SHIELD DOCUMENT

Version 1.0.0

AUTHOR:

Parthiban Nithyanantham, Software Engineer from India. Can be reached using email-id n.parthibann@gmail.com.

INTRODUCTION:

Shield is a python-flask based micro service which is used to

- Create Self-Signed SSL certificates
- Create Organizational certificate chain / Trusted certificate chain
- Create Server certificates / licensing using certificates
- Create Keypair (private key and public key)

All these data will be stored in mongodb and served based on the request.

Pre-requisite:

- Mongodb

Installation Instructions:

1. Download / Clone the repository.
2. Move inside the repository folder and run the following command "pip install –r requirements.txt"
3. Update config file with mongo db details "shield.conf"
4. Start the application by running the "shield" file present in the bin directory (e.g python shield)

API Details:

1. **Create Certificate**

Method: POST

Endpoint: http://<ip>:<port>/v1/certificates

Headers: Content-Type: application/json

Request-body:

```
{
        "common_name": "parthi1",
        "valid_from": "2018-05-21T22:22:26",
        "valid_till": "2019-08-20T22:22:26",
        "cert_type": "ca_root",
        "path_length": 0,
        "country": "IN",
        "state": "Tamilnadu",
        "locality": "chennai",
        "organization_name": "companyX",
        "organization_unit_name": "branchX",
        "subject_alternate_name": "companyx.com",
        "signature_algorithm": "sha256"
}
```

Request Parameter Details:

Parameter Name	**Required / Optional**	**Details**
common_name	Required	Name of the certificate
valid_from	Required	Starting validity of the certificate
valid_till	Required	Ending validity of the certificate
cert_type	Required	Type of the certificate, can be any of the following (ca_root, ca_intermediate, end_entity, self_signed)
path_length	Optional	No of intermediate ca certificates that can exist
Country	Optional	Country name
State	Optional	State name
Locality	Optional	Locality name
organization_name	Optional	Organization name
organization_unit_name	Optional	Organization unit name
subject_alternate_name	Optional	Subject alternate name
signature_algorithm	Optional	Signature algorithm, can be any of the following (sha1 / sha256)
issuer_id	Optional (Required when cert_type is ca_intermediate or end_entity)	Objectid of the issuer certificate

Sample Response:

```
{
   "status": "success",
   "message": "Certificate created successfully.",
   "id": "5b0accf183278510746312d3"
}
```

2. View Certificate

Method: GET

Endpoint: http://<ip>:<port>/v1/certificates/<certid>

Headers: Content-Type: application/json

Sample Response:

```
{
   "status": "success",
   "message": "Certificate information",
   "data": {
      "status": "active",
      "certificate_type": "ca_root",
      "valid_till": "2019-08-20 22:22:26",
      "valid_from": "2018-05-21 22:22:26",
      "path_length": 0,
      "country": "IN",
      "created_at": "2018-05-27 15:21:21.737000",
      "locality": "chennai",
      "organization_name": "companyX",
      "issuer_id": null,
      "signature_algorithm": "sha256",
      "common_name": "parthi1",
      "id": "5b0accf183278510746312d3",
      "organization_unit_name": "branchX"
   }
}
```

3. List Certificate

Method: GET

Endpoint: http://<ip>:<port>/v1/certificates

Headers: Content-Type: application/json

Query params:

Limit: number of records to view

`skip_val: number of records to be skipped`

Sample Response:

```
{
   "status": "success",
   "recordsFiltered": 1,
   "draw": 1,
   "recordsTotal": 1,
   "message": "Certificates list",
   "page_count": 1,
   "data": {
       "certificates": [
           {
               "id": "5b57540d83278511f843a81c",
               "common_name": "my_cert",
               "cert_type": "ca_root",
               "signature_algorithm": "sha256",
               "valid_from": "2018-07-24 21:53:00",
               "valid_till": "2018-07-31 21:53:00",
               "key_id": "5b57540d83278511f843a819",
               "issuer_id": null
           }
       ]
   }
}
```

4. Delete Certificate

Delete certificate just change the state from ‘active’ to ‘in-active’, only soft delete happens and the actual data will still exist in the database as in-active.

Method: DELETE

Endpoint: http://<ip>:<port>/v1/certificates/<certid>

Headers: Content-Type: application/json

Sample Response:

```
{
    "status": "success",
    "message": "Certificate deleted successfully."
}
```

5. Download Resource(certificate / private key / public key)

Method: GET

Endpoint: http://<ip>:<port>/v1/certificates/download/<type>/<resource_id>

Headers: Content-Type: application/json

Sample Response:

The requested resource will get downloaded as a pem file.

