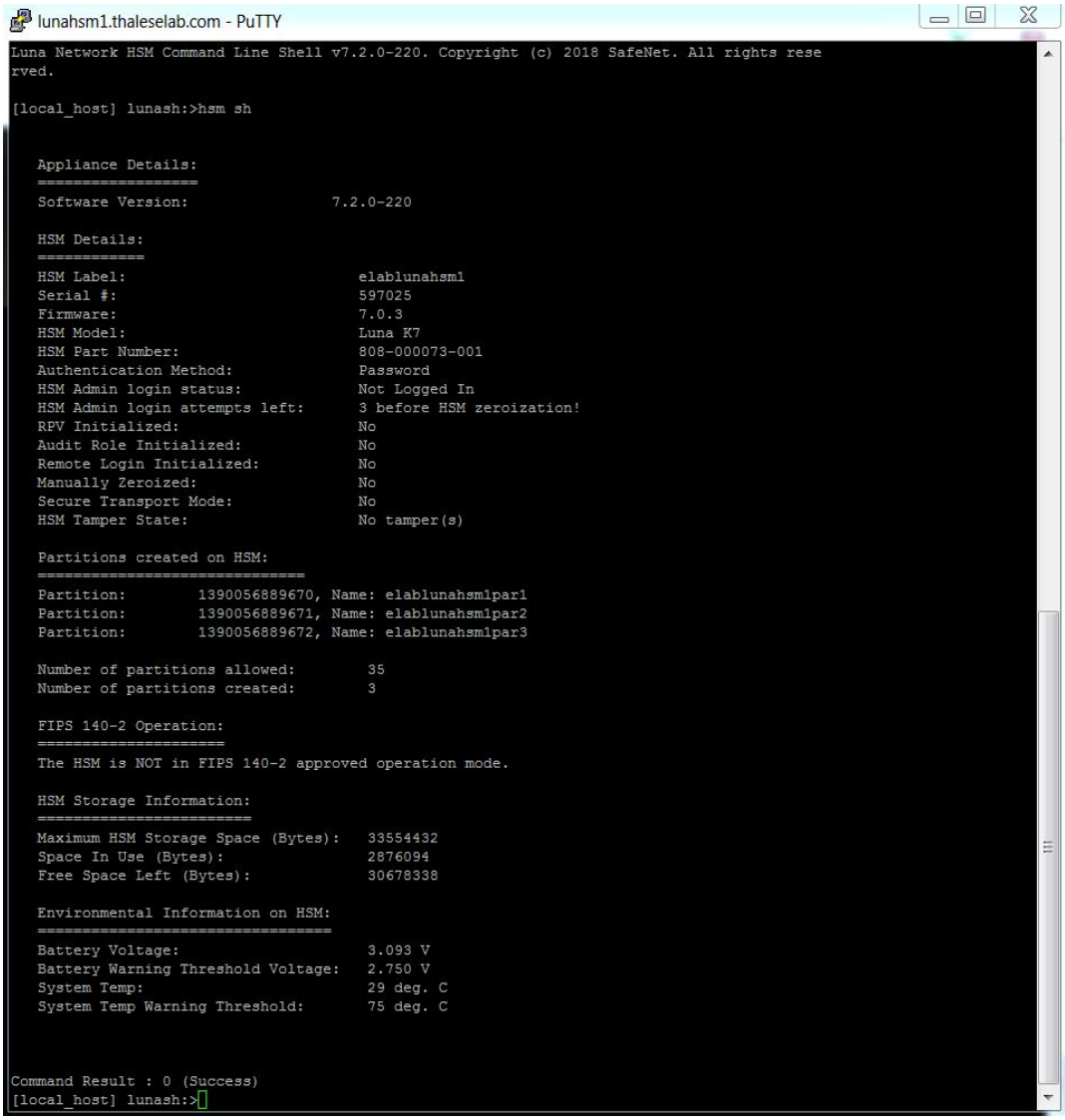
- Accept the server key, select 'Yes'



- Login using the appliance credentials given in the email
- Use Appliance username and Appliance password for this step.
- Example shown below:

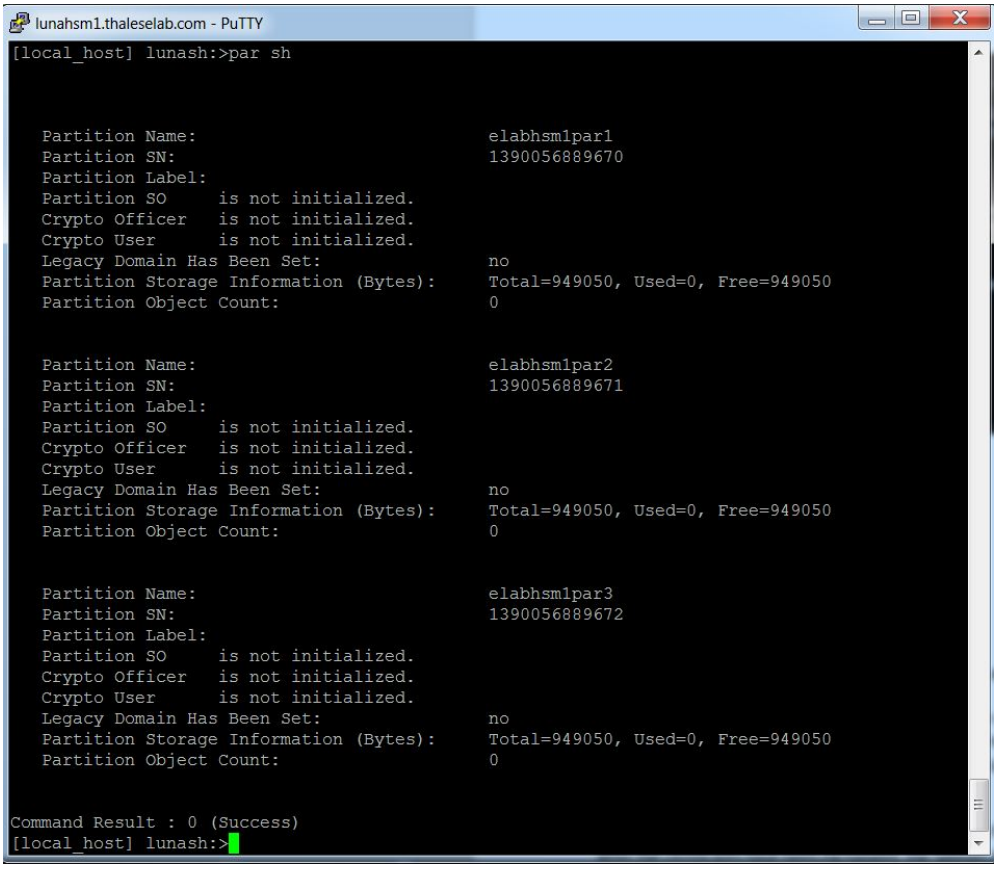


- Run the 'hsm show' command. Verify your assigned partition is on this HSM appliance



```
lunahsm1.thaleselab.com - PuTTY
Luna Network HSM Command Line Shell v7.2.0-220. Copyright (c) 2018 SafeNet. All rights rese
rved.

[local_host] lunash:>hsm sh


   Appliance Details:
   ==================
   Software Version:                7.2.0-220

   HSM Details:
   ============
   HSM Label:                          elablunahsm1
   Serial #:                           597025
   Firmware:                           7.0.3
   HSM Model:                          Luna K7
   HSM Part Number:                    808-000073-001
   Authentication Method:              Password
   HSM Admin login status:             Not Logged In
   HSM Admin login attempts left:      3 before HSM zeroization!
   RPV Initialized:                    No
   Audit Role Initialized:             No
   Remote Login Initialized:           No
   Manually Zeroized:                  No
   Secure Transport Mode:              No
   HSM Tamper State:                   No tamper(s)

   Partitions created on HSM:
   ==============================
   Partition:        1390056889670, Name: elablunahsm1par1
   Partition:        1390056889671, Name: elablunahsm1par2
   Partition:        1390056889672, Name: elablunahsm1par3

   Number of partitions allowed:        35
   Number of partitions created:        3

   FIPS 140-2 Operation:
   =====================
   The HSM is NOT in FIPS 140-2 approved operation mode.

   HSM Storage Information:
   ========================
   Maximum HSM Storage Space (Bytes):   33554432
   Space In Use (Bytes):                2876094
   Free Space Left (Bytes):             30678338

   Environmental Information on HSM:
   =================================
   Battery Voltage:                     3.093 V
   Battery Warning Threshold Voltage:   2.750 V
   System Temp:                         29 deg. C
   System Temp Warning Threshold:       75 deg. C



Command Result : 0 (Success)
[local_host] lunash:>
```

- Enter the following command to view the information about your assigned partition 'partition show -partition <partition name> ‘
- Example below of viewing an assigned partition with the name 'elabhsm1par1'



```
lunahsm1.thaleselab.com - PuTTY
[local_host] lunash:>par sh


   Partition Name:                          elabhsm1par1
   Partition SN:                            1390056889670
   Partition Label:
   Partition SO     is not initialized.
   Crypto Officer   is not initialized.
   Crypto User      is not initialized.
   Legacy Domain Has Been Set:              no
   Partition Storage Information (Bytes):   Total=949050, Used=0, Free=949050
   Partition Object Count:                  0


   Partition Name:                          elabhsm1par2
   Partition SN:                            1390056889671
   Partition Label:
   Partition SO     is not initialized.
   Crypto Officer   is not initialized.
   Crypto User      is not initialized.
   Legacy Domain Has Been Set:              no
   Partition Storage Information (Bytes):   Total=949050, Used=0, Free=949050
   Partition Object Count:                  0


   Partition Name:                          elabhsm1par3
   Partition SN:                            1390056889672
   Partition Label:
   Partition SO     is not initialized.
   Crypto Officer   is not initialized.
   Crypto User      is not initialized.
   Legacy Domain Has Been Set:              no
   Partition Storage Information (Bytes):   Total=949050, Used=0, Free=949050
   Partition Object Count:                  0


Command Result : 0 (Success)
[local_host] lunash:>
```

3.3 Establish a connection between the Luna Client and the Luna appliance HSM partition using the tool "lunacm"

a) Use the client name as suggested below for registering your Luna Client workstation/server.

(Warning: You'll be able to do this step successfully only after sending your public IP from which you'll be accessing the HSM to your sales engineer as mentioned in introduction of this eLab setup)

When registering your client name please use the same exact username provided in the email.

Example: elabhsm1par1usr

b) Establish a connection between the Luna Client and the Luna appliance

Open a command prompt, use 'run as administrator' option. cdto the LunaClient install directory. C:\Program Files\SafeNet\LunaClient>

```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd "\Program Files\SafeNet\LunaClient"

C:\Program Files\SafeNet\LunaClient>dir
 Volume in drive C has no label.
 Volume Serial Number is 0C26-4E8D

 Directory of C:\Program Files\SafeNet\LunaClient

02/22/2019  07:20 AM    <DIR>          .
02/22/2019  07:20 AM    <DIR>          ..
06/08/2018  03:08 PM           149,836 008-010068-001_EULA_HSM7_SW_revA.pdf
02/22/2019  07:19 AM    <DIR>          cert
08/24/2018  04:14 PM           369,544 ckdemo.exe
08/24/2018  04:14 PM           641,416 cklog201.dll
08/24/2018  04:14 PM         1,391,496 Cmu.exe
02/22/2019  07:19 AM    <DIR>          config
08/24/2018  04:14 PM         3,686,792 cryptoki.dll
02/22/2019  07:20 AM             1,899 crystoki.ini
02/22/2019  07:19 AM    <DIR>          CSP
02/22/2019  07:19 AM    <DIR>          data
02/22/2019  07:19 AM    <DIR>          G5Driver
02/22/2019  07:19 AM    <DIR>          JCProv
02/22/2019  07:19 AM    <DIR>          JSP
02/22/2019  07:19 AM    <DIR>          KSP
08/24/2018  04:14 PM         3,339,144 lunacm.exe
08/24/2018  04:14 PM         3,428,232 lunadiag.exe
08/24/2018  04:14 PM            25,992 lunareset.exe
08/24/2018  04:14 PM           379,784 multitoken.exe
12/16/2016  08:29 AM             7,145 openssl.cnf
08/24/2018  04:14 PM         3,478,408 PedClient.exe
02/22/2019  07:19 AM    <DIR>          PedClient_service
08/24/2018  04:14 PM         2,421,640 PedServer.exe
02/22/2019  07:20 AM               698 pedServer.ini
06/08/2018  03:08 PM           451,072 PLINK.EXE
06/08/2018  03:08 PM           458,752 PSCP.EXE
06/08/2018  03:08 PM           640,000 PUTTY.EXE
06/08/2018  03:08 PM           683,594 PUTTY.HLP
08/24/2018  04:14 PM         2,774,408 rbs.exe
08/24/2018  04:14 PM         3,465,096 rbs_processor2.dll
02/22/2019  07:19 AM    <DIR>          RemotePEDDriver
08/24/2018  04:14 PM           283,016 salogin.exe
08/24/2018  04:14 PM           235,912 shim.dll
02/22/2019  07:19 AM    <DIR>          snmp
02/22/2019  07:19 AM    <DIR>          softtoken
08/24/2018  04:14 PM         1,909,640 softtoken.dll
08/24/2018  04:14 PM         3,518,856 VTL.exe
02/22/2019  07:19 AM    <DIR>          win32
               24 File(s)     33,742,372 bytes
               15 Dir(s)  16,219,332,608 bytes free

C:\Program Files\SafeNet\LunaClient>_
```

- Run 'lunacm' command

```
Administrator: Command Prompt - lunacm.exe
C:\Program Files\SafeNet\LunaClient>lunacm.exe
lunacm.exe (64-bit) v10.1.0-32. Copyright (c) 2019 SafeNet. All rights reserved.


        Available HSMs:


        Current Slot Id: None


lunacm:>
```

Now, run the following command

Syntax:

lunacm:>clientconfig deploy -server <IP or Hostname of Luna appliance> –user <appliance username provided in email> -password <Luna appliance PW provided in email> -client <Client name as suggested in step 3.a> -partition <partition name provided in email>

```
lunacm:> clientconfig deploy -server lunahsm1.thaleselab.com -username elabhsm1par2usr -password J0inThal#$ -c
lient elabhsm1par2usr -partition elabhsm1par2
Error: unknown parameter 'username'.

  Register the client with the appliance.
  The following options are available:

  Options              Short     Description
  ------------------------------------------

  -server              -n        Server hostname or IP address  (mandatory)
  -client              -c        Client hostname or IP address (mandatory)
  -partition           -par      Partition name to assign to the client (mandatory)
  -password            -pw       Appliance admin role user's password
  -user                -ur       Appliance admin role user's name, default is admin
  -hsmPassword         -hsmPw    HSM SO role password, only needed if HSM SO login enforcement is enabled
  -regen               -rg       Regenerate new and replace existing client's certificate
  -force               -f        Force Action
  -verbose             -v        Show verbose logs

  Syntax: clientconfig deploy -server <string> -client <string> -partition <string> [-password <string>] [-use
r <string>] [-hsmPassword <string>] [-regen] [-force] [-verbose]

Command Result : 0x4 (Invalid arguments)


lunacm:> clientconfig deploy -server lunahsm1.thaleselab.com -ur elabhsm1par2usr -password J0inThal#$ -client
elabhsm1par2usr -partition elabhsm1par2

Please wait while we set up the connection to the HSM. This may take several minutes...

Please enter appliance admin role user's password:
```

Example:
lunacm:>clientconfig deploy -server lunahsm1.thaleselab.com -ur elabhsm1par1usr1 -password J0inThal3$ -client elabhsm1par1usr1 -partition elabhsm1par1

```
Administrator: Command Prompt - lunacm.exe

lunacm:> clientconfig deploy -server lunahsm1.thaleselab.com -ur elabhsm1par1usr
1 -password J0inThal3$ -client elabhsm1par1usr101 -partition elabhsm1par1

Please wait while we set up the connection to the HSM. This may take several min
utes...

Command Result : No Error

lunacm.exe (64-bit) v10.1.0-32. Copyright (c) 2019 SafeNet. All rights reserved.



        Available HSMs:

        Slot Id ->              0
        Label ->                elabhsm1par1
        Serial Number ->        1390056889670
        Model ->                LunaSA 7.4.0
        Firmware Version ->     7.3.3
        Configuration ->        Luna User Partition With SO (PW) Key Export With
 Cloning Mode
        Slot Description ->     Net Token Slot
        FM HW Status ->         FM Ready


        Current Slot Id: 0


lunacm:>
```

Back to the putty session verify client is registered on the Luna appliance.
run 'client list' command

```
lunahsm1.thaleselab.com - PuTTY
[local_host] lunash:>client list

registered client 1: elabhsm1par1usr1


Command Result : 0 (Success)
[local_host] lunash:>
```

Show the details of the client registered.
Run 'client show -client <client name or IP>
Example:

```
lunahsm1.thaleselab.com - PuTTY
[local_host] lunash:>client show  -client elabhsm1par1usr1


ClientID:     elabhsm1par1usr1
Hostname:     elabhsm1par1usr1
Partitions:   "elabhsm1par1"


Command Result : 0 (Success)
[local_host] lunash:>
```

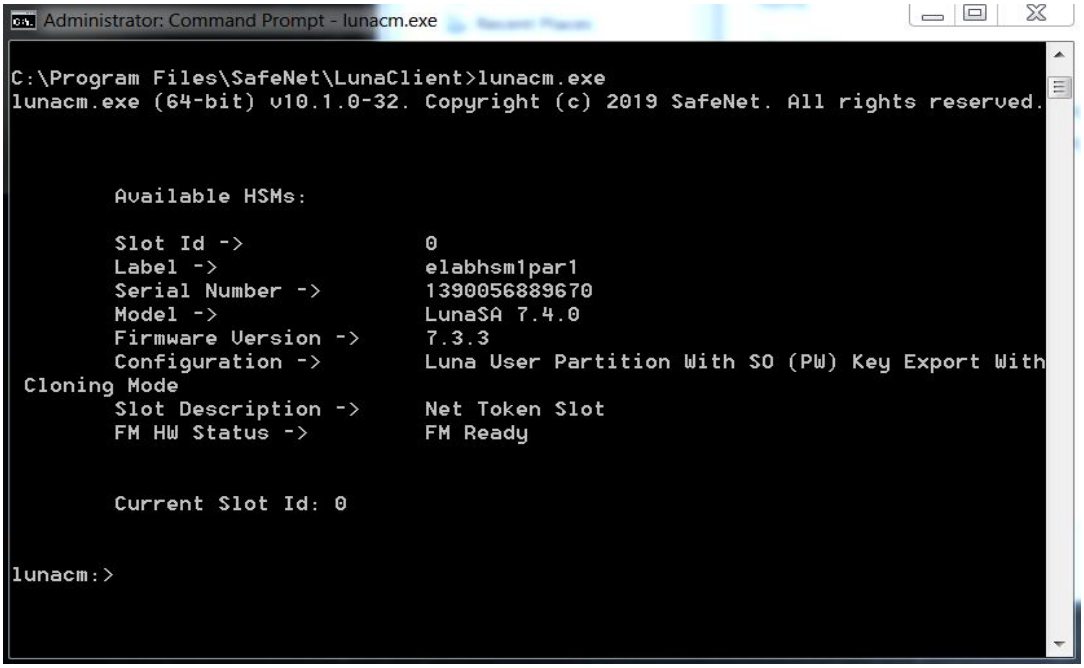
Back to the Luna client cmd window [C:\Program Files\SafeNet\LunaClient>] to verify Luna appliance registration on the client side
Run **'vtl listservers'** command
Run **'vtl verify'** command

```
Administrator: Command Prompt

C:\Program Files\SafeNet\LunaClient>vtl listservers
vtl (64-bit) v10.1.0-32. Copyright (c) 2019 SafeNet. All rights reserved.

Server: lunahsm1.thaleselab.com

C:\Program Files\SafeNet\LunaClient>vtl verify
vtl (64-bit) v10.1.0-32. Copyright (c) 2019 SafeNet. All rights reserved.


The following Luna SA Slots/Partitions were found:

Slot    Serial #               Label
====    ===============        =====
   0       1390056889670       elabhsm1par1
```

3.4 Verify Network Trust Link Service (i.e. NTLS) – Port 1792

The Luna Network HSM is accessed by the HSM Client (i.e. app server) and other network appliances via an encrypted IP protocol called NTLs listening on TCP port 1792. Network Trust Links (NTL) is secure, authenticated network connections between the HSM and Clients. NTLs use SSL with client authentication to protect all communications between HSM server and its Clients.
The Communication is secured through the exchange of certificates between the HSM Client and the HSM server. After the certificates have been exchanged, the SafeNet HSM module registers the app server as a client and assigns a partition. The HSM client then registers the SafeNet HSM and partition as a server.