INFO: type will be any of the following (cert, private_key, public_key), and resource_id will be the id of the type specified.

6. **Download Resource(certificate / private key / public key)**

Method: GET

Endpoint: http://<ip>:<port>/v1/certificates?action=calist

Headers: Content-Type: application/json

Sample Response:

```
{
   "ca_root": [],
   "self_signed": [],
   "ca_intermediate": [
      {
         "id": "5b57540d83278511f843a81c",
         "name": "my_cert"
      }
   ],
   "end_entity": [
      {
         "id": "5b57540d83278511f843a81c",
         "name": "my_cert"
      }
   ]
}
```

USAGE:

- **CREATE SELF-SIGNED CERTIFICATE**

Using create certificate api, create a self-signed ssl certificate. Sample request body given below:

```
{
         "common_name": "parthi_selfsigned",
         "valid_from": "2018-05-21T22:22:26",
         "valid_till": "2019-08-20T22:22:26",
         "cert_type": "self_signed",
         "path_length": 0,
         "country": "IN",
         "state": "Tamilnadu",
         "locality": "chennai",
         "organization_name": "companyX",
         "organization_unit_name": "branchX",
         "subject_alternate_name": "companyx.com",
         "signature_algorithm": "sha256"
}
```

The response will contain the id of the certificate created. Then, using download resources api, download the certificate pem file

- **CREATE TRUSTED CERTIFICATE CHAIN**

STEP 1: Using create certificate api, create CA root certificate, sample request body given below.

```
{
        "common_name": "parthiban_caroot",
        "valid_from": "2018-05-21T22:22:26",
        "valid_till": "2019-08-20T22:22:26",
        "cert_type": "ca_root",
        "path_length": 2,
        "country": "IN",
        "state": "Tamilnadu",
        "locality": "chennai",
        "organization_name": "companyX",
        "organization_unit_name": "branchX",
        "subject_alternate_name": "companyx.com",
        "signature_algorithm": "sha256"
}
```

Response will contain the certificate id.

STEP2: Create ca intermediate certificate, same request body given below. Use above generated certificate id as issuer id.

```
{
        "common_name": "parthiban_cainter",
        "valid_from": "2018-05-21T22:22:26",
        "valid_till": "2019-08-20T22:22:26",
        "cert_type": "ca_inter",
        "path_length": 2,
        "country": "IN",
        "state": "Tamilnadu",
        "locality": "chennai",
        "organization_name": "companyX",
        "organization_unit_name": "branchX",
        "subject_alternate_name": "companyx.com",
        "signature_algorithm": "sha256",
        “issuer_id”: “5b0adc828327853758dabf81”
}
```

Response will contain the certificate id.

STEP3: Create end_entity / leaf / ssl certificate, sample request body given below. Use above generated certificate id as issuer id.

```
{
        "common_name": "parthiban_endentity ",
        "valid_from": "2018-05-21T22:22:26",
        "valid_till": "2019-08-20T22:22:26",
        "cert_type": "end_entity",
        "path_length": 0,
        "country": "IN",
        "state": "Tamilnadu",
        "locality": "chennai",
        "organization_name": "companyX",
        "organization_unit_name": "branchX",
        "subject_alternate_name": "companyx.com",
        "signature_algorithm": "sha256",
        “issuer_id”: “5b0adcbf8327853758dabf8f”
}
```

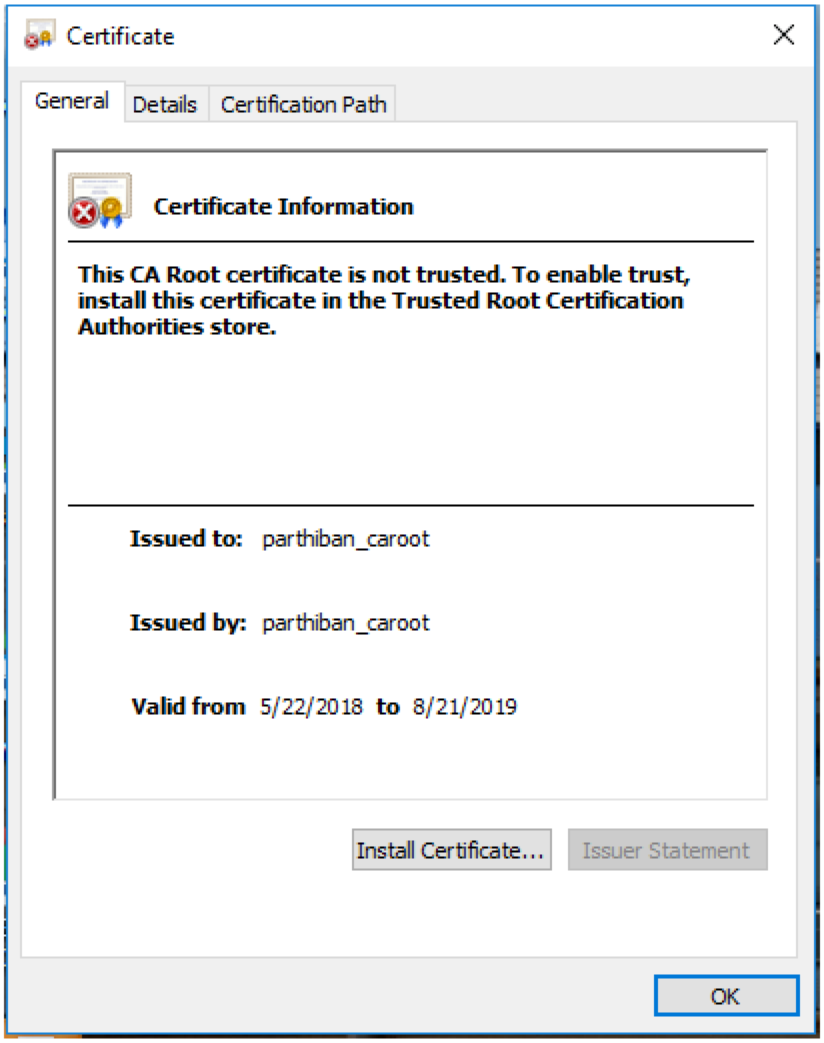
Response will contain the certificate id.

ACCEPTANCE TESTING:

STEP1: We have created an organization certificate chain above, download all the 3 certificates pem file using download resource api.

STEP2: Change the file extension from .pem to .crt and then open the certificate.

STEP3: When you open the certificate, you get the following information “**This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.**”


Certificate

General | Details | Certification Path

Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

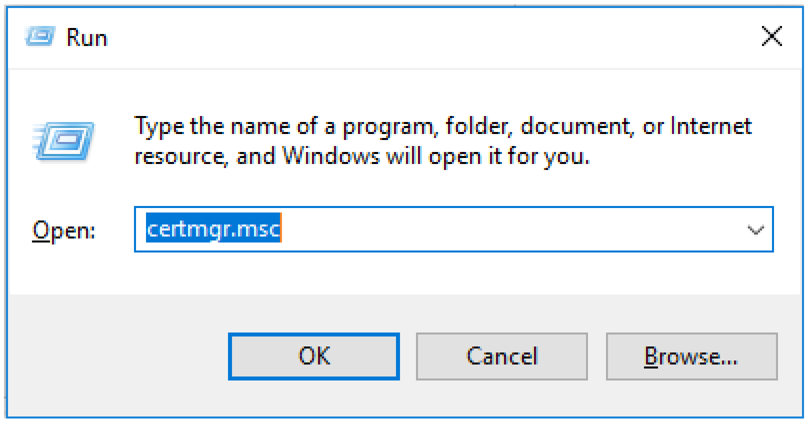
Issued to: parthiban_caroot

Issued by: parthiban_caroot

Valid from 5/22/2018 **to** 8/21/2019

Steps to install certificate in Trusted Root Certification Authorities store (windows):