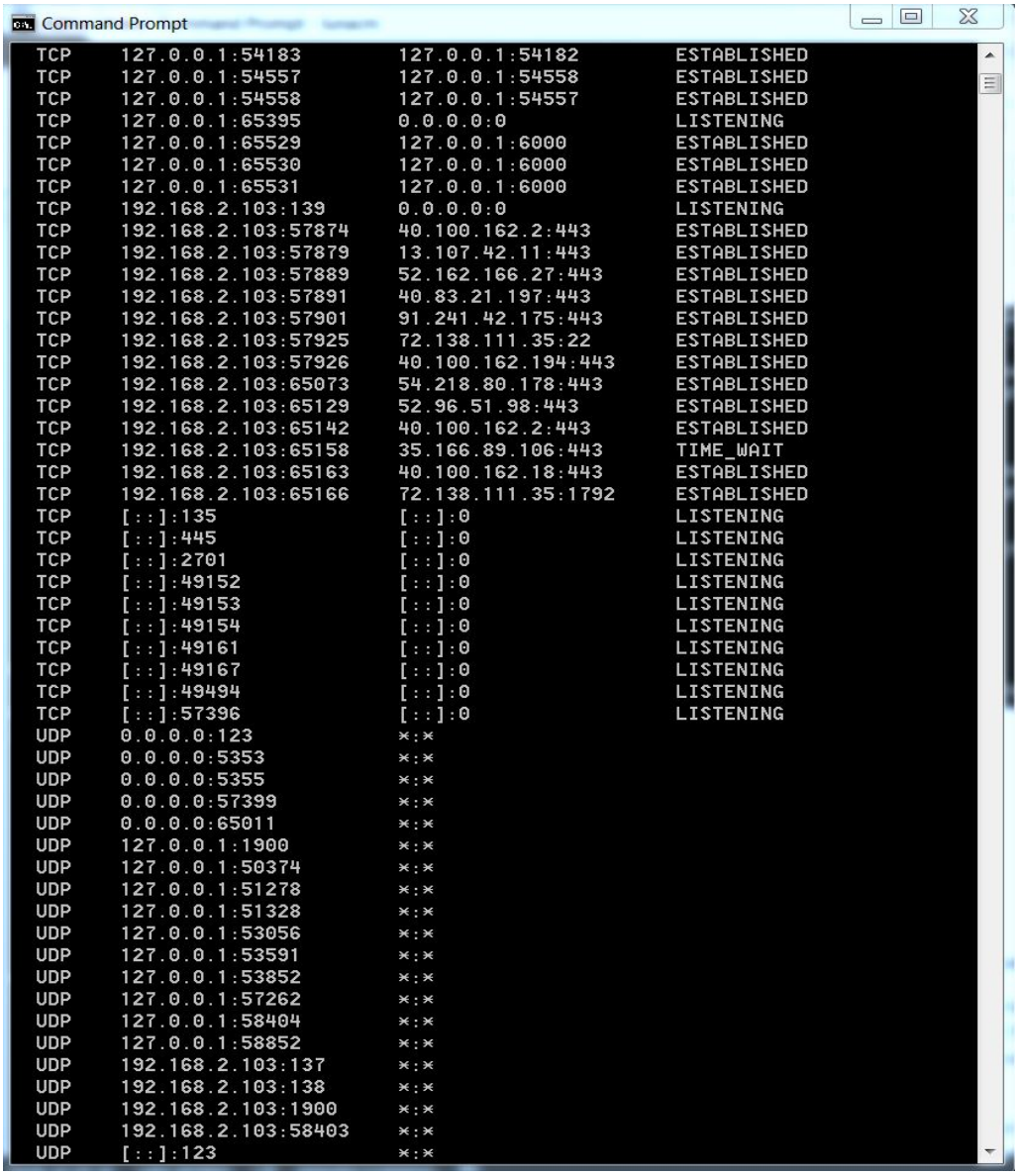
a. **Launch "lunacm" (i.e. application) to initialize a connection with the HSM**



b. Open a new CMD prompt, verify NTLS is established

Run >**netstat -an | find "1792“** (or Run >netstat –an)
Example shown below:

```
Command Prompt
C:\Users\jjesudos>netstat -an | find "1792"
  TCP    192.168.2.103:65166    72.138.111.35:1792     ESTABLISHED

C:\Users\jjesudos>netstat -an | find "22"
  TCP    0.0.0.0:52230          0.0.0.0:0              LISTENING
  TCP    192.168.2.103:57925    72.138.111.35:22       ESTABLISHED

C:\Users\jjesudos>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2701           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:6000           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:16386          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49161          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49167          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49494          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:52230          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:57396          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1501         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:6000         127.0.0.1:65529        ESTABLISHED
  TCP    127.0.0.1:6000         127.0.0.1:65530        ESTABLISHED
  TCP    127.0.0.1:6000         127.0.0.1:65531        ESTABLISHED
  TCP    127.0.0.1:7778         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:7778         127.0.0.1:49190        ESTABLISHED
  TCP    127.0.0.1:16388        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49155        127.0.0.1:49156        ESTABLISHED
  TCP    127.0.0.1:49156        127.0.0.1:49155        ESTABLISHED
  TCP    127.0.0.1:49162        127.0.0.1:49163        ESTABLISHED
  TCP    127.0.0.1:49163        127.0.0.1:49162        ESTABLISHED
  TCP    127.0.0.1:49164        127.0.0.1:49165        ESTABLISHED
  TCP    127.0.0.1:49165        127.0.0.1:49164        ESTABLISHED
  TCP    127.0.0.1:49188        127.0.0.1:49189        ESTABLISHED
  TCP    127.0.0.1:49189        127.0.0.1:49188        ESTABLISHED
  TCP    127.0.0.1:49190        127.0.0.1:7778         ESTABLISHED
  TCP    127.0.0.1:49191        127.0.0.1:49192        ESTABLISHED
  TCP    127.0.0.1:49192        127.0.0.1:49191        ESTABLISHED
  TCP    127.0.0.1:49234        127.0.0.1:49235        ESTABLISHED
  TCP    127.0.0.1:49235        127.0.0.1:49234        ESTABLISHED
  TCP    127.0.0.1:49236        127.0.0.1:49237        ESTABLISHED
  TCP    127.0.0.1:49237        127.0.0.1:49236        ESTABLISHED
  TCP    127.0.0.1:49241        127.0.0.1:49242        ESTABLISHED
  TCP    127.0.0.1:49242        127.0.0.1:49241        ESTABLISHED
  TCP    127.0.0.1:54182        127.0.0.1:54183        ESTABLISHED
  TCP    127.0.0.1:54183        127.0.0.1:54182        ESTABLISHED
```



```
Command Prompt
  TCP    127.0.0.1:54183        127.0.0.1:54182        ESTABLISHED
  TCP    127.0.0.1:54557        127.0.0.1:54558        ESTABLISHED
  TCP    127.0.0.1:54558        127.0.0.1:54557        ESTABLISHED
  TCP    127.0.0.1:65395        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:65529        127.0.0.1:6000         ESTABLISHED
  TCP    127.0.0.1:65530        127.0.0.1:6000         ESTABLISHED
  TCP    127.0.0.1:65531        127.0.0.1:6000         ESTABLISHED
  TCP    192.168.2.103:139      0.0.0.0:0              LISTENING
  TCP    192.168.2.103:57874    40.100.162.2:443       ESTABLISHED
  TCP    192.168.2.103:57879    13.107.42.11:443       ESTABLISHED
  TCP    192.168.2.103:57889    52.162.166.27:443      ESTABLISHED
  TCP    192.168.2.103:57891    40.83.21.197:443       ESTABLISHED
  TCP    192.168.2.103:57901    91.241.42.175:443      ESTABLISHED
  TCP    192.168.2.103:57925    72.138.111.35:22       ESTABLISHED
  TCP    192.168.2.103:57926    40.100.162.194:443     ESTABLISHED
  TCP    192.168.2.103:65073    54.218.80.178:443      ESTABLISHED
  TCP    192.168.2.103:65129    52.96.51.98:443        ESTABLISHED
  TCP    192.168.2.103:65142    40.100.162.2:443       ESTABLISHED
  TCP    192.168.2.103:65158    35.166.89.106:443      TIME_WAIT
  TCP    192.168.2.103:65163    40.100.162.18:443      ESTABLISHED
  TCP    192.168.2.103:65166    72.138.111.35:1792     ESTABLISHED
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:2701              [::]:0                 LISTENING
  TCP    [::]:49152             [::]:0                 LISTENING
  TCP    [::]:49153             [::]:0                 LISTENING
  TCP    [::]:49154             [::]:0                 LISTENING
  TCP    [::]:49161             [::]:0                 LISTENING
  TCP    [::]:49167             [::]:0                 LISTENING
  TCP    [::]:49494             [::]:0                 LISTENING
  TCP    [::]:57396             [::]:0                 LISTENING
  UDP    0.0.0.0:123            *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5355           *:*
  UDP    0.0.0.0:57399          *:*
  UDP    0.0.0.0:65011          *:*
  UDP    127.0.0.1:1900         *:*
  UDP    127.0.0.1:50374        *:*
  UDP    127.0.0.1:51278        *:*
  UDP    127.0.0.1:51328        *:*
  UDP    127.0.0.1:53056        *:*
  UDP    127.0.0.1:53591        *:*
  UDP    127.0.0.1:53852        *:*
  UDP    127.0.0.1:57262        *:*
  UDP    127.0.0.1:58404        *:*
  UDP    127.0.0.1:58852        *:*
  UDP    192.168.2.103:137      *:*
  UDP    192.168.2.103:138      *:*
  UDP    192.168.2.103:1900     *:*
  UDP    192.168.2.103:58403    *:*
  UDP    [::]:123               *:*
```

4. Windows (Microsoft Authenticode) signing

4.1 Configuration

4.1.1 SignTool installation on Windows 2016 server

Microsoft SignTool is a command line tool that signs, verifies, and timestamps files to simplify the code signing process. The tool is installed in the \bin folder of the Microsoft Windows Software Development Kit (SDK) installation path.

- SignTool is available as part of the Windows SDK, which you can download from https://go.microsoft.com/fwlink/?LinkID=698771
- Install the SignTool application and it will be located in the C:\Program Files (x86)\Windows Kits\10\bin\x64

- Once installation completes, add SignTool application path in the environment variable to access from everywhere.

3.2 Signing Windows executable on Windows server 2016

Backend configuration of the Signing Windows Executable

- Install **Nodejs** version minimum 10.0.1 (If you have already installed, please ignore this step)
- Install **pm2** globally using bellow command (Run command prompt as administrator):

```
> pm2 install –g
```

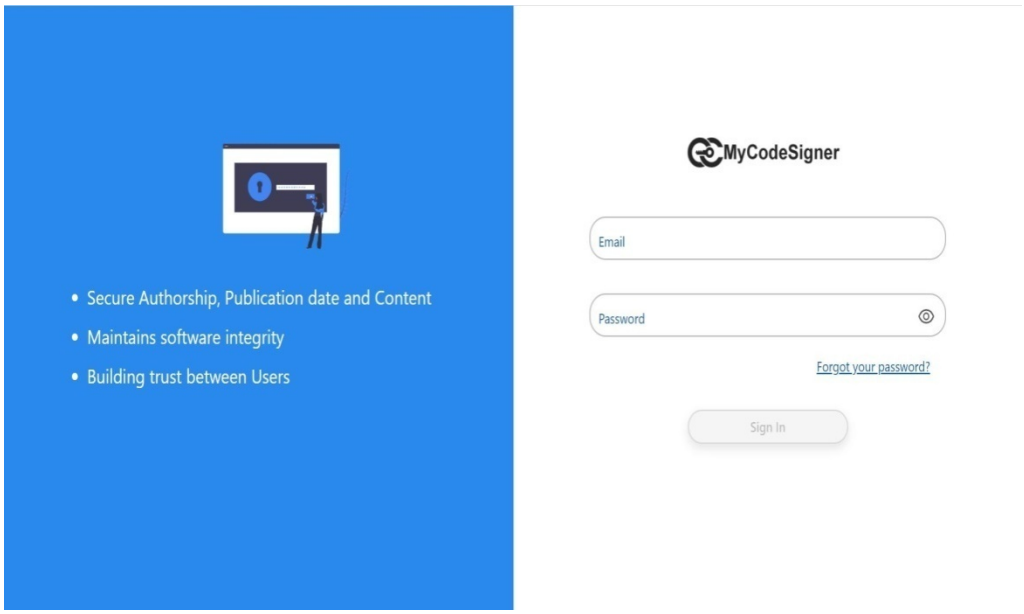
- Create a Directory called **gemaltoBackendServer** in **C:\Users\gemaltoBackendServer**.
- Copy **gemaltoBackendServer.zip** in the **C:\Users\gemaltoBackendServer** Directory.
- Unzip **gemaltoBackendServer.zip** in same Directory.
- Go to **C:\Users\ gemaltoBackendServer** Directory and run **server.js** as a **pm2**.

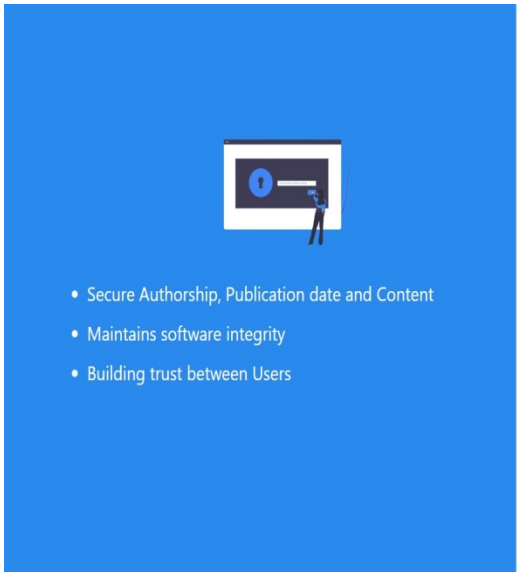
```
>pm2 start server.js
```

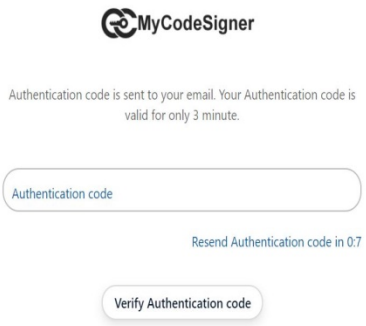
3.3 Signing windows executable using MyCodesigner application

3.3.1 Login

- Go to MyCodesigner virtual machine, open browser and type localhost in the URL. It will take into the login screen.
- Enter CO Admin credentials and do 2-factor-authentication.
- Screenshot of login and 2-factor authentication

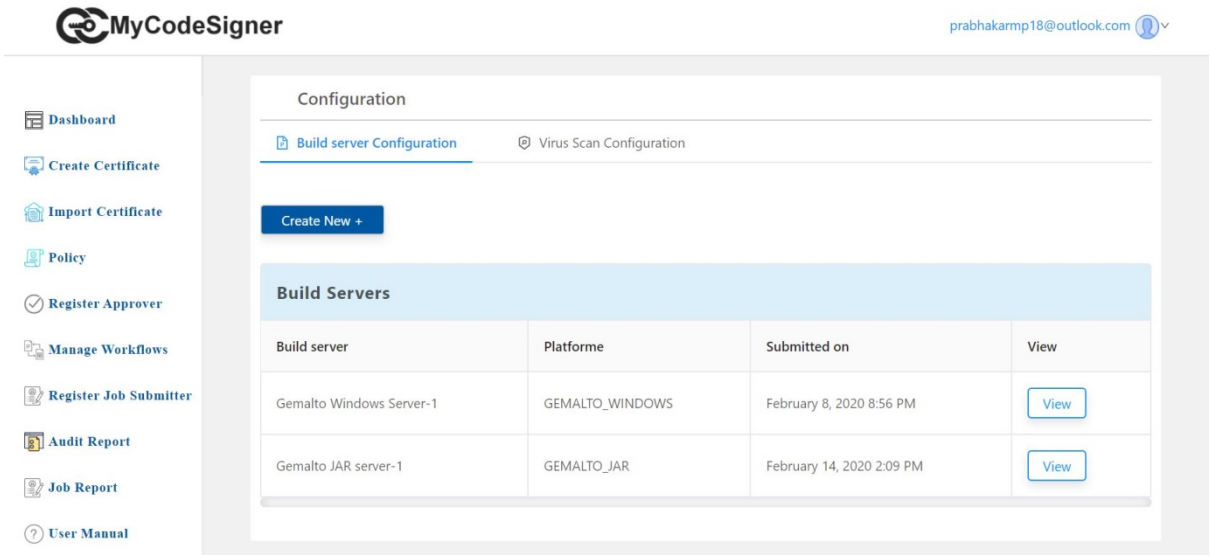


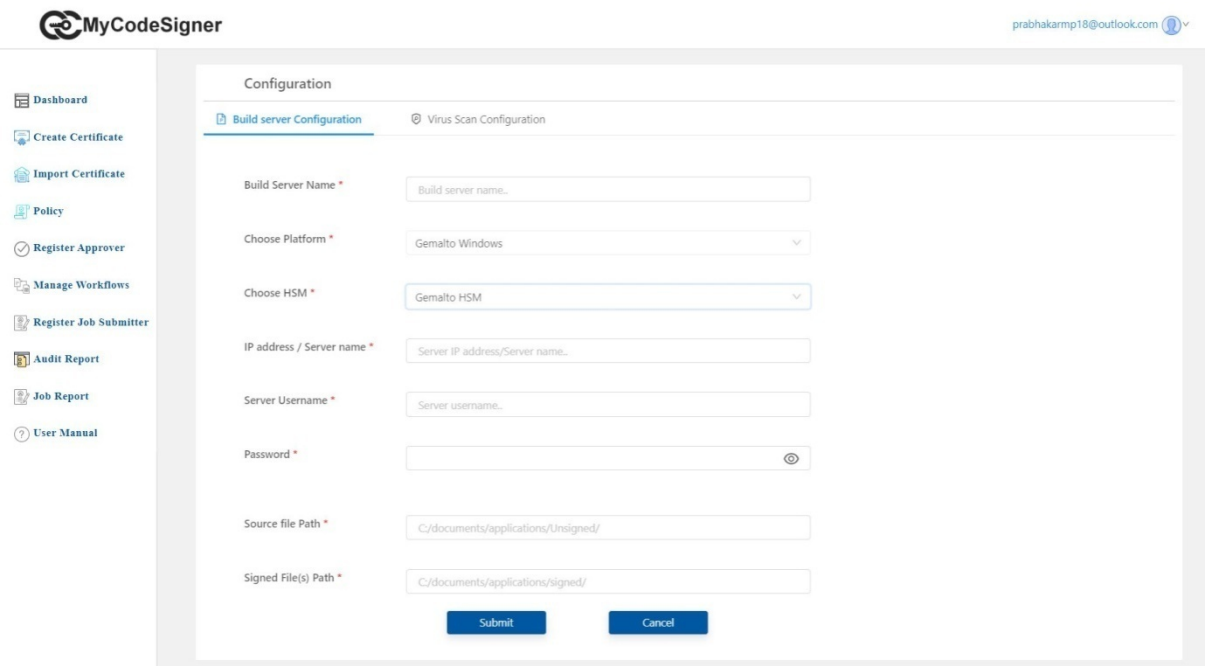


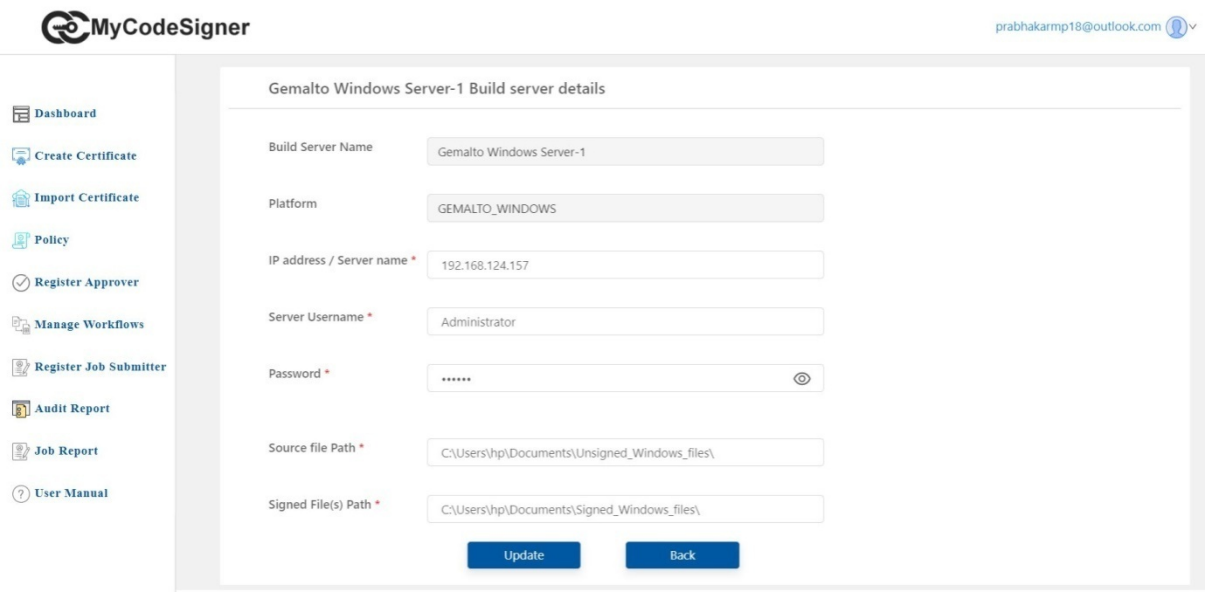


3.3.2 Create build server

- After login into CO Admin, it will take you to the dashboard and click on Configuration option to create build server.
- On Build server configuration tab, click on Create new button to create new build server.
- Enter build server name, choose platform as Gemalto Windows from drop-down, choose HSM as Gemalto HSM from drop-down, enter the other required details and click submit button to create a new build server.
- To view build server configuration, click view button on the table to view particular build server.
- To update build server configuration, update the details except Build server name and Platform, fill the other changes want to update and click Update button to update the Build server details.
- Screenshots of the build server

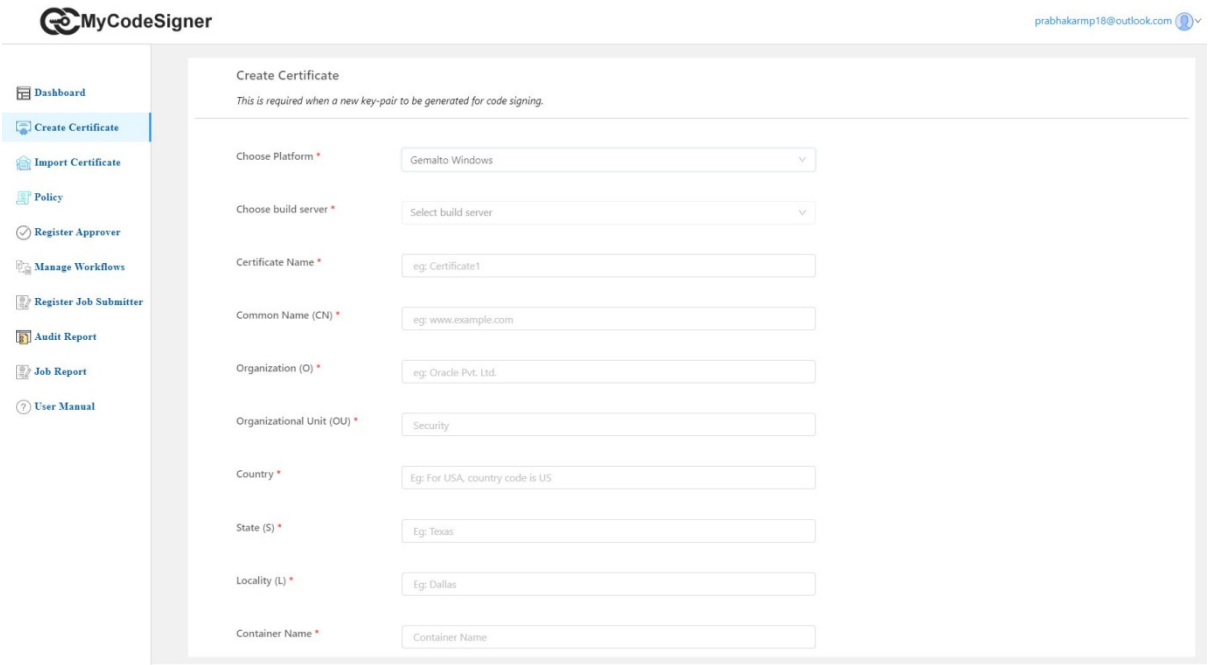






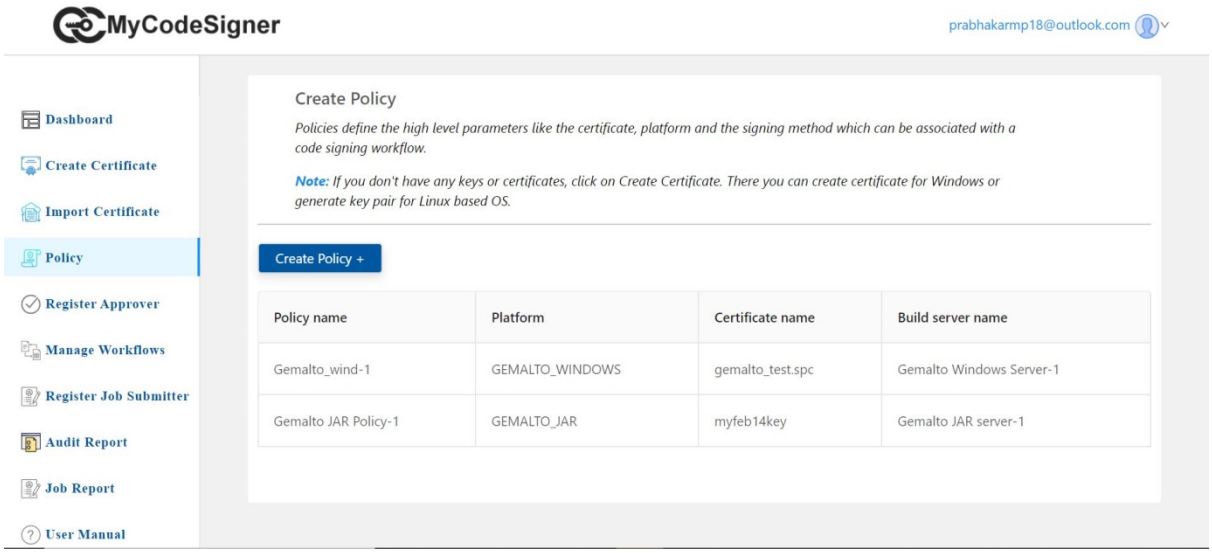
3.3.3 Create Certificate

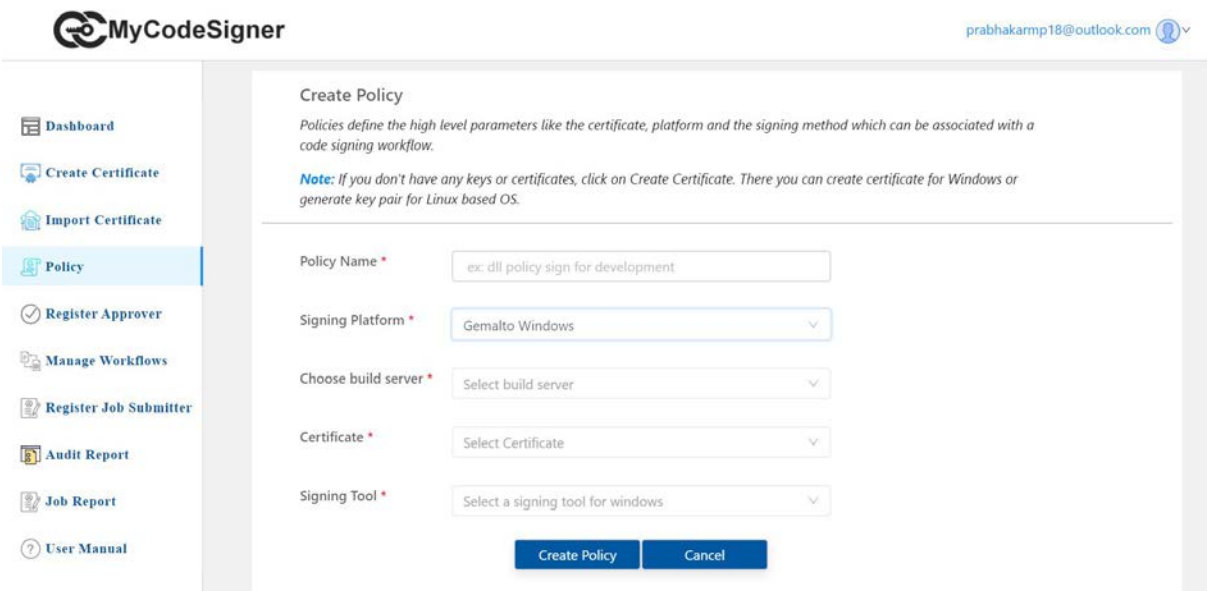
- Create certificate is a self-signed certificate to sign the windows artifact types like exe, dll, msi, etc.
- Click on Create certificate in Dashboard.
- Choose Platform as Gemalto windows and fill the all other details.
- Click Create button to create a certificate.
- Screenshots of the create certificate for windows