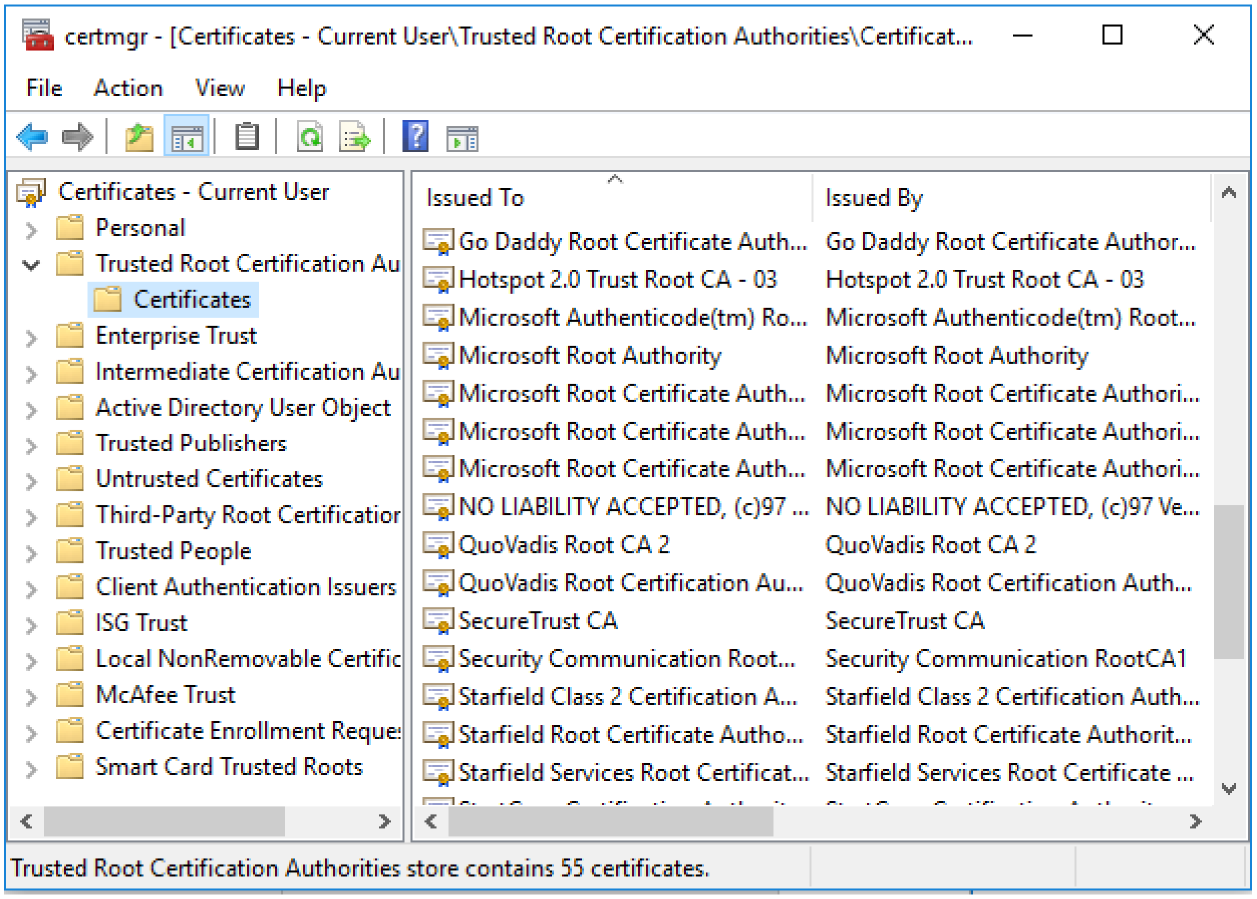
1. Open run prompt and type “certmgr.msc” and press enter.


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: certmgr.msc

2. In “certmgr” window, select -> Trusted Root Certification Authorities -> certificates



certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificat...

File Action View Help

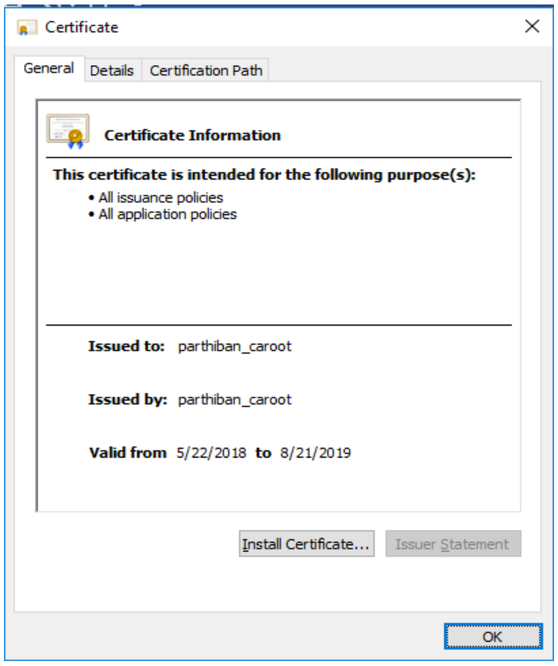
Issued To	Issued By
Go Daddy Root Certificate Auth...	Go Daddy Root Certificate Author...
Hotspot 2.0 Trust Root CA - 03	Hotspot 2.0 Trust Root CA - 03
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...
Microsoft Root Authority	Microsoft Root Authority
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
NO LIABILITY ACCEPTED, (c)97 ...	NO LIABILITY ACCEPTED, (c)97 Ve...
QuoVadis Root CA 2	QuoVadis Root CA 2
QuoVadis Root Certification Au...	QuoVadis Root Certification Auth...
SecureTrust CA	SecureTrust CA
Security Communication Root...	Security Communication RootCA1
Starfield Class 2 Certification A...	Starfield Class 2 Certification Auth...
Starfield Root Certificate Autho...	Starfield Root Certificate Authorit...
Starfield Services Root Certificat...	Starfield Services Root Certificate ...