3.3.4 Create Policy

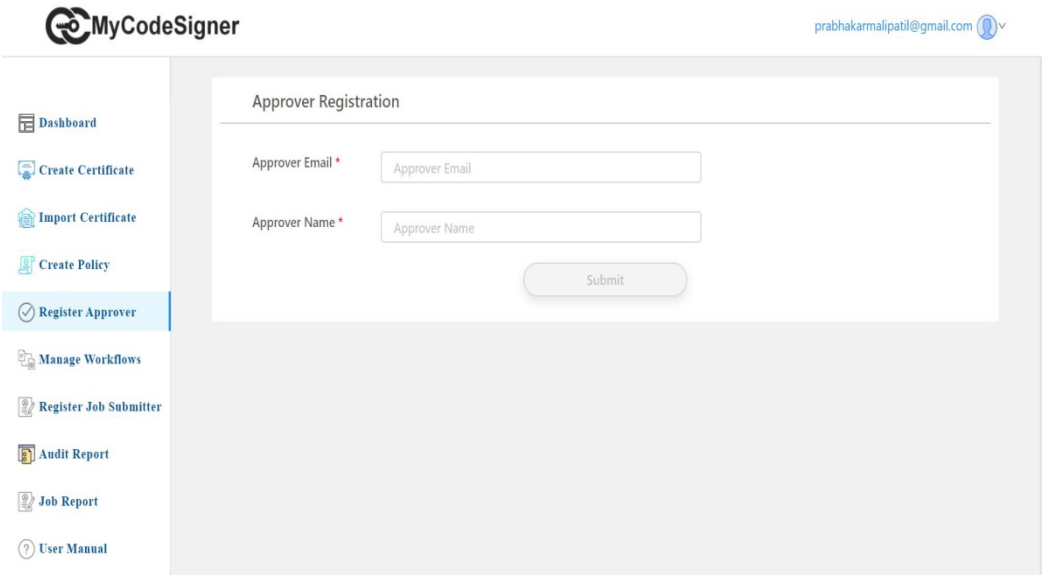
- Policies define the high level parameters like the certificate, platform and the signing method which can be associated with a code signing workflow.
- To create a policy, click on Create policy menu in dashboard.
- Enter the required details to create policy and click on Create policy button.
- Screenshots of the create certificate for windows





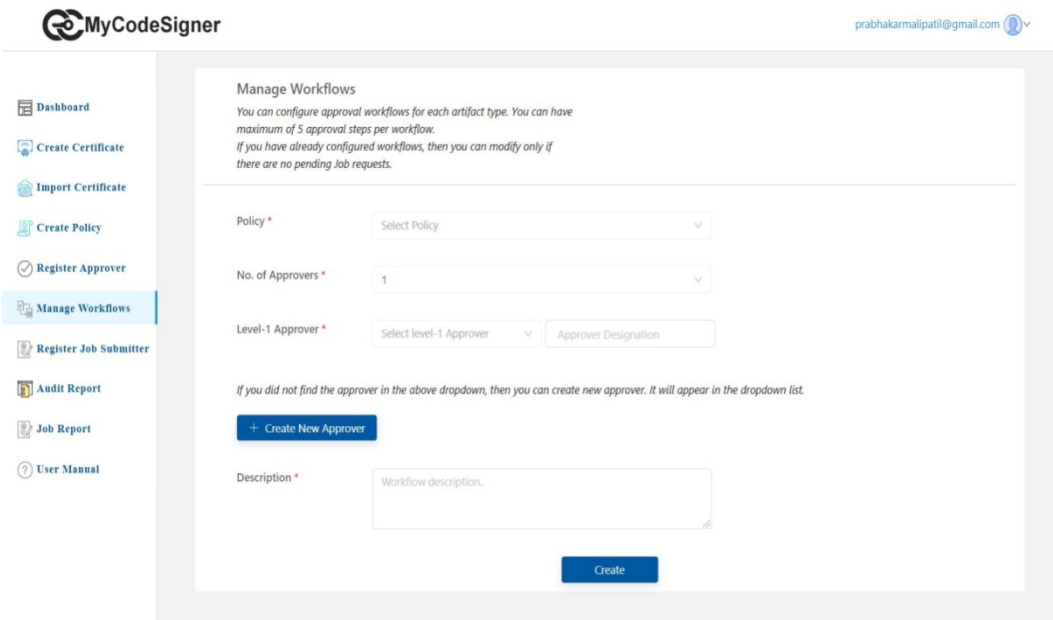
3.3.5 Register Job approver

- Click on Register approver menu in the dashboard.
- Enter Approver email and approver name.
- Click on Submit button to register job approver.
- Screenshot of register job approver



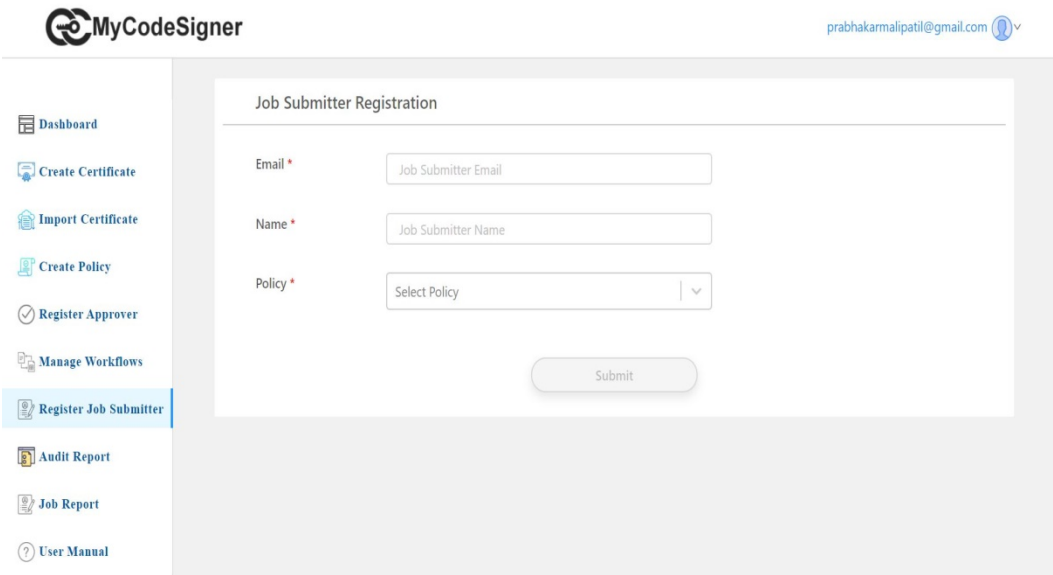
3.3.6 Manage Workflow

- Click on Manage workflow menu in the dashboard.
- Choose policy from the drop-down, select no. of approver, workflow description and each level approver and their designation.
- Click Create button to create new workflow
- Screenshot of the manage workflow



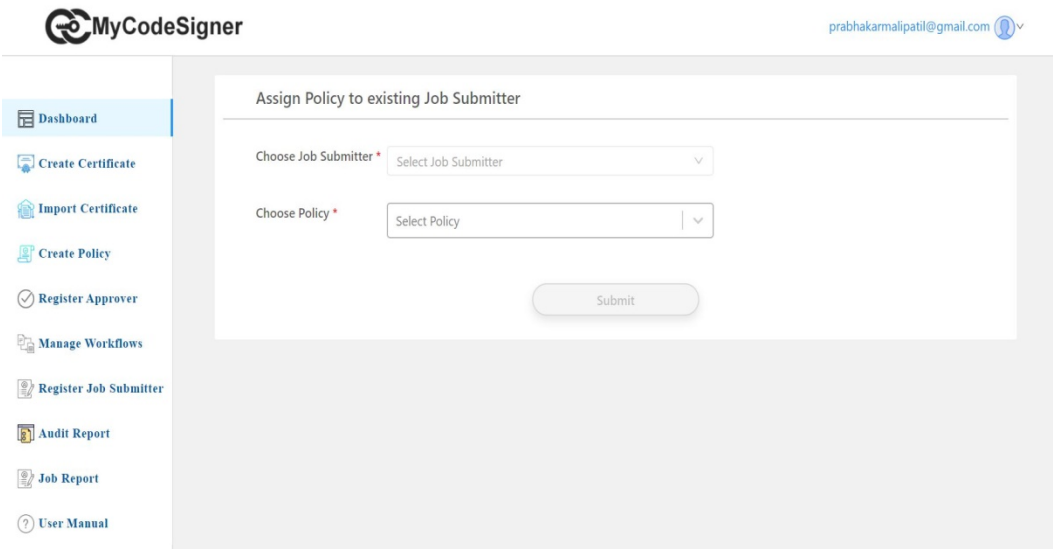
3.3.7 Register Job submitter

- Click on the Register Job submitter menu from the dashboard.
- Enter the job submitter email, name and select policy from the drop-down.
- Click on Submit button to register new Job submitter.
- Screenshot of Register job submitter



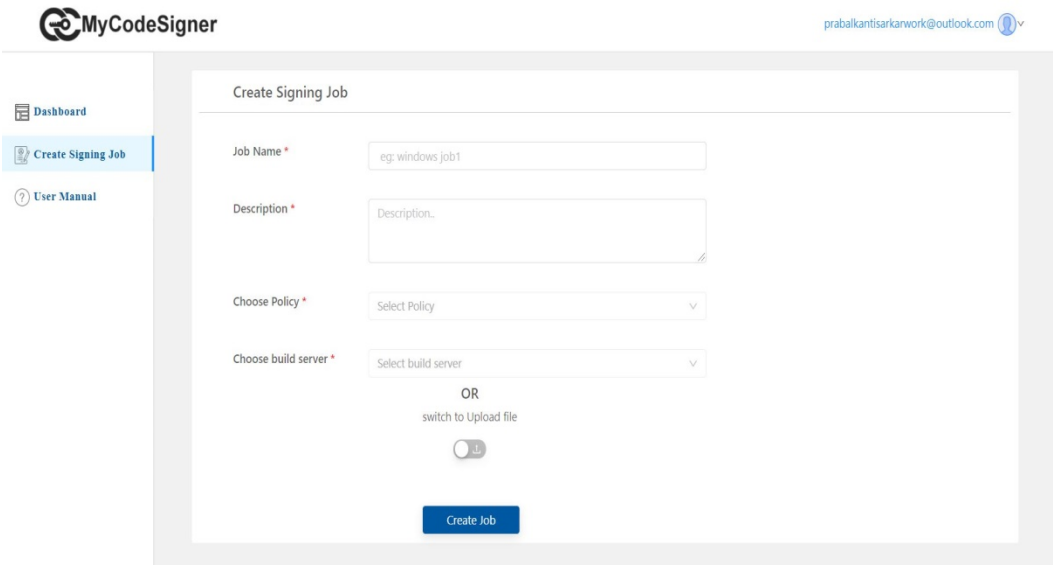
3.3.8 Manage Job submitter

- Click on the Manage Job submitter menu from the dashboard.
- Choose Job submitter from the drop-down and select policy from the drop-down.
- Click on Submit button to assign new policy to the existing job submitter.
- Screenshot of Register job submitter



3.3.9 Create a signing job

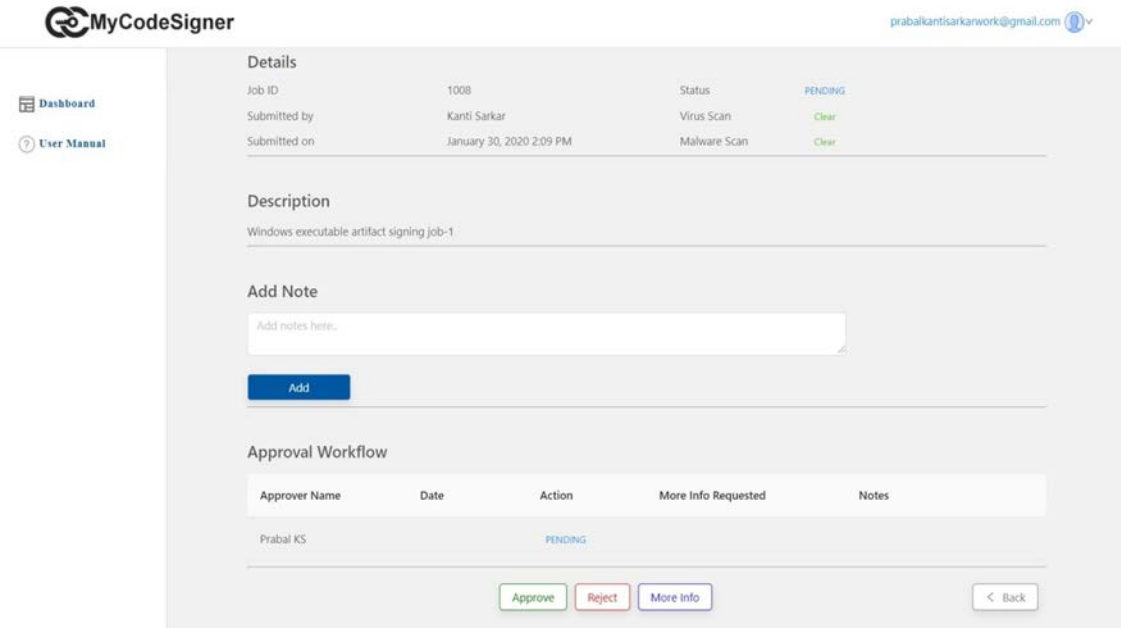
- Login as a job submitter and click on Create signing job menu in the dashboard.
- Enter the job name, description, choose policy from the drop-down, and select the build server from the drop-down.
- Enter the file name to sign the relative path.
- Click on Create Job button to create a job.
- Screenshot of Create a signing job



3.3.10 Action on job (Approve, Reject and more info)

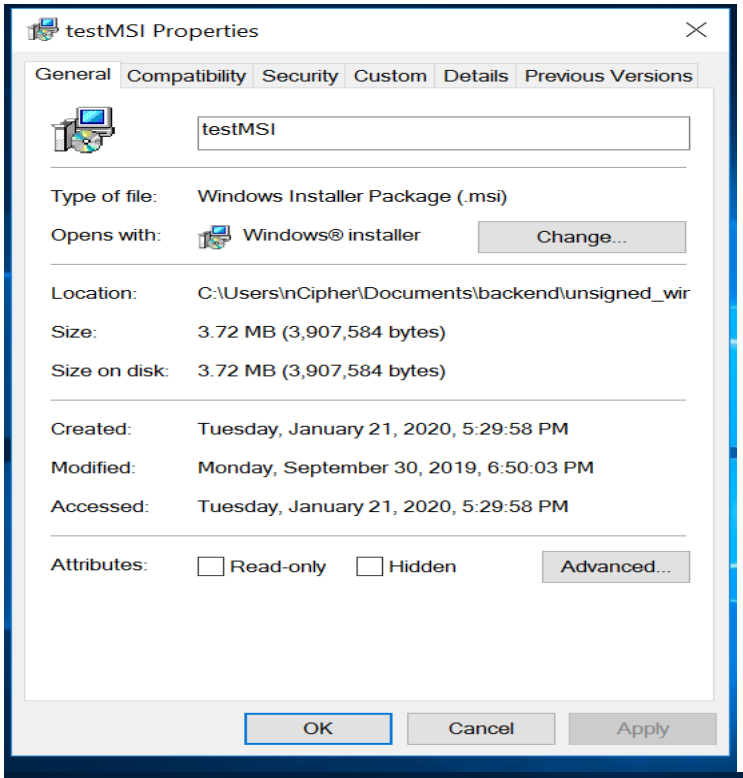
- Login as a job approver and click on Action button in the Signing requests waiting for your approval table of particular job.
- You can add notes to the particular job.

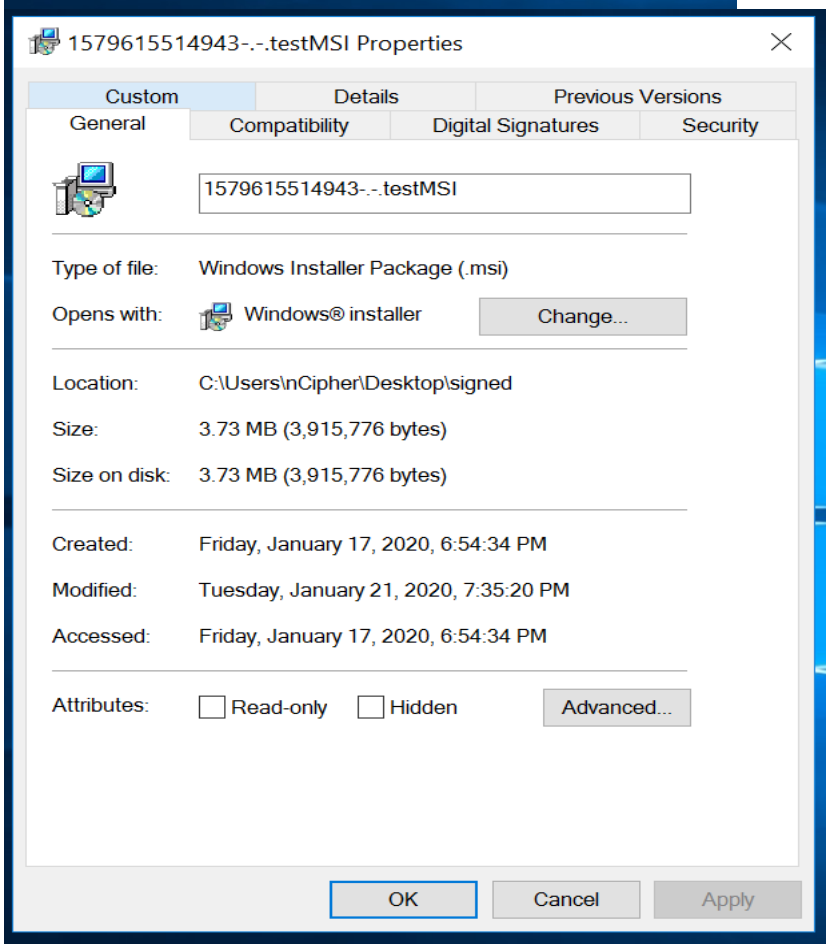
- To approve a job, click on Approve button in the approval workflow table, once the job is approved it will go for signing.
- To reject a job, click on Reject button in the approval workflow table.
- To request more information about a job, click on more info requested button in the approval workflow table.
- Screenshot of Action on job



3.3.11 Verify windows executable after signing

- Go to your windows vm station and go to signed folder/directory and see the digital signature in the main tab.
- Screenshot of verifying windows executable.

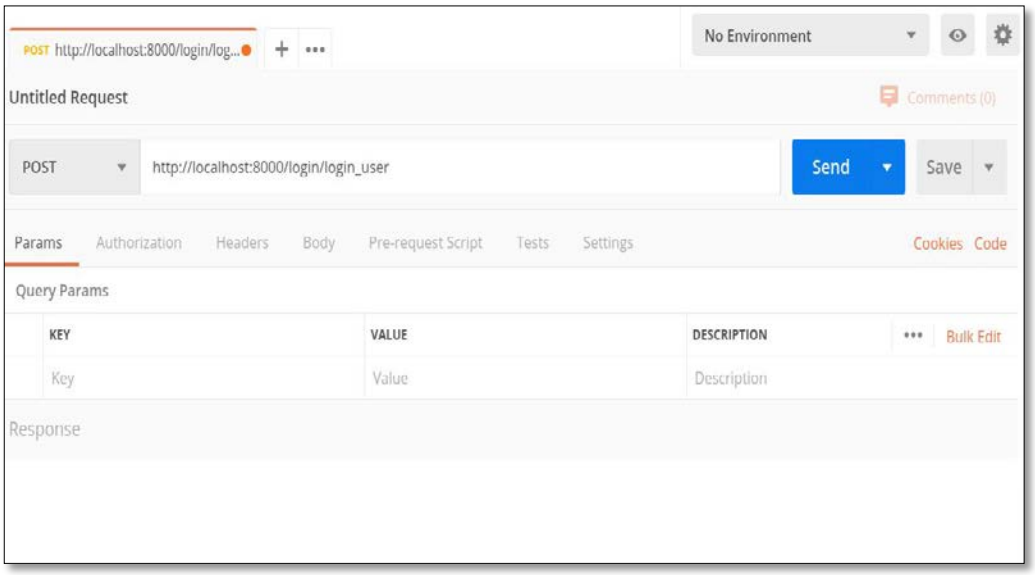

testMSI Properties
General
Compatibility
Security
Custom
Details
Previous Versions
testMSI
Type of file:
Windows Installer Package (.msi)
Opens with:
Windows® installer
Change...
Location:
C:\Users\nCipher\Documents\backend\unsigned_wir
Size:
3.72 MB (3,907,584 bytes)
Size on disk:
3.72 MB (3,907,584 bytes)
Created:
Tuesday, January 21, 2020, 5:29:58 PM
Modified:
Monday, September 30, 2019, 6:50:03 PM
Accessed:
Tuesday, January 21, 2020, 5:29:58 PM
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply


1579615514943-.-.testMSI Properties
Custom
Details
Previous Versions
General
Compatibility
Digital Signatures
Security
1579615514943-.-.testMSI
Type of file:
Windows Installer Package (.msi)
Opens with:
Windows® installer
Change...
Location:
C:\Users\nCipher\Desktop\signed
Size:
3.73 MB (3,915,776 bytes)
Size on disk:
3.73 MB (3,915,776 bytes)
Created:
Friday, January 17, 2020, 6:54:34 PM
Modified:
Tuesday, January 21, 2020, 7:35:20 PM
Accessed:
Friday, January 17, 2020, 6:54:34 PM
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply

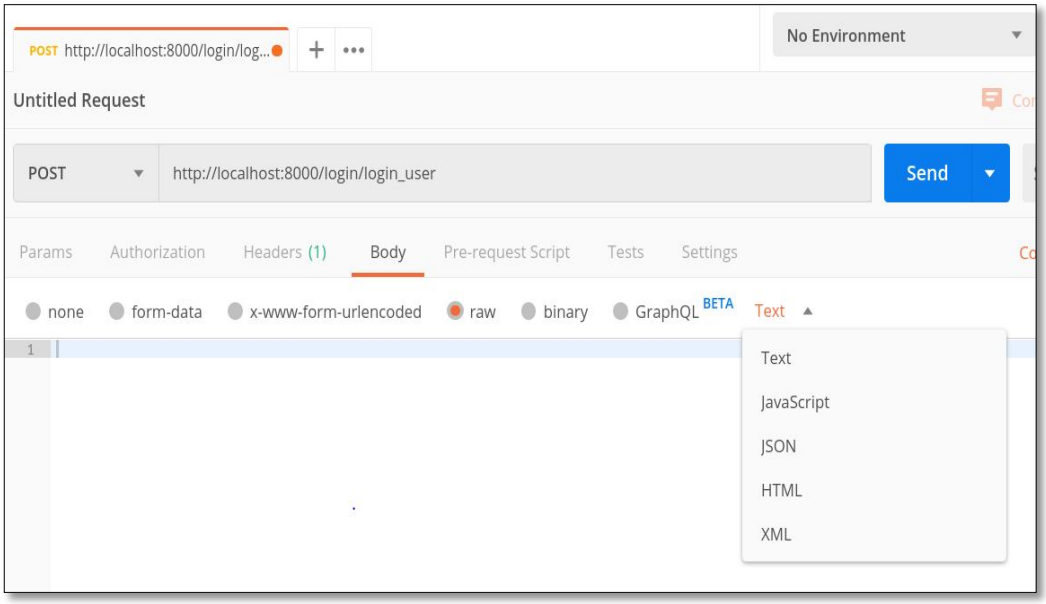
4. Signing windows executable using REST API with postman