Trusted Root Certification Authorities store contains 55 certificates.

Certificates - Current User
Personal
Trusted Root Certification Au
Certificates
Enterprise Trust
Intermediate Certification Au
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certificatior
Trusted People
Client Authentication Issuers
ISG Trust
Local NonRemovable Certific
McAfee Trust
Certificate Enrollment Reque
Smart Card Trusted Roots

3. Right click on Certificates -> All Tasks -> Import and then import the root certificate.

STEP4: Once the import is successful, open the certificate again, this time you will see the intention of the certificate message.



Certificate Information

This certificate is intended for the following purpose(s):

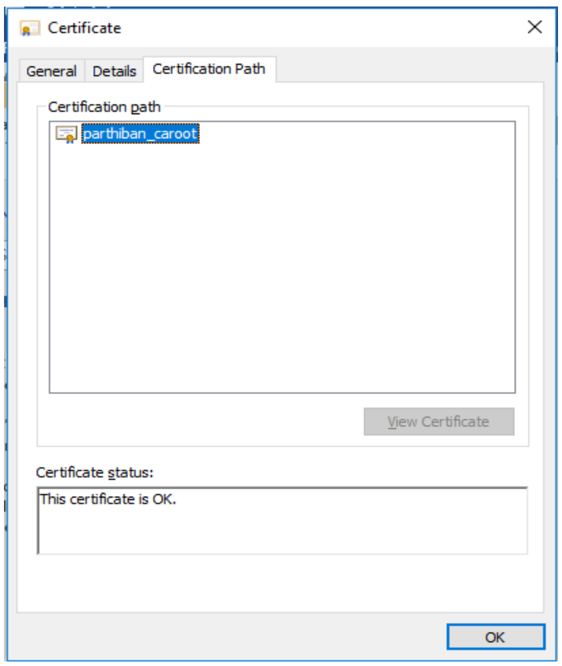
- All issuance policies
- All application policies

Issued to: parthiban_caroot

Issued by: parthiban_caroot

Valid from 5/22/2018 **to** 8/21/2019

STEP5: Also navigate to “Certification Path” tab in the certificate and check the Certification Path and Certification status:



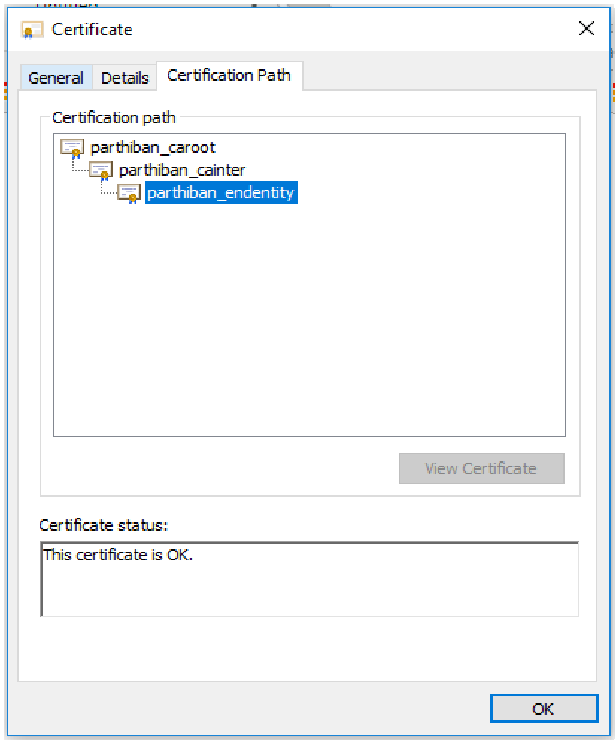
parthiban_caroot

Certificate status:

This certificate is OK.