4.1 Login

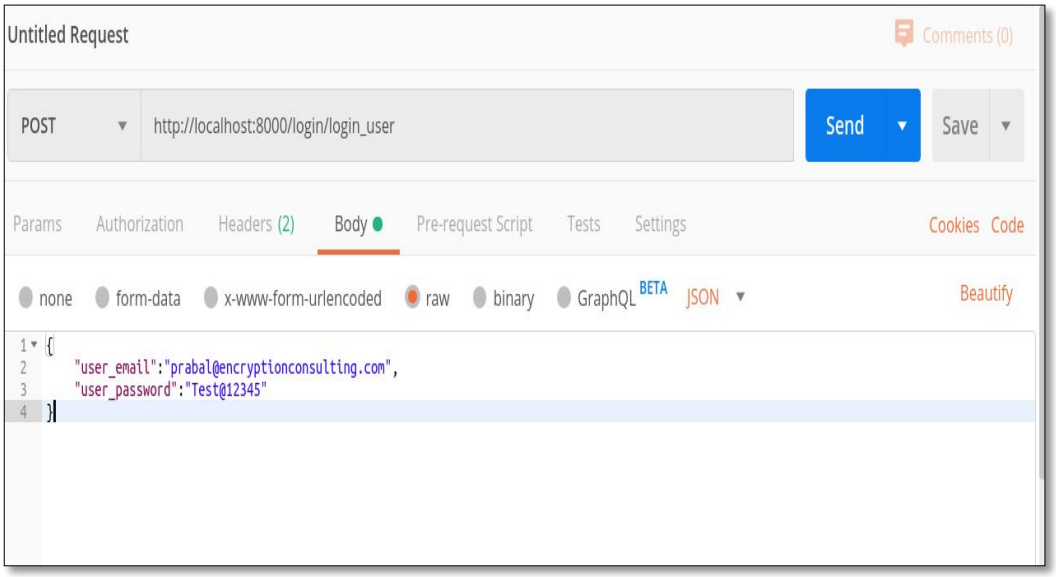
- Here the Customer Organization admin will enter the login credentials like user email address and password to login. On successful verification of login credentials, one token will be generated and assigned to the user. Concurrently, the user will also get one email containing authentication code. User will use the token and authentication code to complete 2 factor authentications for successful login.
- Enter login URL and Choose Method POST Screenshot



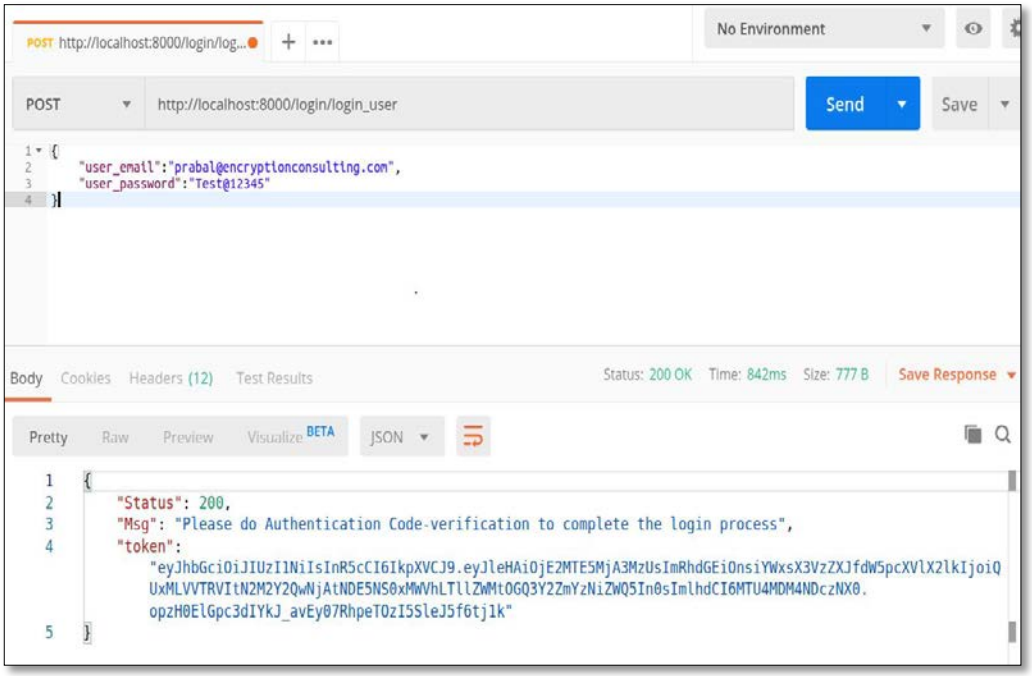
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Inside the body, enter the Customer organization login credentials as per shared screenshot format. Then Click Send Button

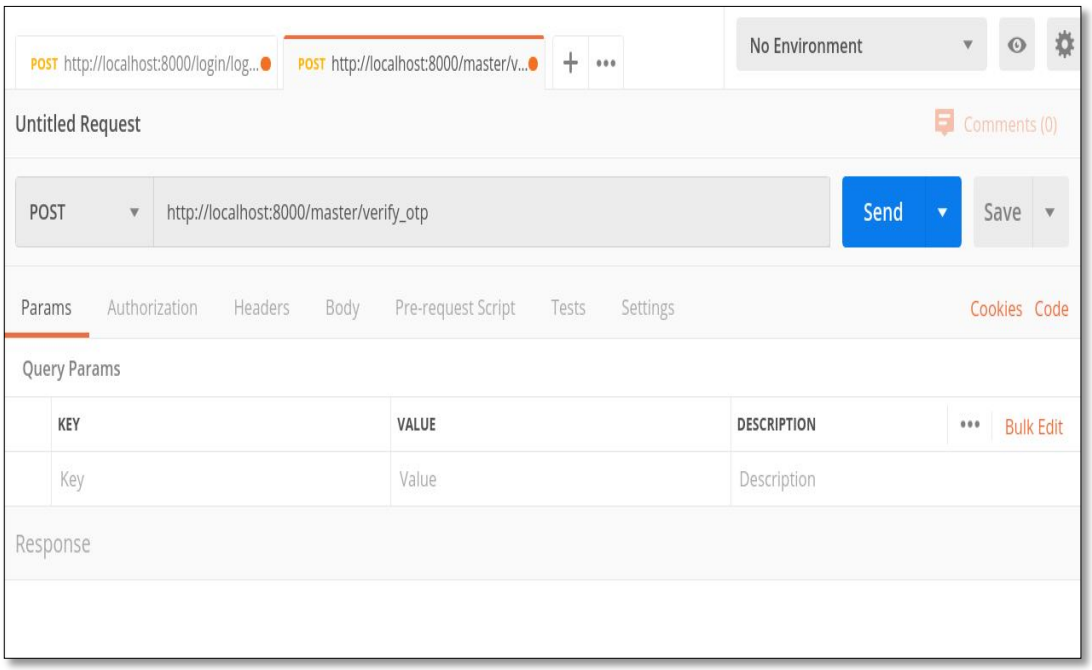


- On Successful login, token is generated.

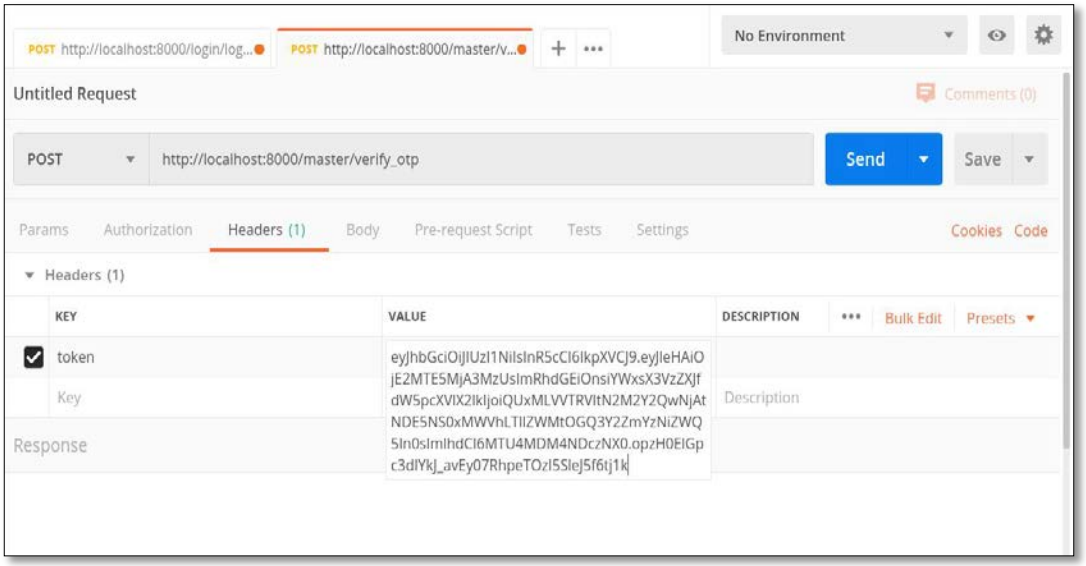


4.2 Verify authentication code to complete two factor authentications

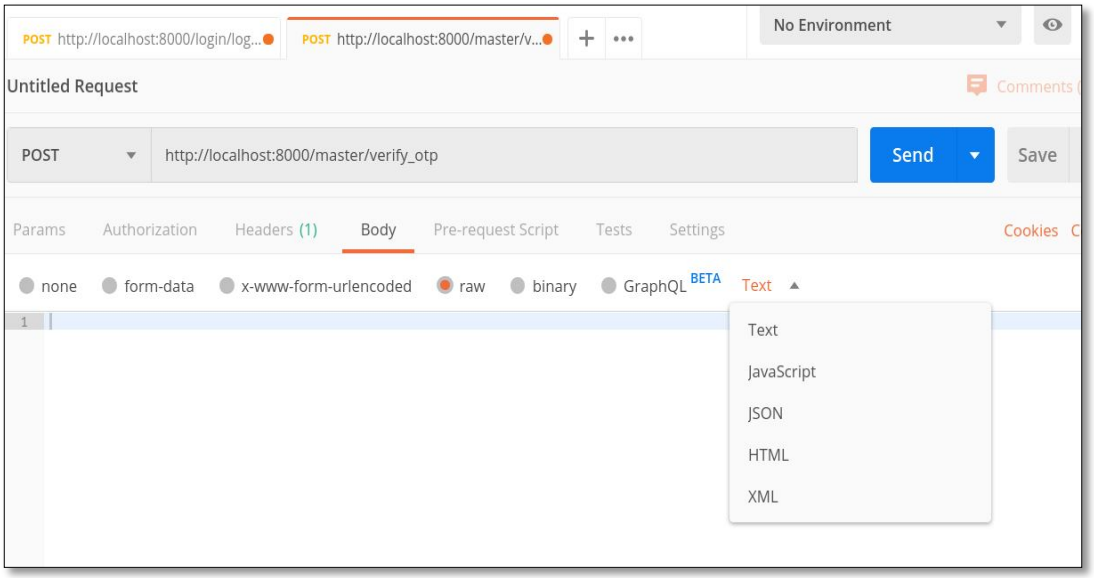
- Enter verify authentication code URL and Choose Method POST



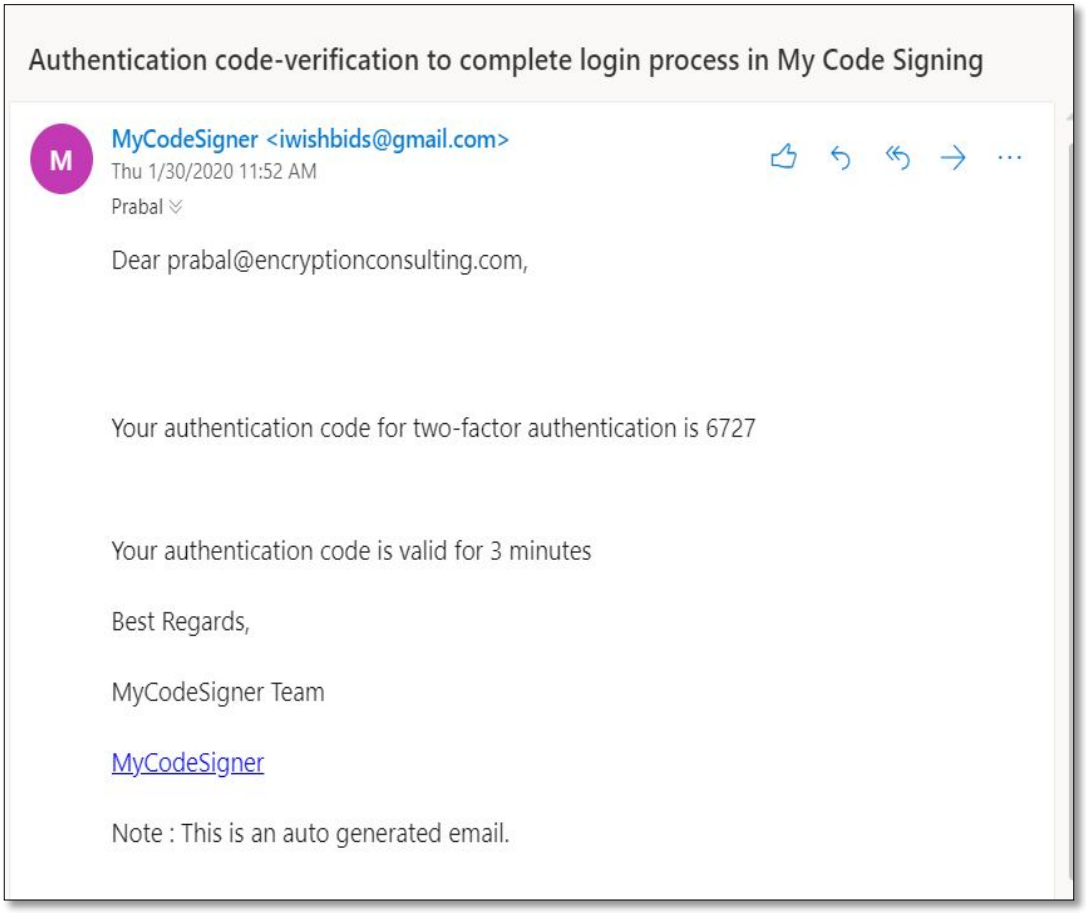
- Choose Headers option and Enter token in key data. Use the token value (generated on successful login)



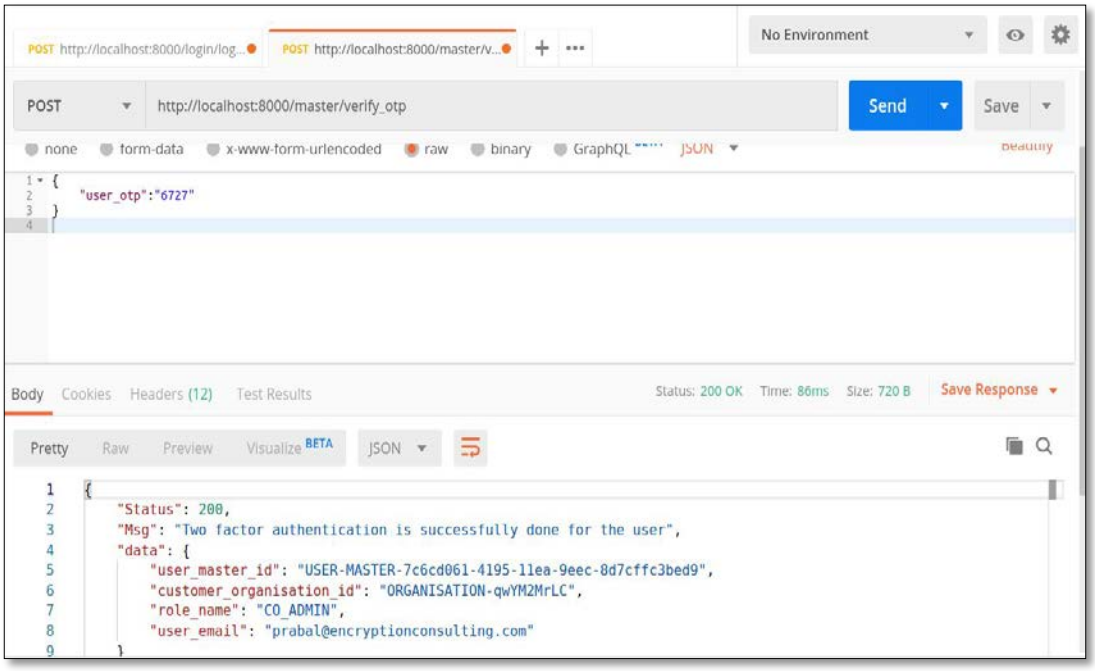
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Sample of email received with Authentication Code



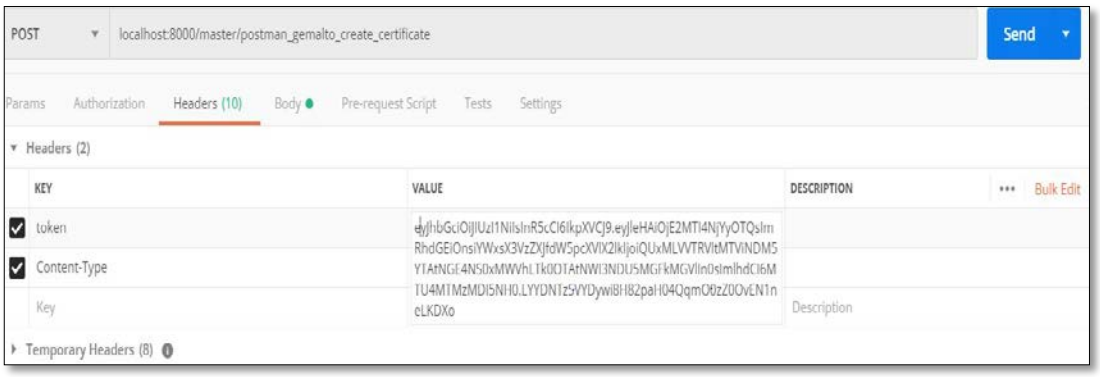
- Inside the body, enter the Authentication Code as per shared screenshot format. Then Click Send Button.



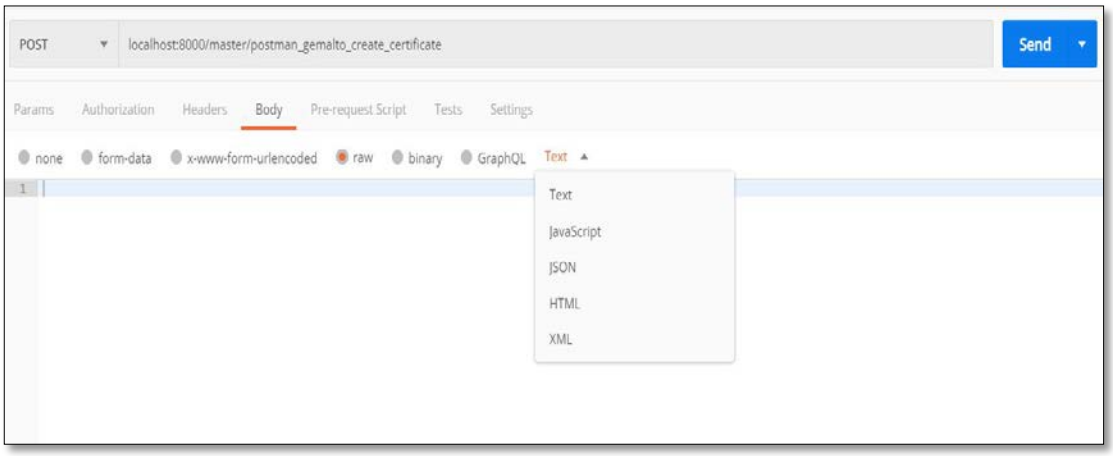
4.3 Create Certificate

- Enter creation of certificate URL and Choose Method POST

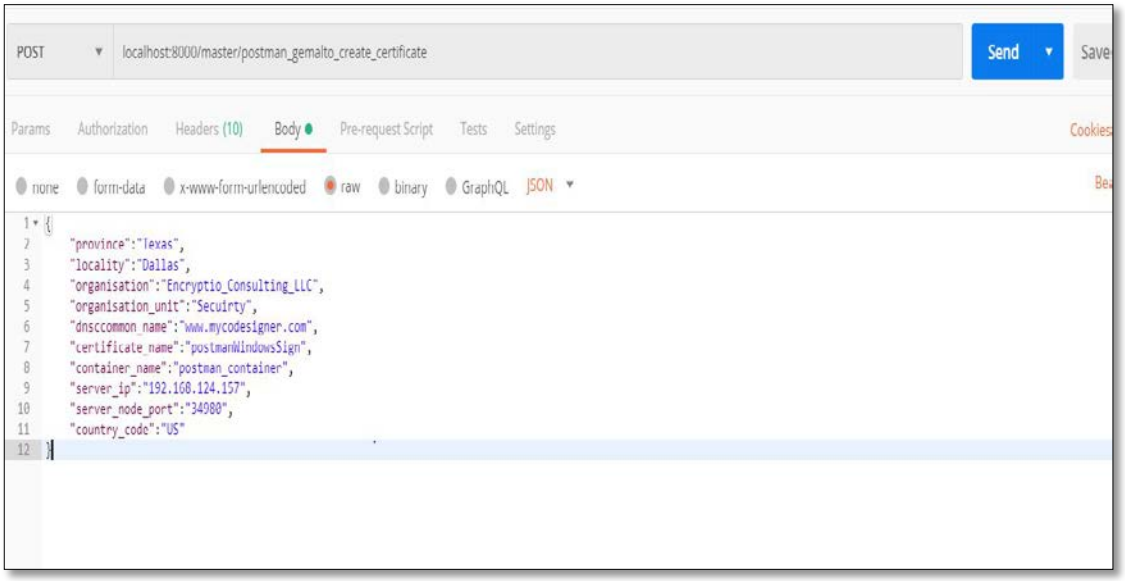
- Choose Headers option and Enter token in key data. Use the token value (generated on successful login)



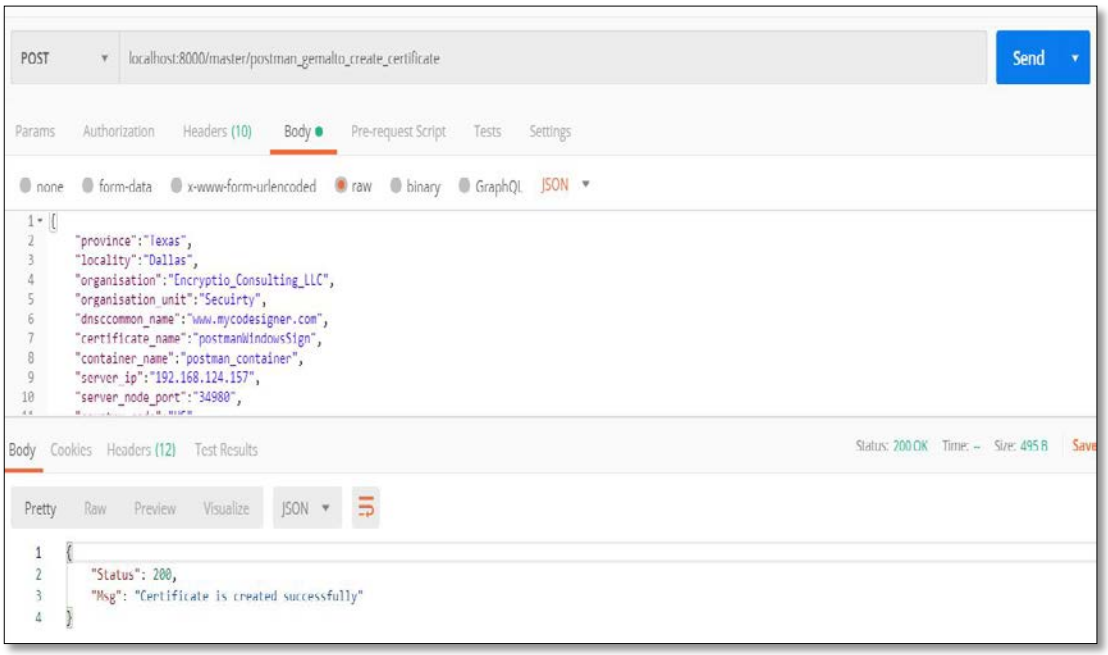
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Inside the body, enter the mandatory data to create certificate as per shared screenshot format. Then Click Send Button.

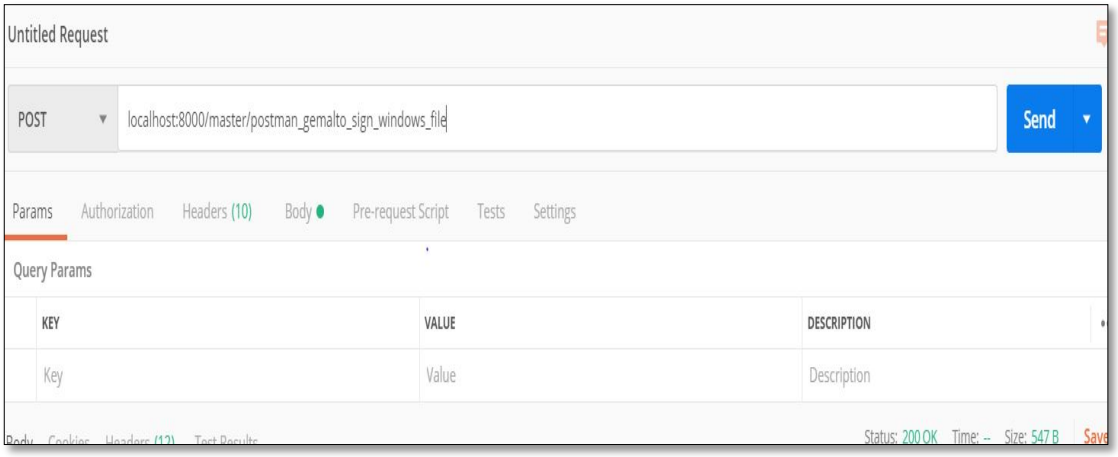


- On successful certificate creation, user will get the Status Response 200 like given in the screenshot.

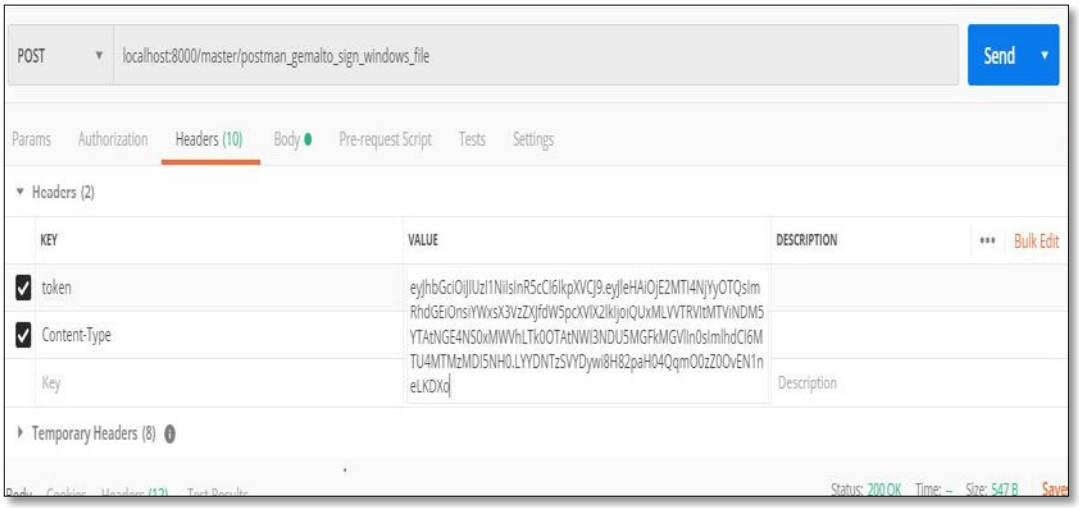


4.4 Sign the Windows Application

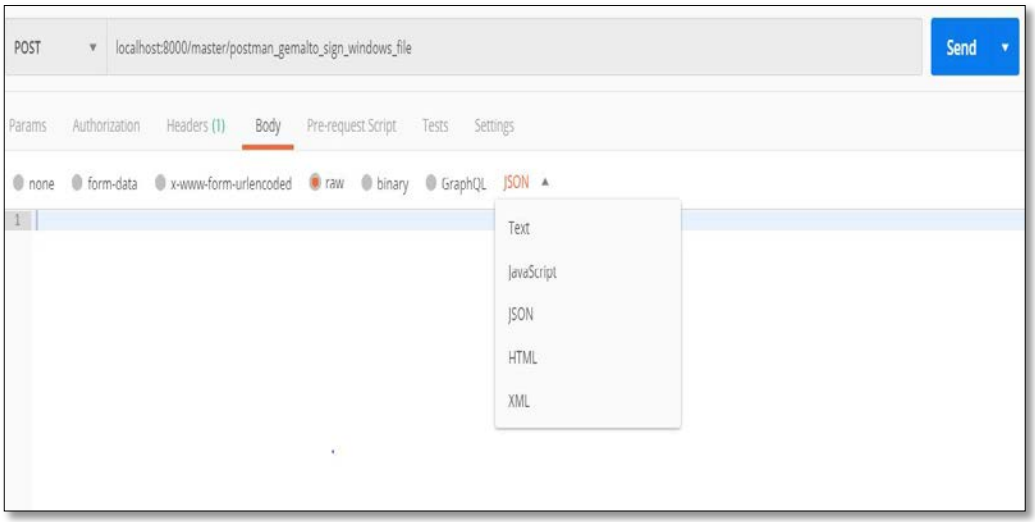
- Enter signing of windows application URL and Choose Method POST



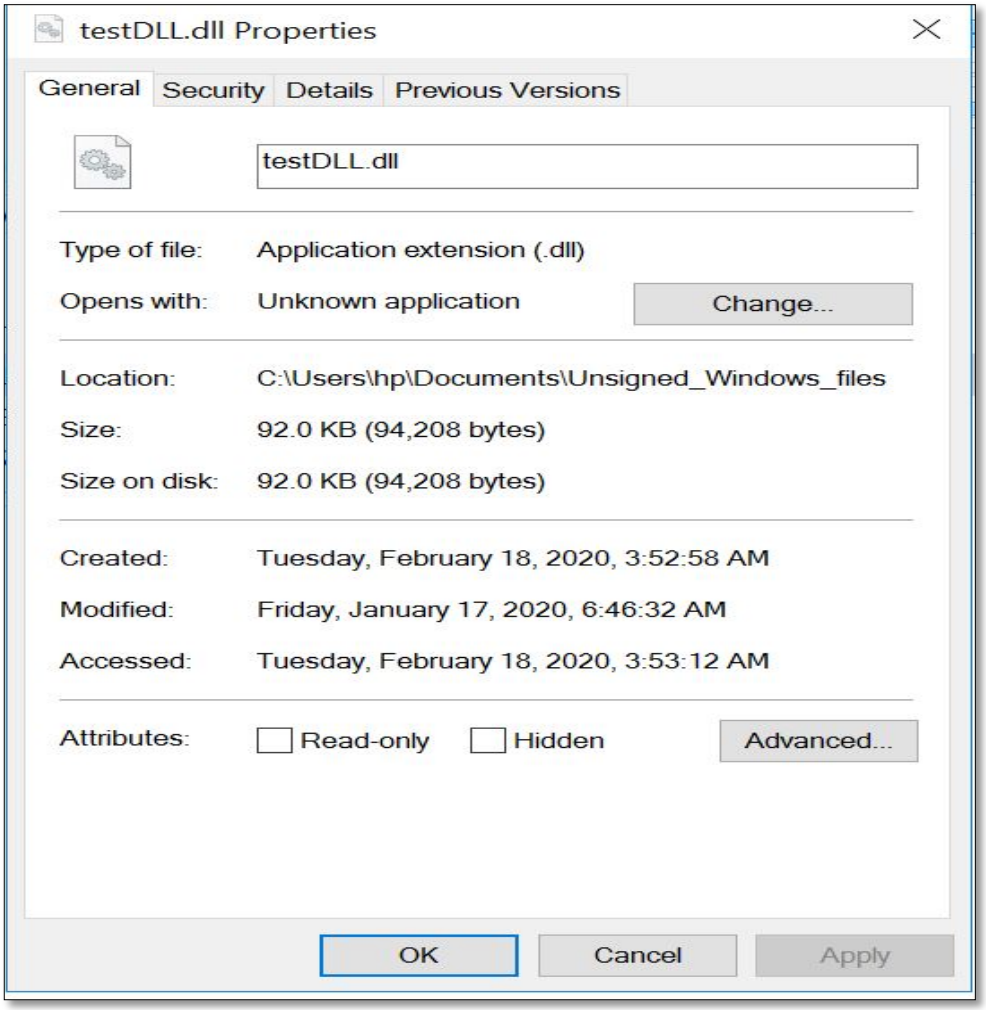
- Choose Headers option and Enter token in key data. Use the token value (generated on successful login)



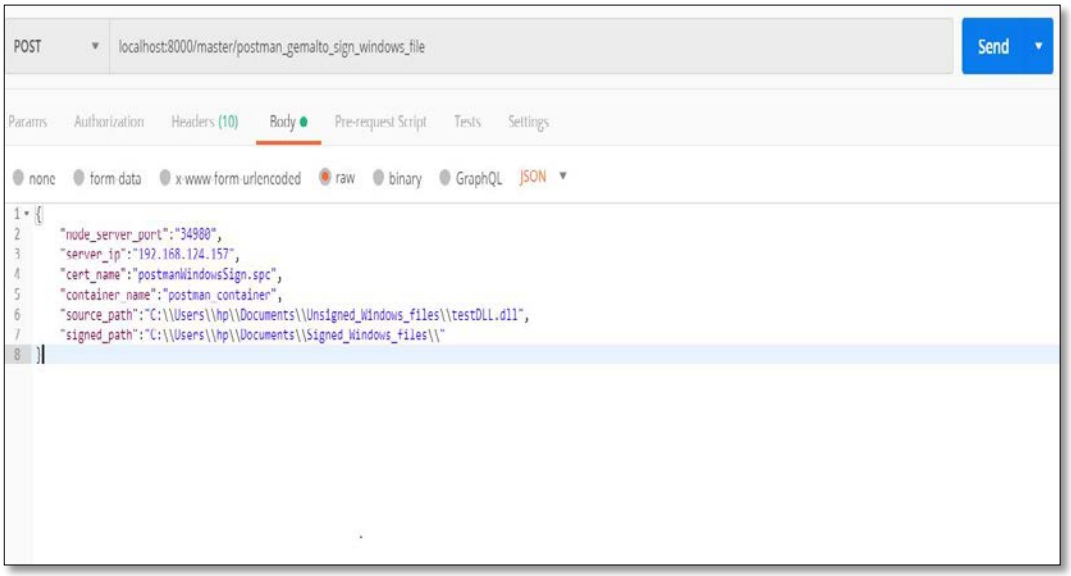
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



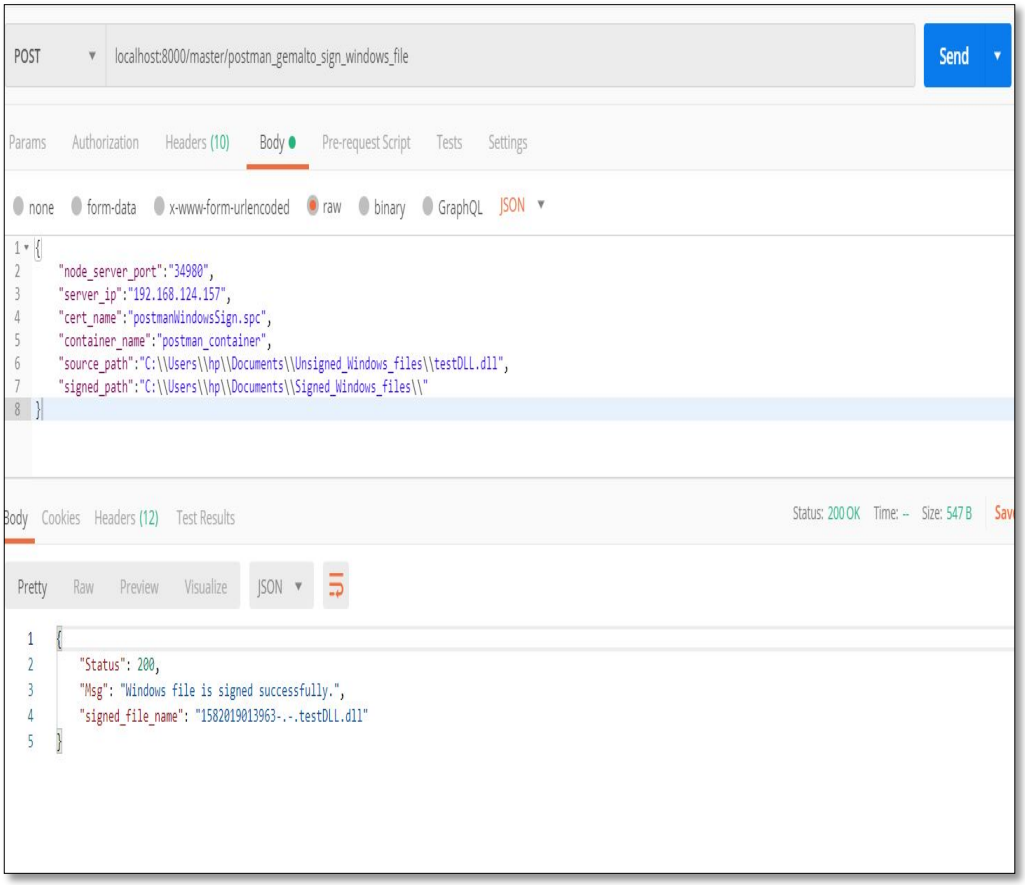
- Check Digital Signature of Application. Right Click on the application and choose properties option. In the tab, there should not be any Digital Signature tab.



- Inside the body, enter the mandatory data to digitally sign the application as per shared screenshot format. Then Click Send Button.

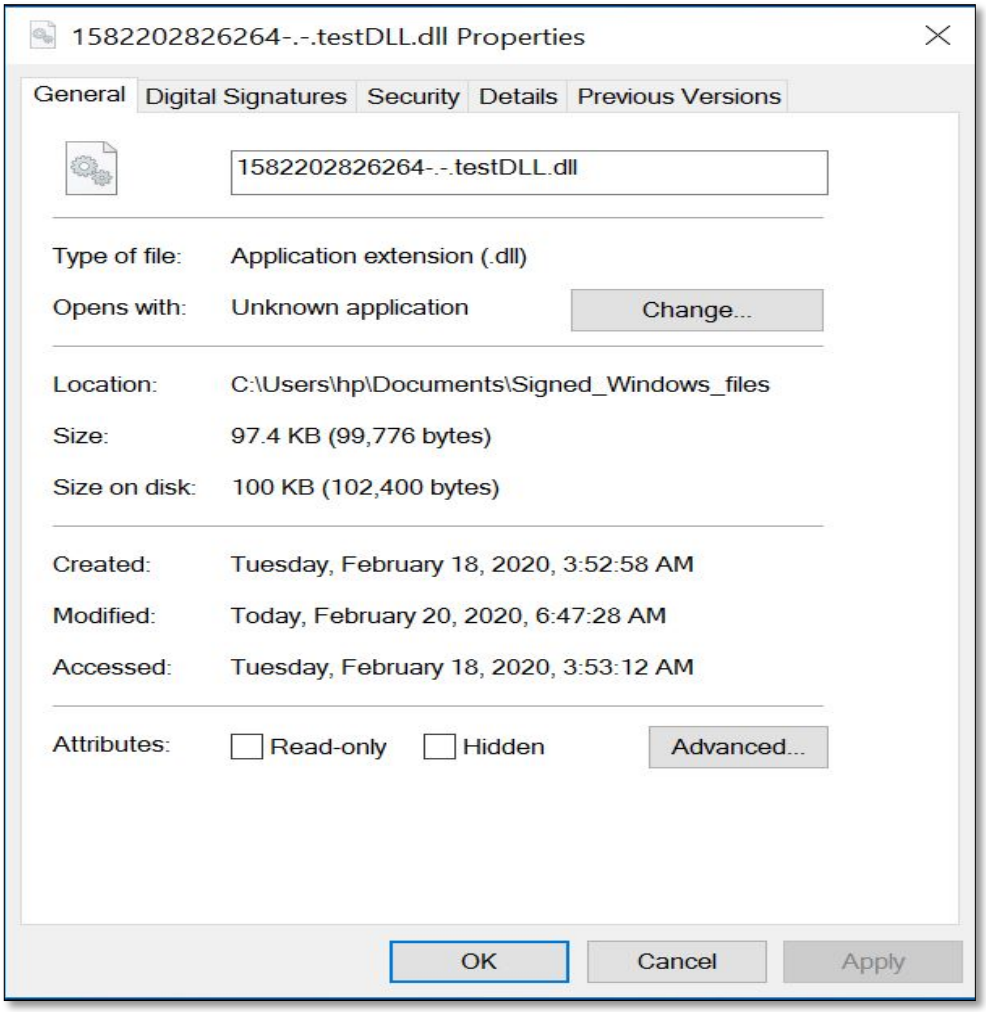


- On Successful signing, the response status will be 200.



4.5 Verify the Windows signed application

- Verify whether application is signed. Right click on the properties of the application. One Digital Signature tab is appeared.



5. SafeNet Luna HSM integration with Oracle JDK 8

Java code signing is used for signing Java applications for desktops, digitally signing .jar files and Netscape Object signing recognized by Java Runtime Environment (JRE).
The integration guide supports all the versions of Luna Client notably Luna 10.1 Universal Client, Luna 7.5 Client and Luna 7.2 Client with Windows Server 2016 Operating system.

5.1 Prerequisite

Please make sure the Luna Client is installed and configured successfully. To verify the successful installation and configuration, please follow the below steps.

1. Open Command prompt as administrator.
2. Go to the folder where lunacm is located using cd command.(Example of lunacm file path is C:\Program Files\SafeNet\LunaClient)
3. Run the command lunacm.exe (for Windows).

4. Output should look like the below screenshot. If not, please do Luna client configuration first.

5.2 Installation and configuration of Java

- Only supported version for using Luna SafeNet HSM is Java 8.Download Java 8 from the official website https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html as per your operating system (Windows Server 2016)
- Add "C:\Program Files\Java\jdk1.8.0_241\bin" in PATH variable (Environment).
- Update the Luna Provider in the java.security file in the folder C:\Program Files\Java\jdk1.8.0_241\jre\lib\security\java.security file using notepad. Add com.safenetinc.luna.provider.LunaProvider at the last of security.provider list. Output might be look like

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC
security.provider.10=sun.security.mscapi.SunMSCAPI
security.provider.11=com.safenetinc.luna.provider.LunaProvider
```

- Save the changed to the java.security file.
- Copy the LunaAPI.dll (Windows) and the LunaProvider.jar files from the C:\Program Files\SafeNet\LunaClient\JSP\lib\jsp\lib to the Java extension folder located atC:\Program Files\Java\jdk1.8.0_241\jre\lib\ext.
- Create a file called lunastore where the **project(**C:\Users\hp\Documents\backend**)** is running and add the following line:

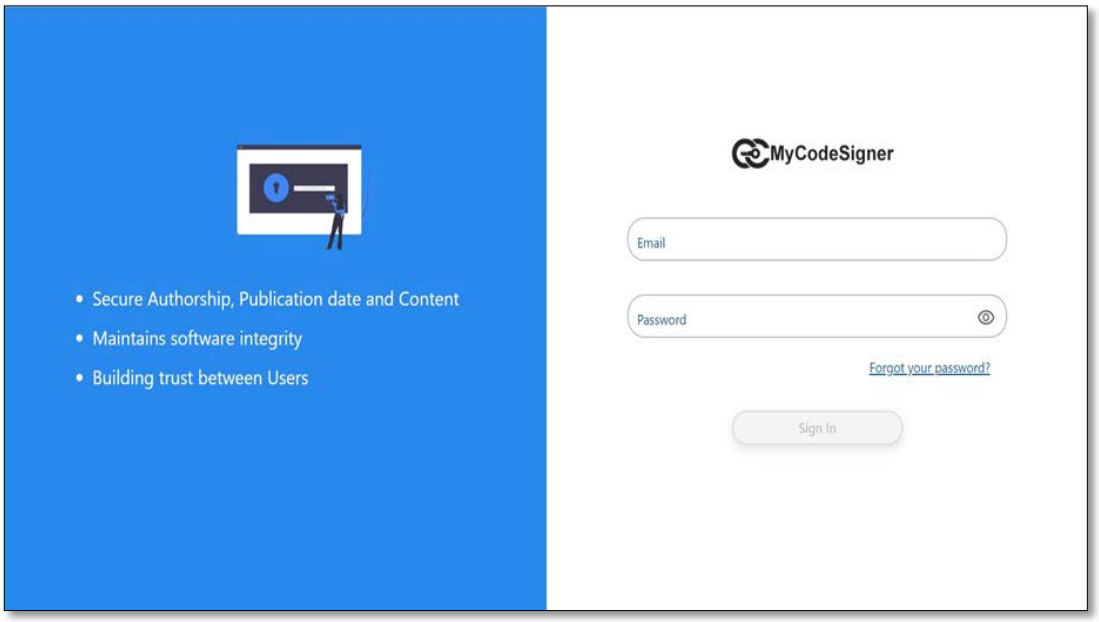
  tokenlabel:<service_label>
  (Please replace the <service_label> with the partition name like elabhsm1par9.Example: tokenlabel:elabhsm1par9)

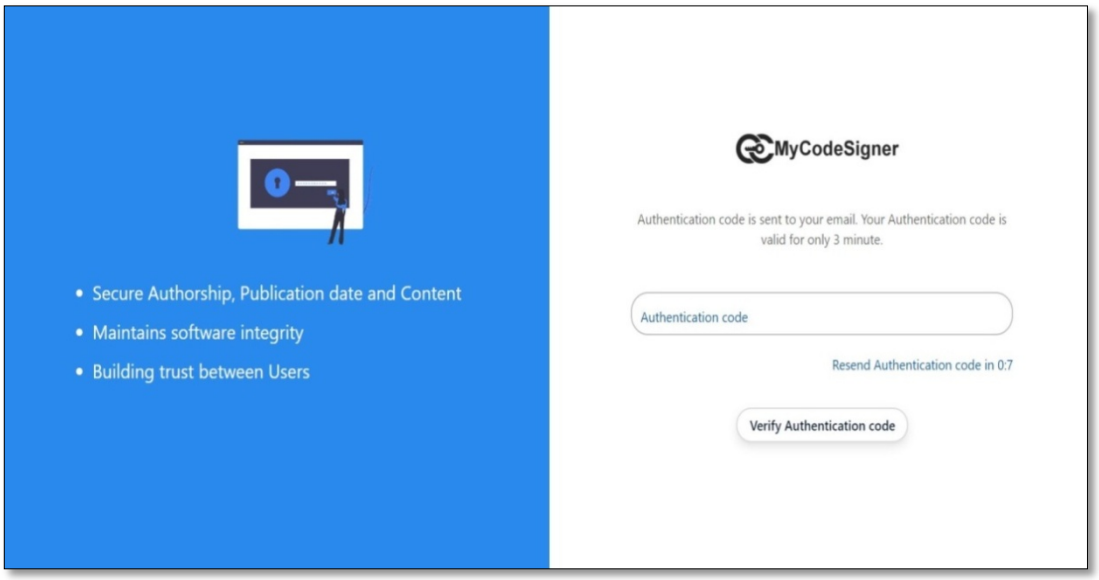
Installation and Configuration of JDK is completed.

6. Signing JAR files using MyCodesigner application

6.1 Login

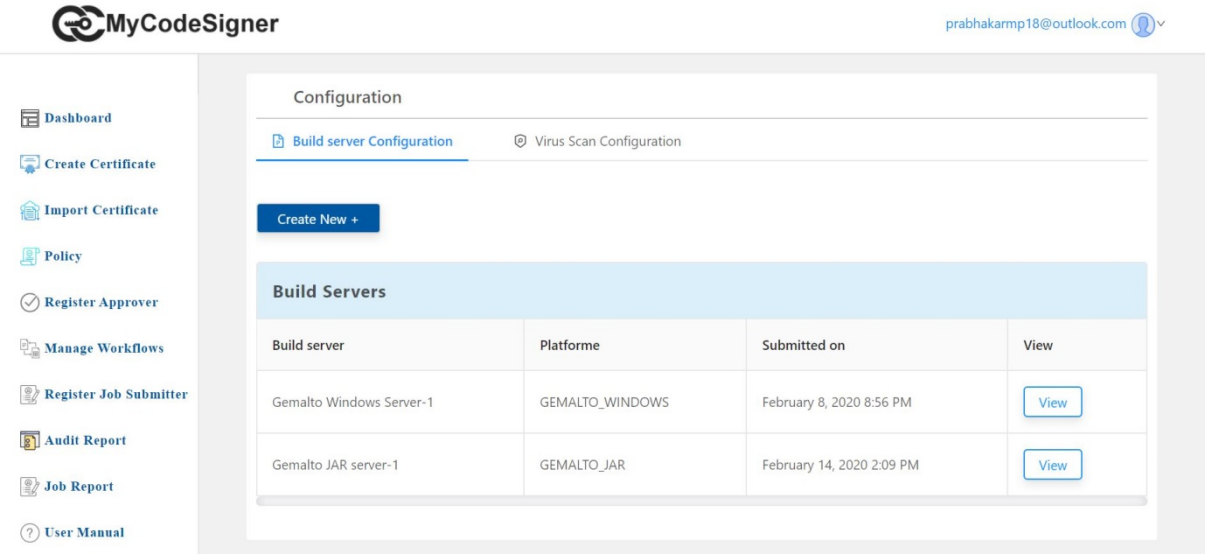
- Go to MyCodesigner virtual machine, open browser and type localhost in the URL. It will take into the login screen.
- Enter CO Admin credentials and do 2-factor-authentication.
- Screenshot of login and 2-factor authentication

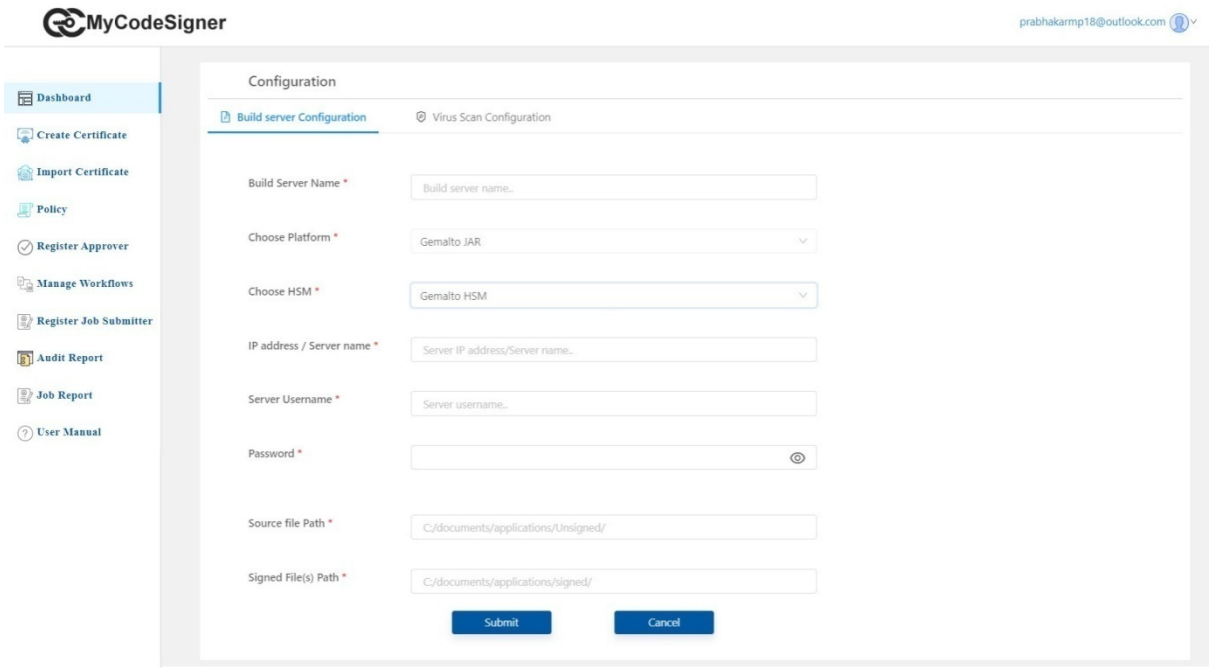


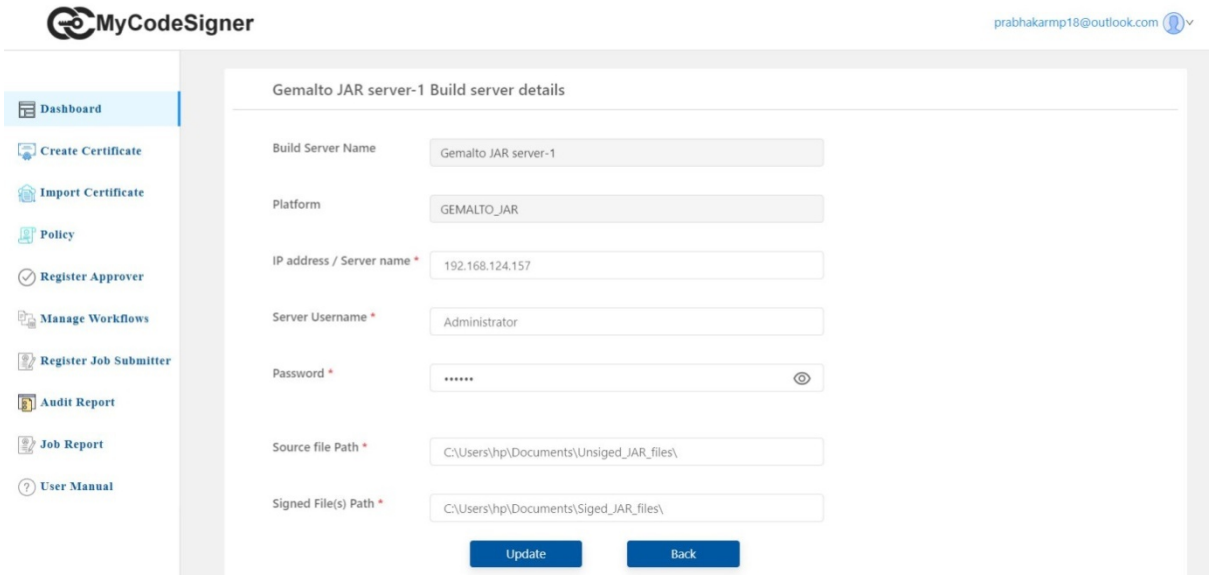


6.2 Create build server

- After login into CO Admin, it will take you to the dashboard and click on Configuration option to create build server.
- On Build server configuration tab, click on Create new button to create new build server.
- Enter build server name, choose platform as GEMALTO JAR from drop-down, choose HSM as GEMALTO HSM from drop-down, enter the other required details and click submit button to create a new build server.
- To view build server configuration, click view button on the table to view particular build server.
- To update build server configuration, update the details except Build server name and Platform, fill the other changes want to update and click Update button to update the Build server details.
- Screenshots of the build server

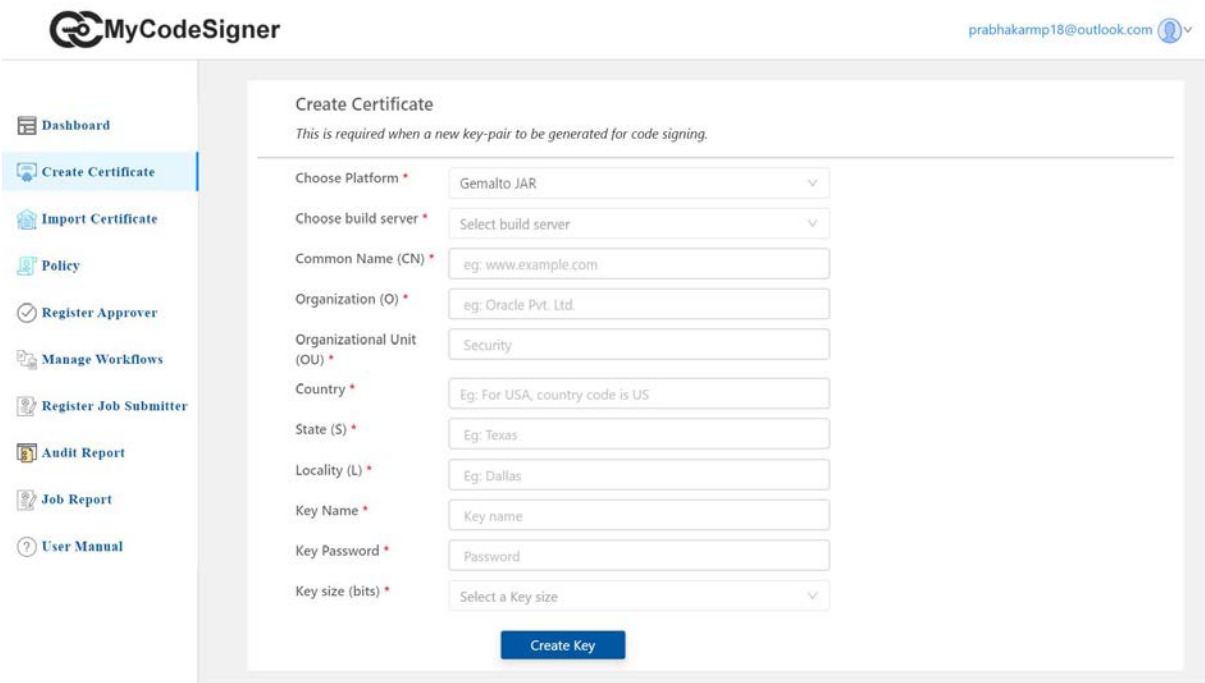






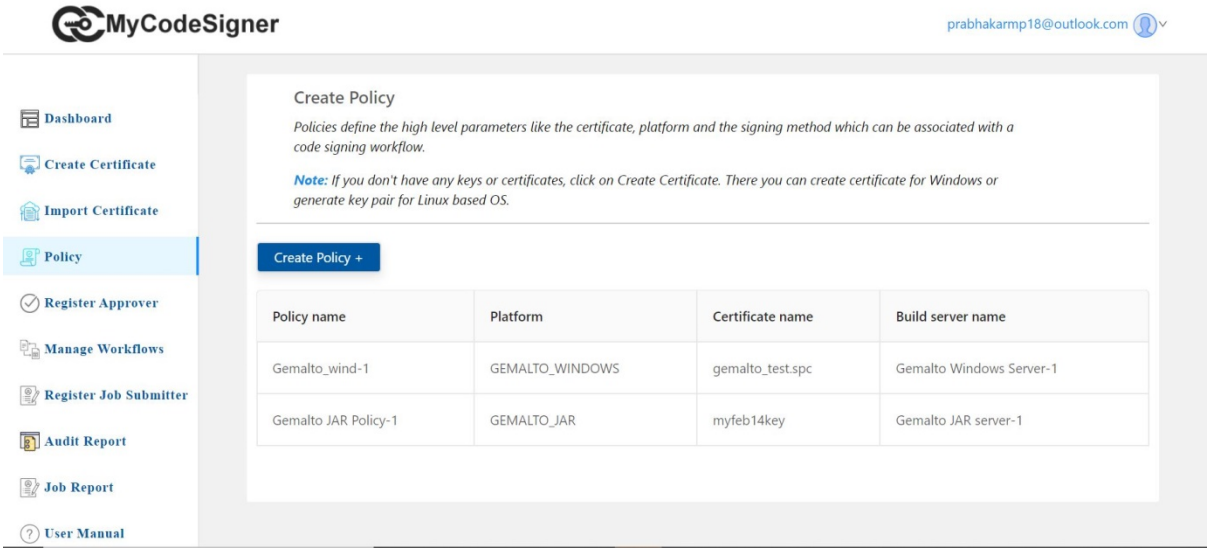
6.3 Create Certificate

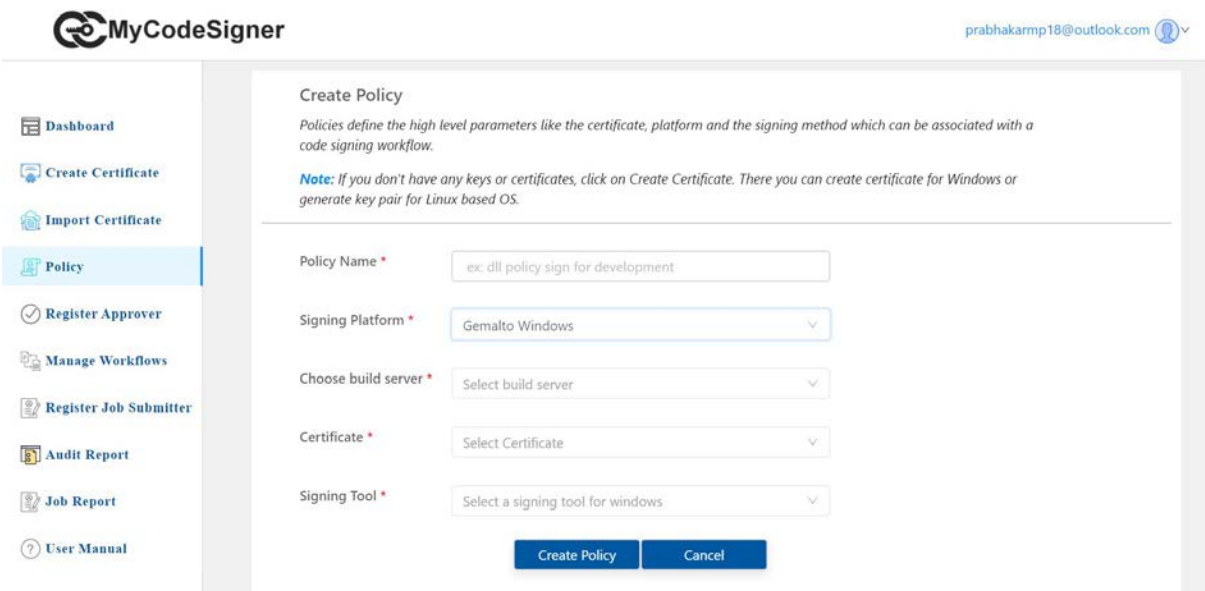
- Create certificate is a self-signed certificate to sign the jar file type.
- Click on Create certificate in Dashboard.
- Choose Platform as GEMALTO JAR and fill the all other details.
- Click Create button to create a certificate.
- Screenshots of the create certificate for JAR



6.4 Create Policy

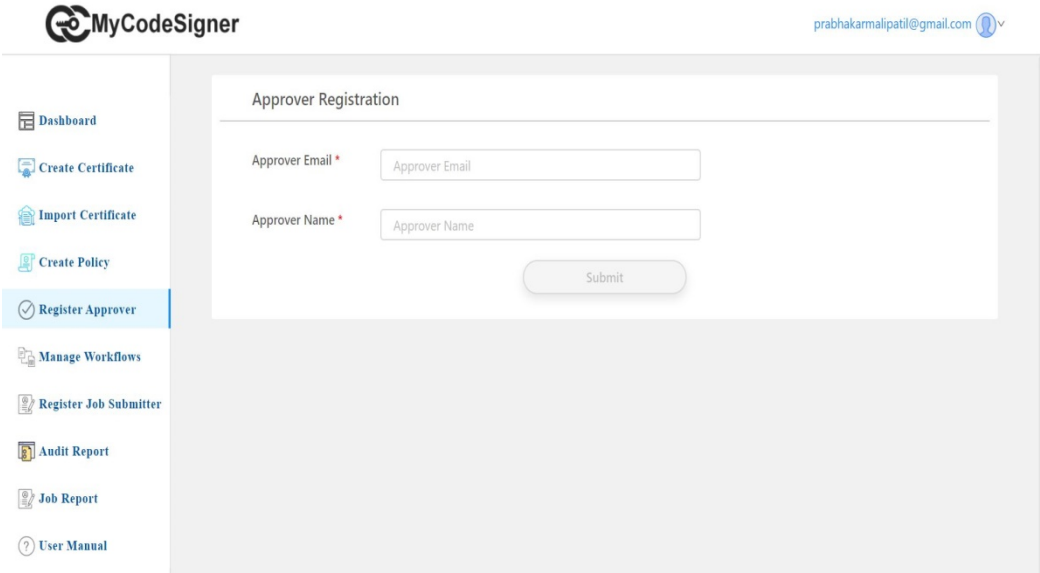
- Policies define the high level parameters like the certificate, platform and the signing method which can be associated with a code signing workflow.
- To create a policy, click on Create policy menu in dashboard.
- Enter the required details to create policy and click on Create policy button.
- Screenshots of the create policy





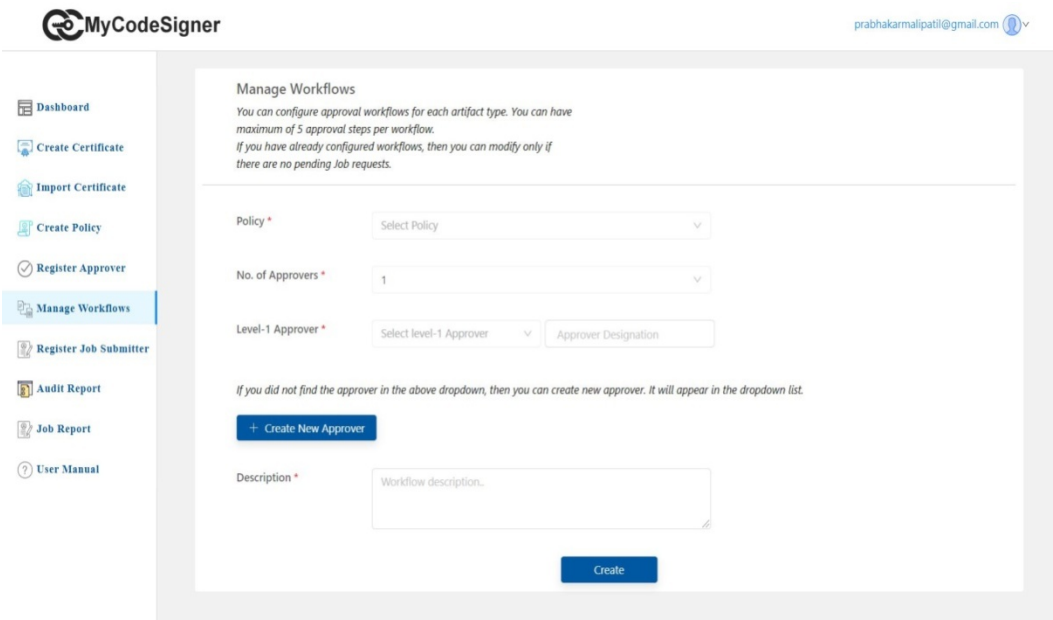
6.4 Register Job approver

- Click on Register approver menu in the dashboard.
- Enter Approver email and approver name.
- Click on Submit button to register job approver.
- Screenshot of register job approver



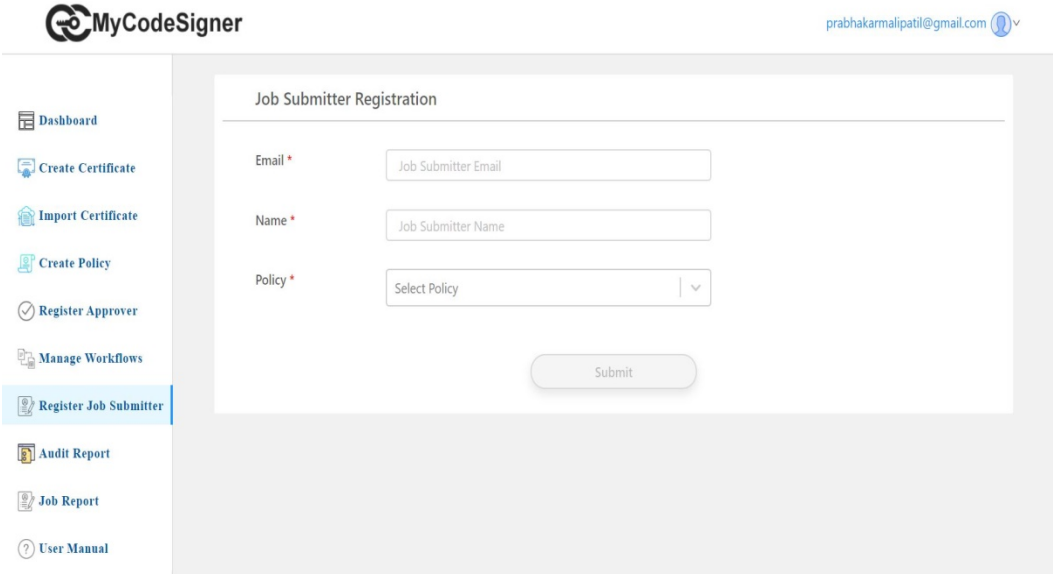
6.5 Manage Workflow

- Click on Manage workflow menu in the dashboard.
- Choose policy from the drop-down, select no. of approver, workflow description and each level approver and their designation.
- Click Create button to create new workflow
- Screenshot of Manage Workflow



6.6 Register Job submitter

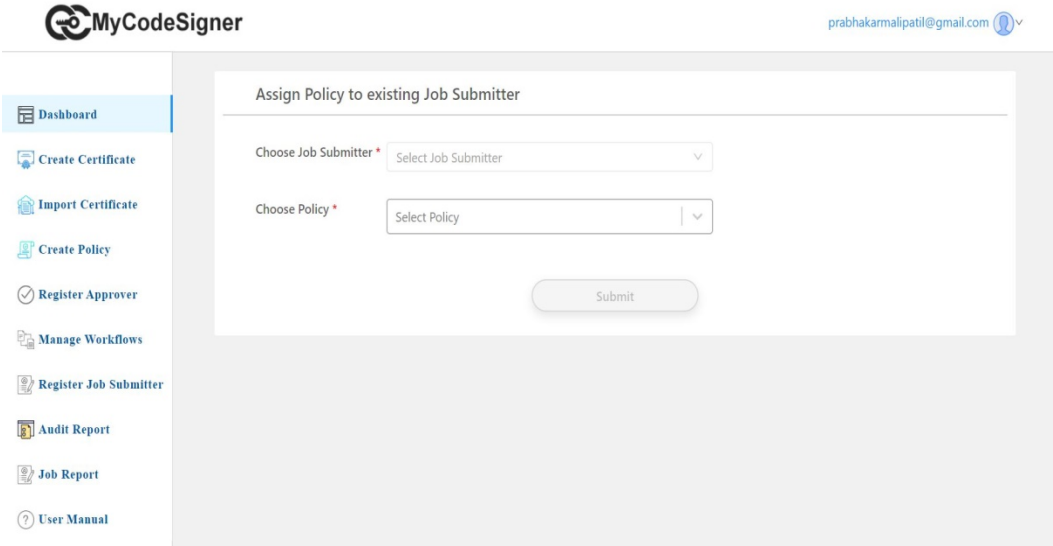
- Click on the Register Job submitter menu from the dashboard.
- Enter the job submitter email, name and select policy from the drop-down.
- Click on Submit button to register new Job submitter.
- Screenshot of Register Job submitter



6.7 Manage Job submitter

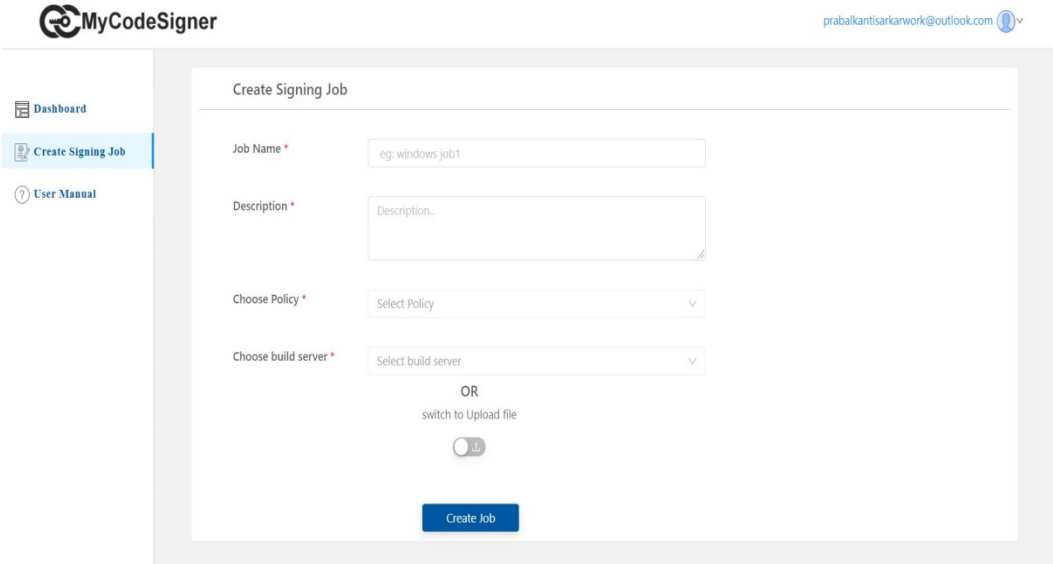
- Click on the Manage Job submitter menu from the dashboard.
- Choose Job submitter from the drop-down and select policy from the drop-down.

- Click on Submit button to assign new policy to the existing job submitter.
- Screenshot of Manage Job submitter



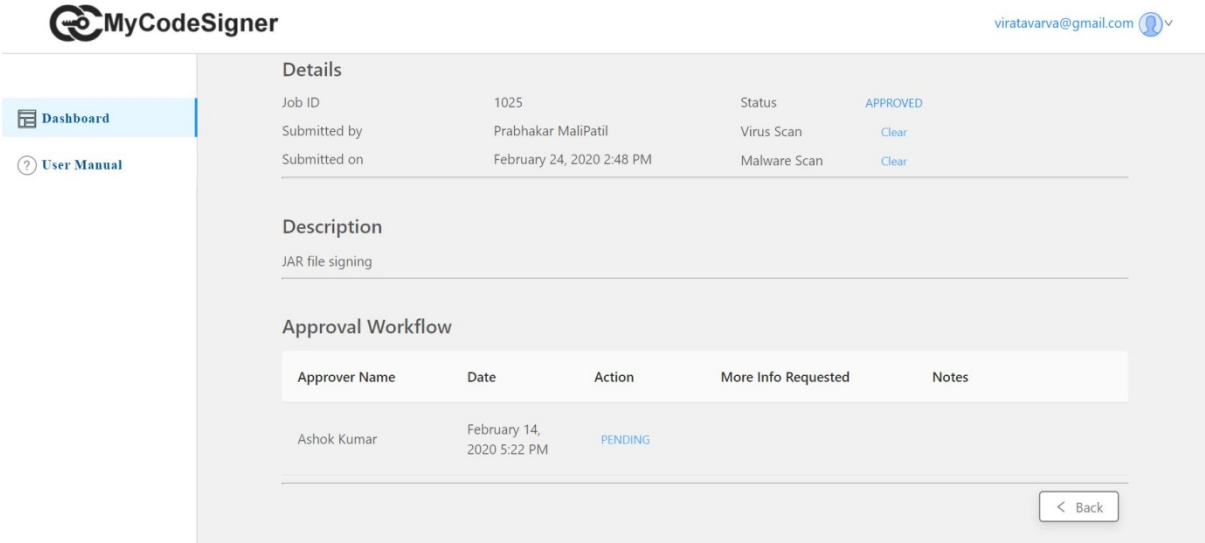
6.8 Create a signing job

- Login as a job submitter and click on Create signing job menu in the dashboard.
- Enter the job name, description, choose policy from the drop-down, and select the build server from the drop-down.
- Enter the file name to sign the relative path.
- Click on Create Job button to create a job.
- Screenshot of Create a signing job

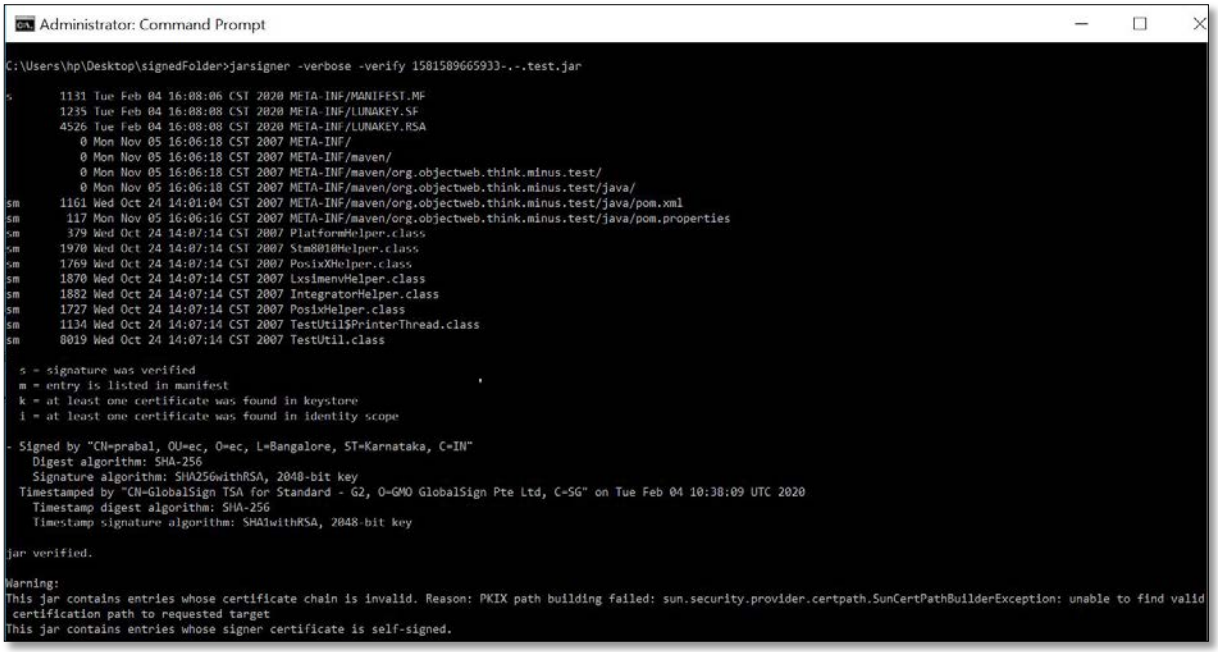


6.9 Action on job (Approve, Reject and more info)

- Login as a job approver and click on Action button in the Signing requests waiting for your approval table of particular job.
- You can add notes to the particular job.
- To approve a job, click on Approve button in the approval workflow table, once the job is approved it will go for signing.
- To reject a job, click on Reject button in the approval workflow table.
- To request more information about a job, click on more info requested button in the approval workflow table.
- Screenshot of Approver



6.10 Verify JAR file after signing



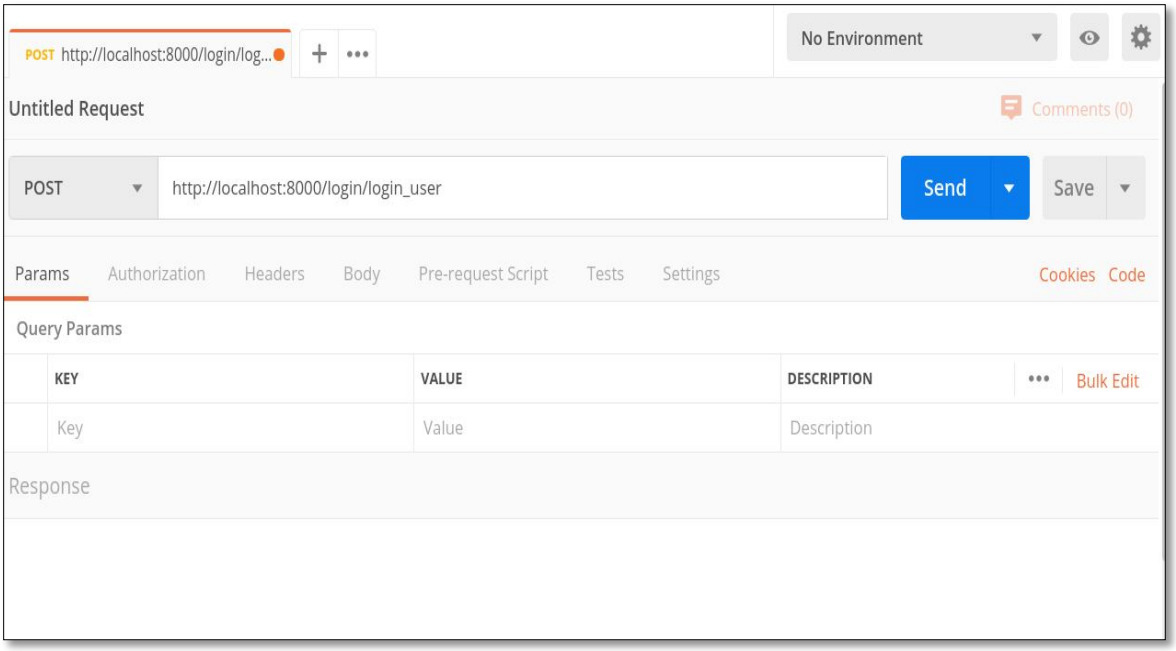
```
Administrator: Command Prompt

C:\Users\hp\Desktop\signedFolder>jarsigner -verbose -verify 1581589665933-.-.test.jar

s       1131 Tue Feb 04 16:08:06 CST 2020 META-INF/MANIFEST.MF
        1235 Tue Feb 04 16:08:08 CST 2020 META-INF/LUNAKEY.SF
        4526 Tue Feb 04 16:08:08 CST 2020 META-INF/LUNAKEY.RSA
           0 Mon Nov 05 16:06:18 CST 2007 META-INF/
           0 Mon Nov 05 16:06:18 CST 2007 META-INF/maven/
           0 Mon Nov 05 16:06:18 CST 2007 META-INF/maven/org.objectweb.think.minus.test/
           0 Mon Nov 05 16:06:18 CST 2007 META-INF/maven/org.objectweb.think.minus.test/java/
sm      1161 Wed Oct 24 14:01:04 CST 2007 META-INF/maven/org.objectweb.think.minus.test/java/pom.xml
sm       117 Mon Nov 05 16:06:16 CST 2007 META-INF/maven/org.objectweb.think.minus.test/java/pom.properties
sm       379 Wed Oct 24 14:07:14 CST 2007 PlatformHelper.class
sm      1970 Wed Oct 24 14:07:14 CST 2007 Stm8010Helper.class
sm      1769 Wed Oct 24 14:07:14 CST 2007 PosixXHelper.class
sm      1870 Wed Oct 24 14:07:14 CST 2007 LxsimenvHelper.class
sm      1882 Wed Oct 24 14:07:14 CST 2007 IntegratorHelper.class
sm      1727 Wed Oct 24 14:07:14 CST 2007 PosixHelper.class
sm      1134 Wed Oct 24 14:07:14 CST 2007 TestUtil$PrinterThread.class
sm      8019 Wed Oct 24 14:07:14 CST 2007 TestUtil.class

  s = signature was verified
  m = entry is listed in manifest
  k = at least one certificate was found in keystore
  i = at least one certificate was found in identity scope

- Signed by "CN=prabal, OU=ec, O=ec, L=Bangalore, ST=Karnataka, C=IN"
    Digest algorithm: SHA-256
    Signature algorithm: SHA256withRSA, 2048-bit key
  Timestamped by "CN=GlobalSign TSA for Standard - G2, O=GMO GlobalSign Pte Ltd, C=SG" on Tue Feb 04 10:38:09 UTC 2020
    Timestamp digest algorithm: SHA-256
    Timestamp signature algorithm: SHA1withRSA, 2048-bit key

jar verified.

Warning:
This jar contains entries whose certificate chain is invalid. Reason: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
 certification path to requested target
This jar contains entries whose signer certificate is self-signed.
```

7. Signing JAR files with REST API using postman

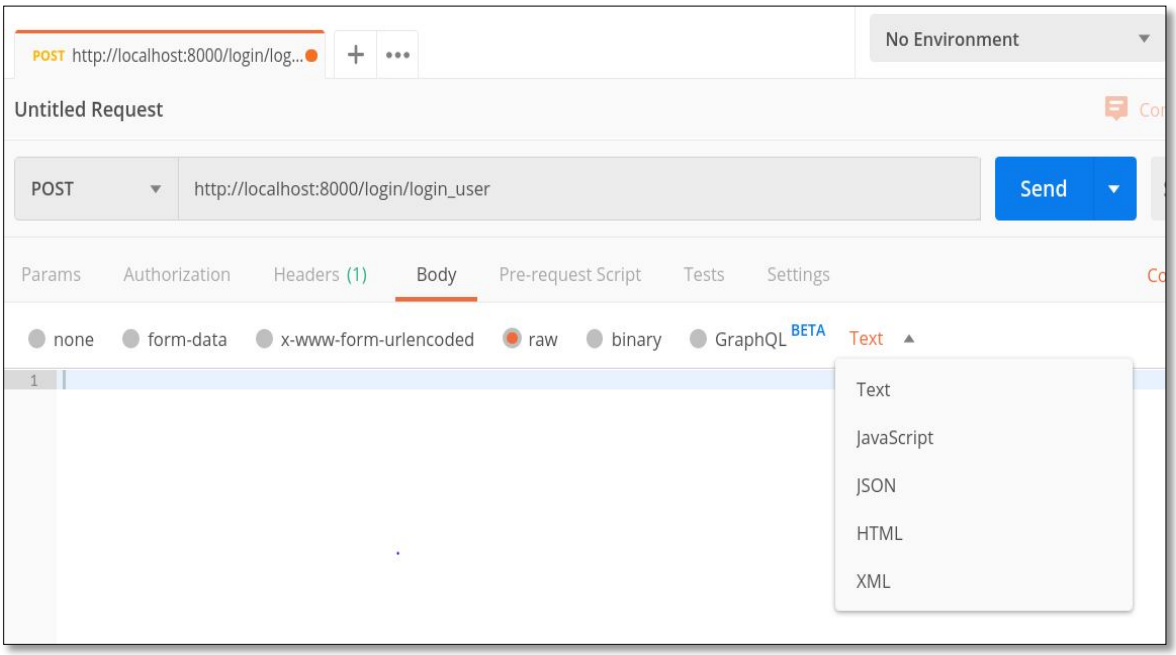
7.1 Login

Here the Customer Organization admin will enter the login credentials like user email address and password to login. On successful verification of login credentials, one token will be generated and assigned to the user. Concurrently, the user will also get one email containing authentication code. User will use the token and authentication code to complete 2 factor authentications for successful login.

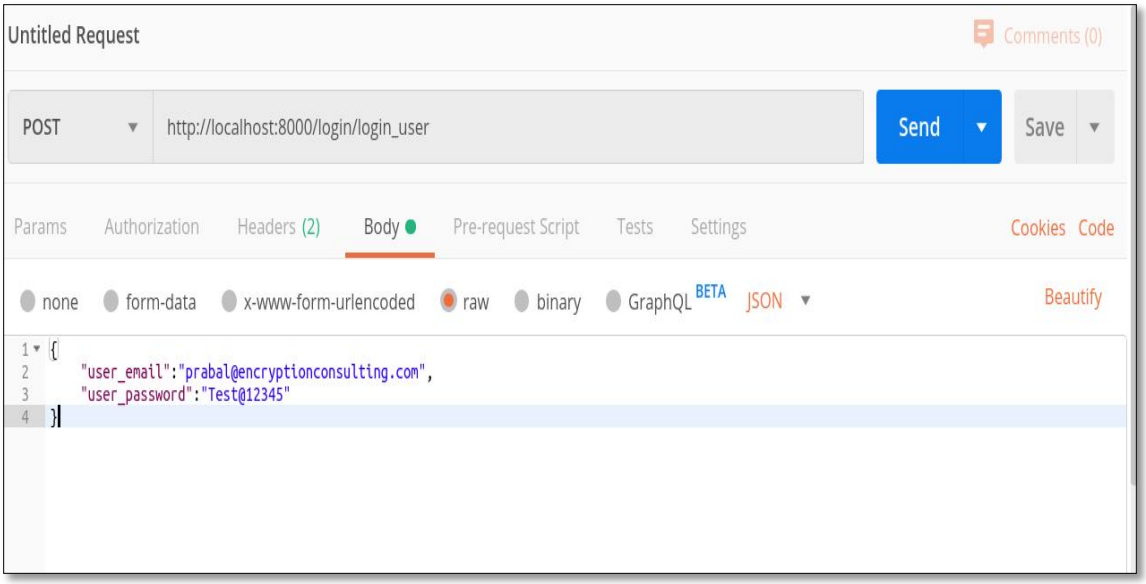
- Enter the login URL and Choose Method POST



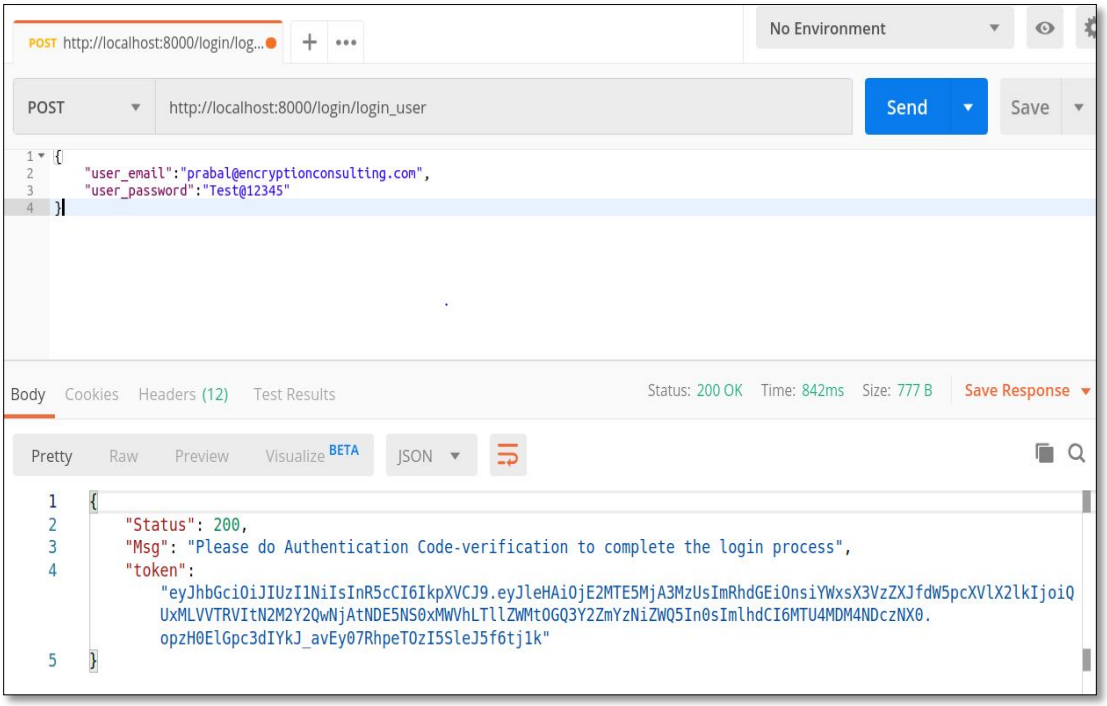
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Inside the body, enter the Customer organization login credentials as per shared screenshot format. Then Click Send Button.

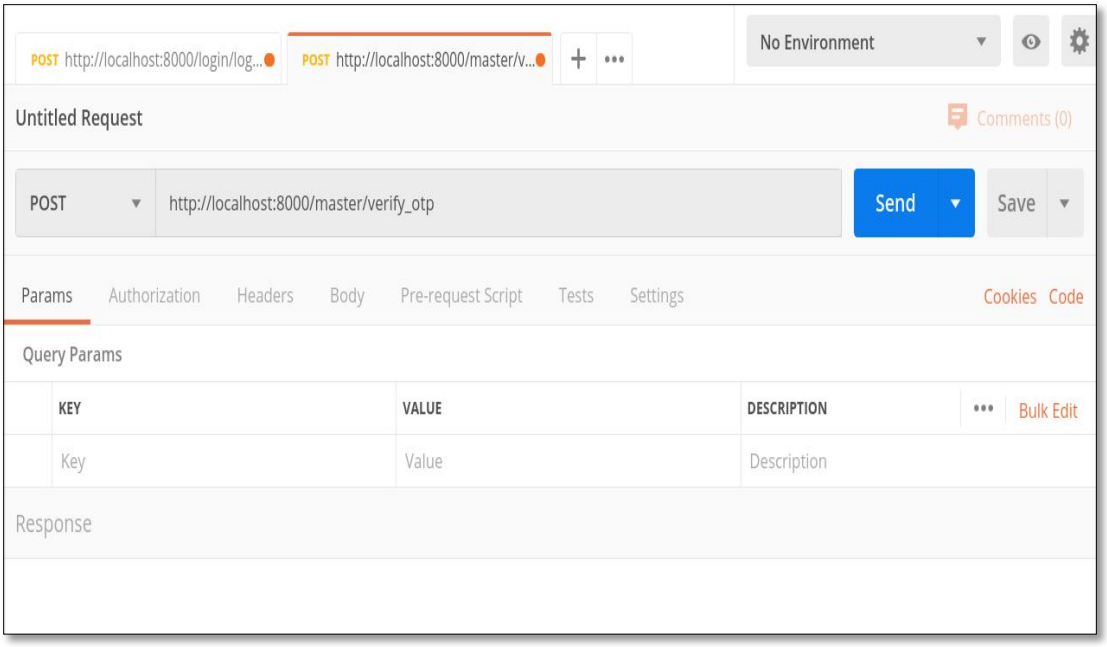


- On Successful login, token is generated.

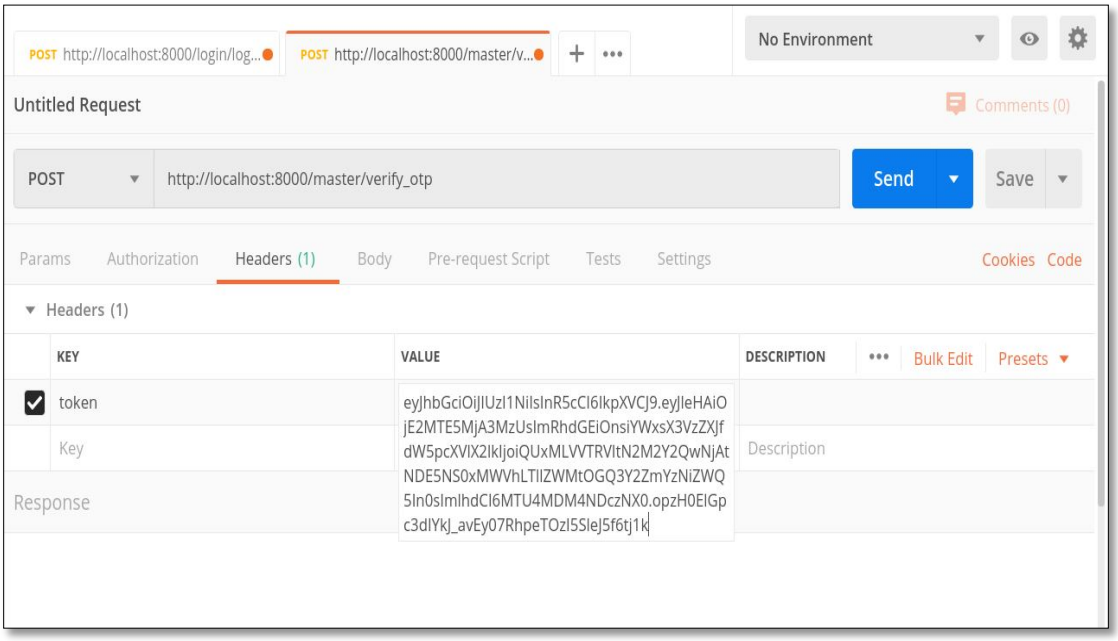


7.2 Verify authentication code to complete two factor authentications

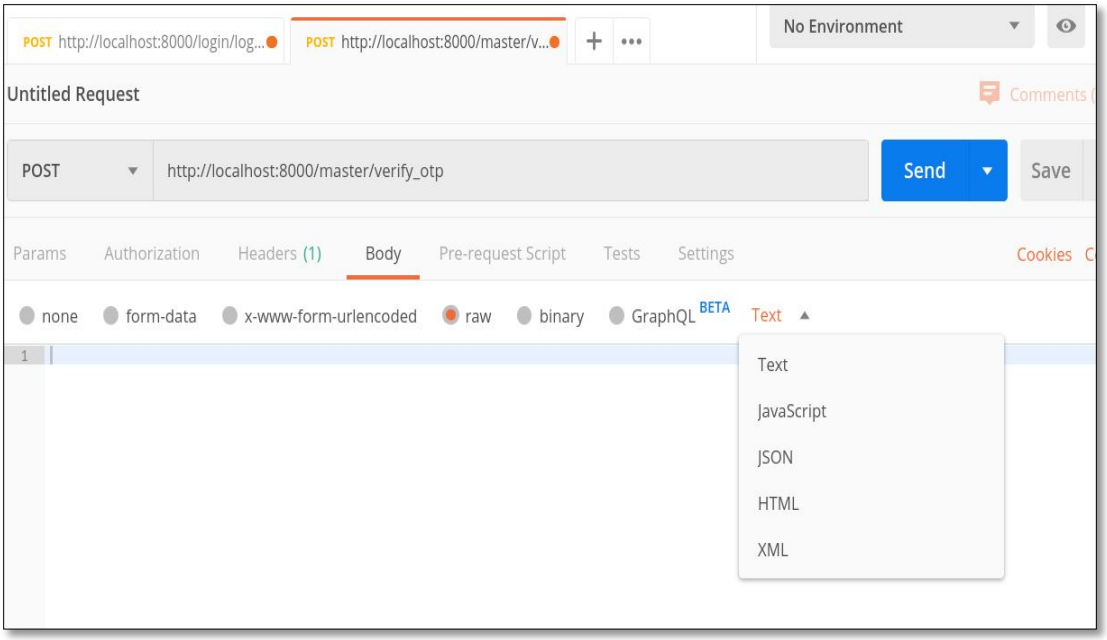
- Enter the verify authentication code URL and Choose Method POST



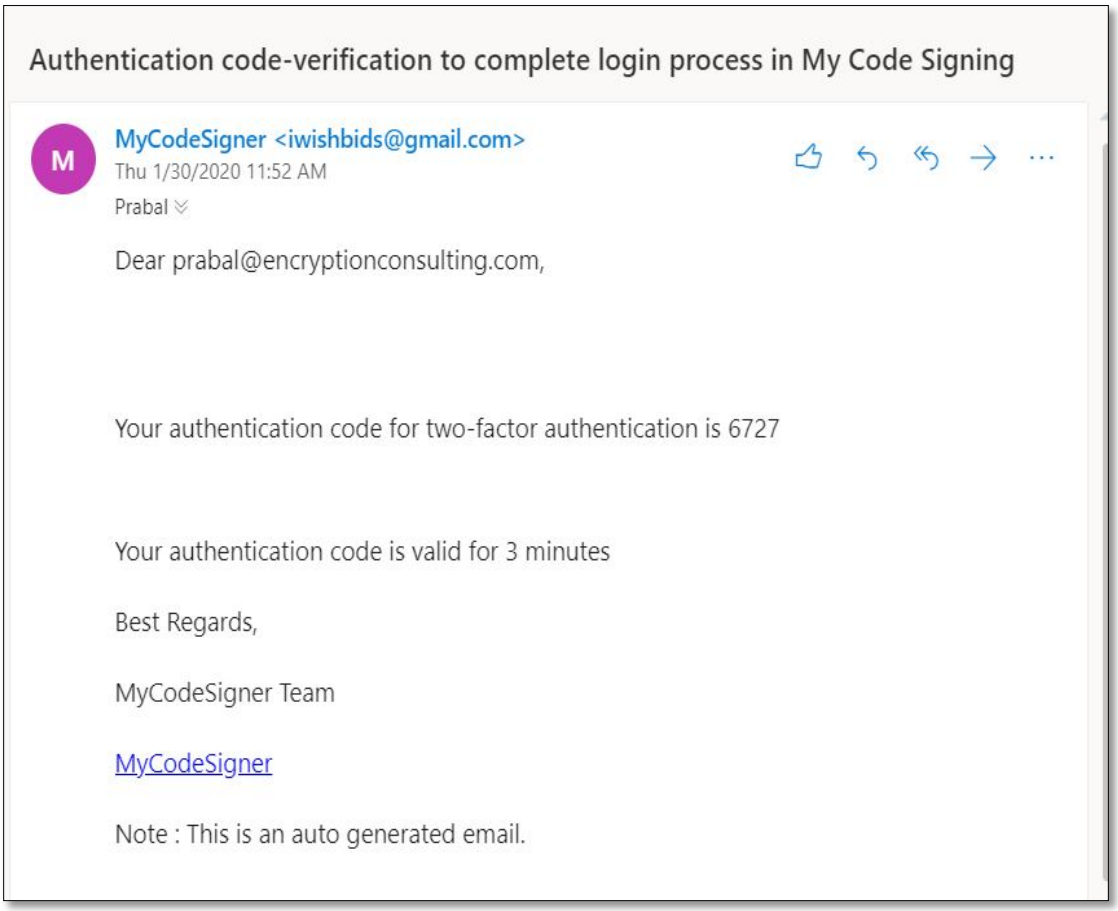
- Choose Headers option and Enter token in key data. Use the token value (generated on successful login)



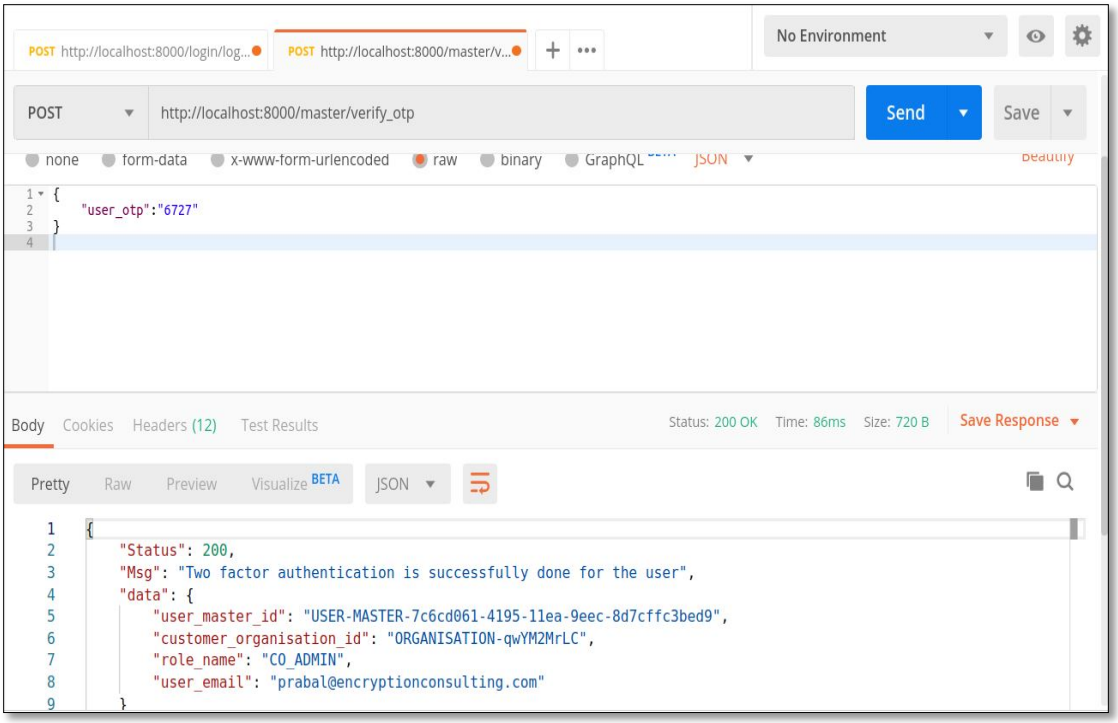
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Sample of email received with Authentication Code



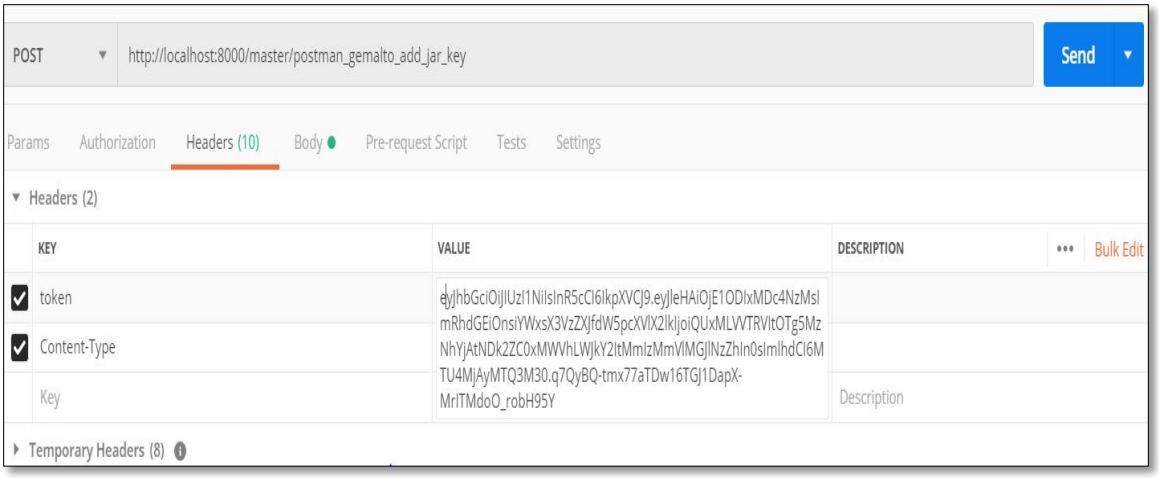
- Inside the body, enter the Authentication Code as per shared screenshot format. Then Click Send Button.



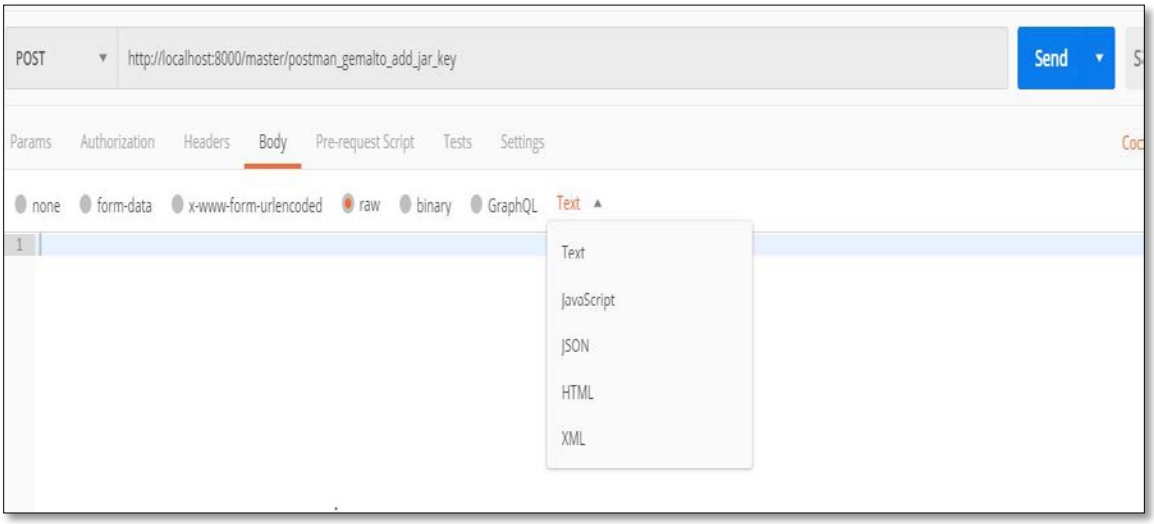
7.3 Generate Key

- Enter the key generation for signing jar application URL and Choose Method POST.

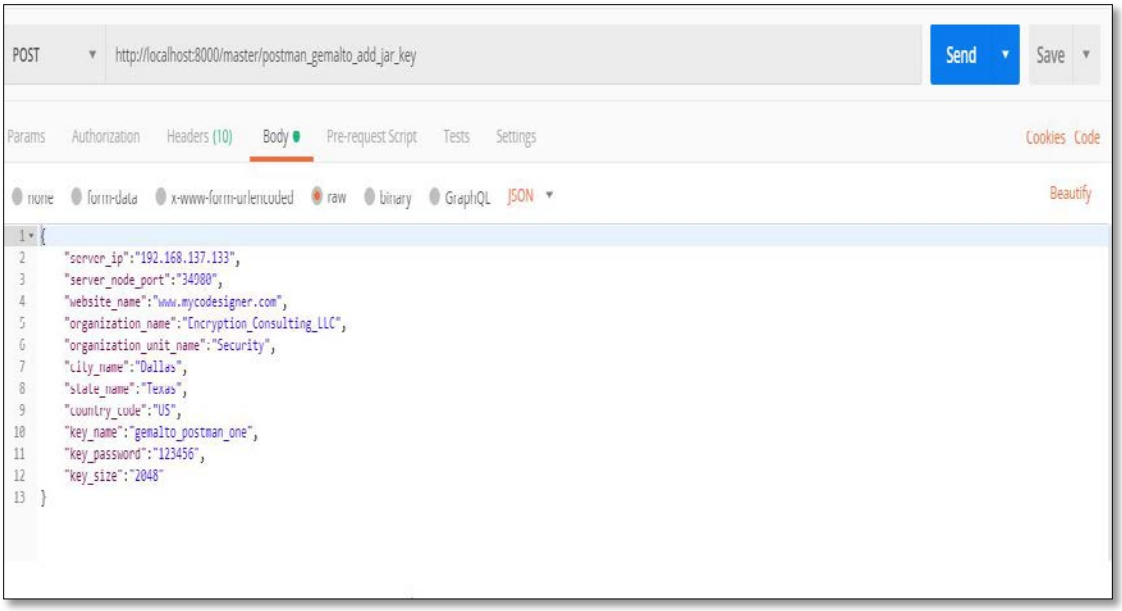
- Choose Headers option and Enter token in key data. Use the token value (generated on successful login).



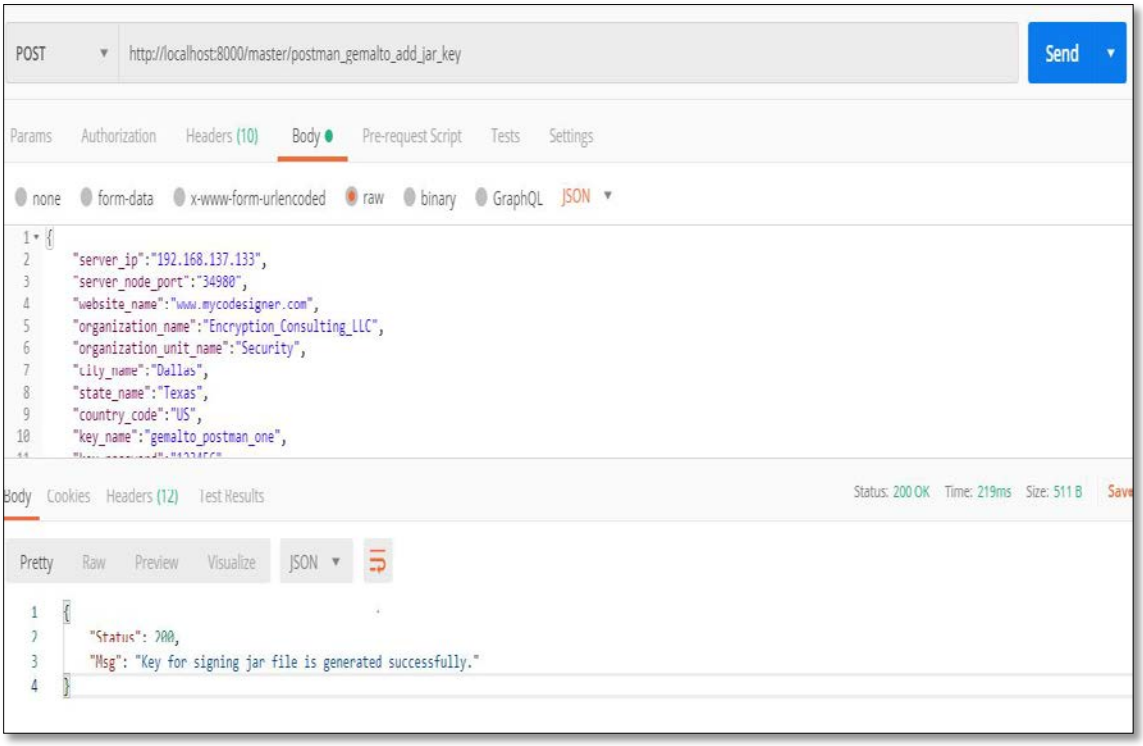
- Choose the Body option. Once Body option is chosen, click the raw radio button and choose JSON from the drop down.



- Inside the body, enter the mandatory data to generate key as per shared screenshot format. Then Click Send Button.

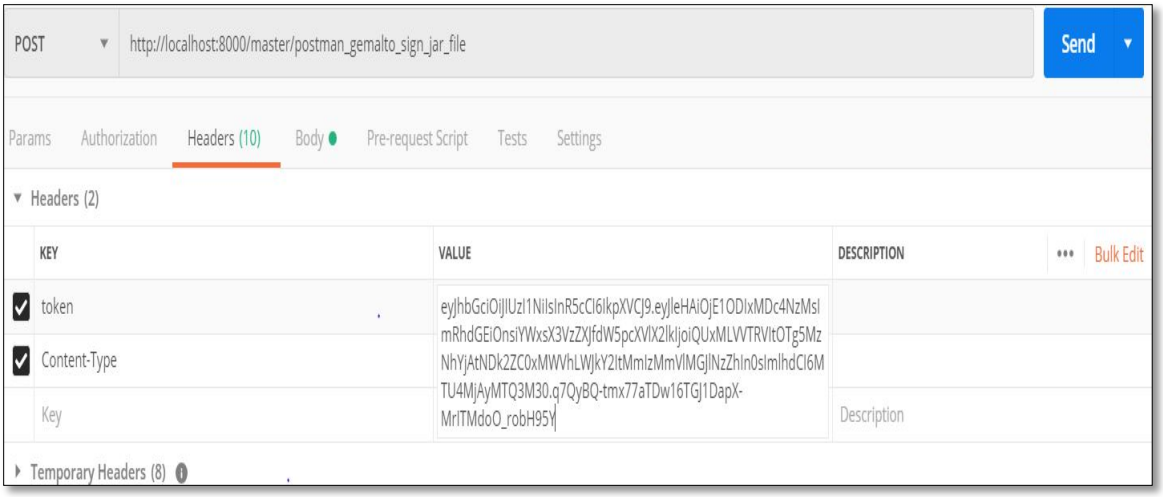


- On Successful key generation, the response status will be 200.

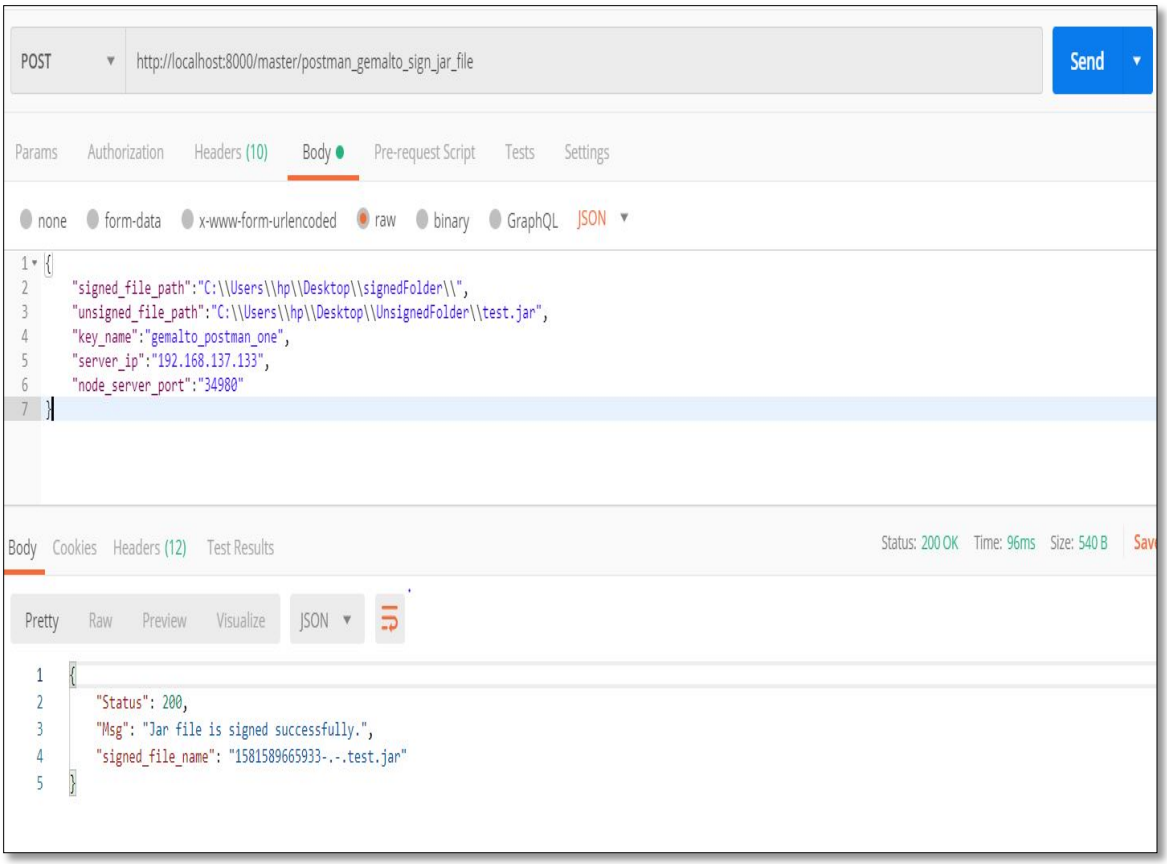


7.4 Signing Jar application

- Enter signing jar application URL and Choose Method POST. Choose Headers option and Enter token in key data. Use the token value (generated on successful login).

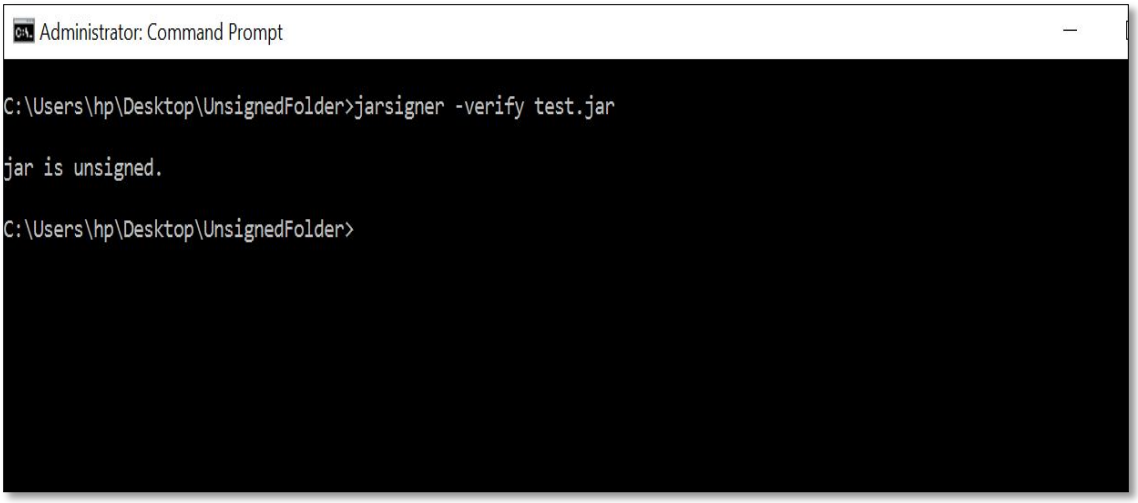


- Inside the body, enter the mandatory data to sign jar application as per shared screenshot format. Then Click Send Button. On Successful signing, the response status will be 200 along with the signed file name.



7.5 Verify JAR file after signing

- Check whether jar application is signed. For unsigned jar application, output of the command will have message “jar is unsigned”. The command to check digital signature of jar is “jarsigner –verify test.jar”



- Verify the digital signature of the signed jar application. The command to check digital signature of jar is “jarsigner –verbose –verify 1581589665933-.-.test.jar”