STEP6: Likewise, install the intermediate certificate in “Intermediate Certificate Authorities” store.

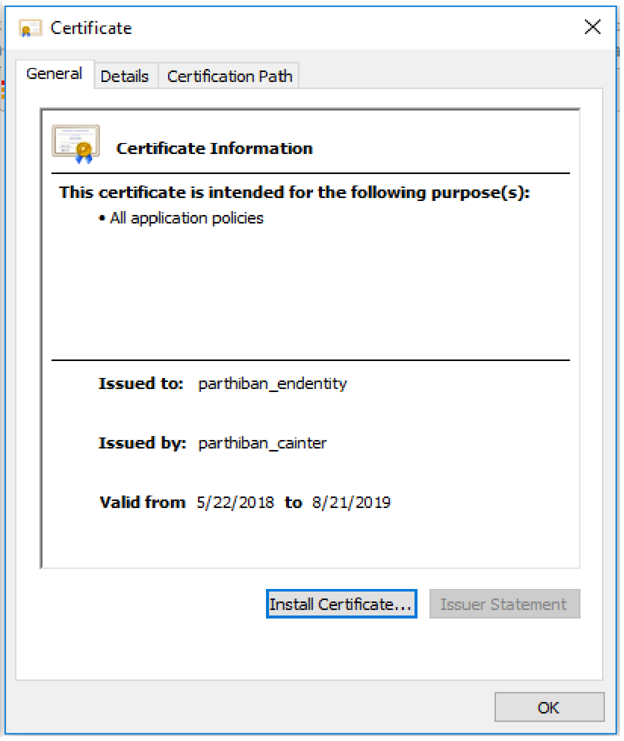
Step7: Finally open the end-entity certificate, and check for the certification path and certification status, if everything is ok then the certificate is valid for usage.



parthiban_caroot
parthiban_cainter
parthiban_endentity

Certificate status:

This certificate is OK.



Certificate Information

This certificate is intended for the following purpose(s):

- All application policies

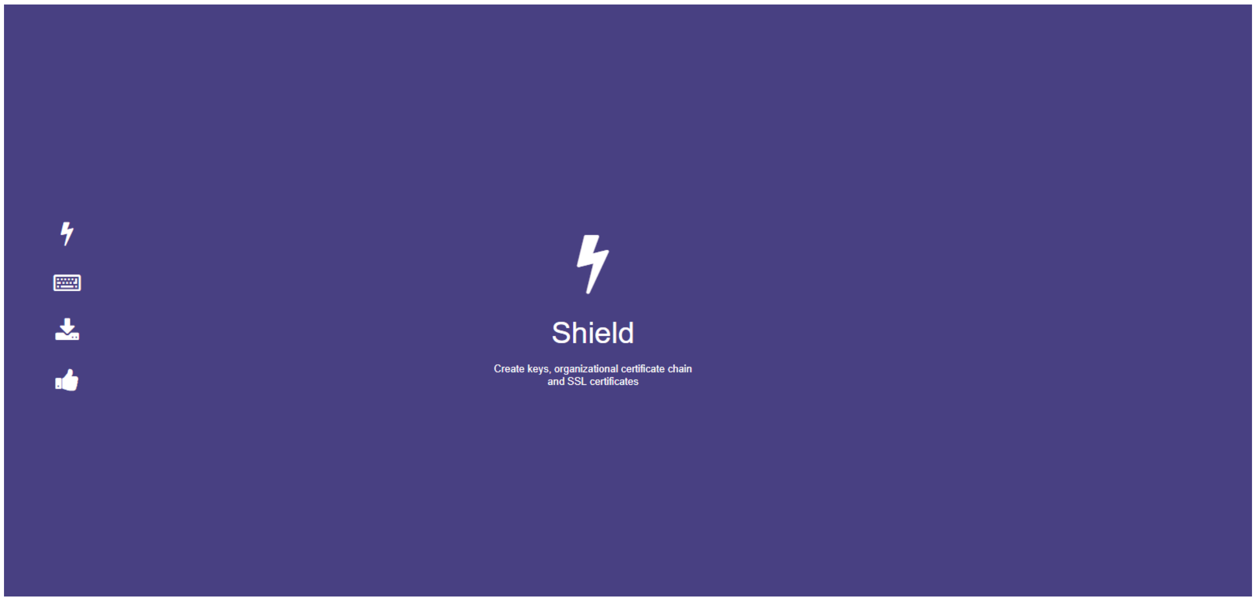
Issued to: parthiban_endentity

Issued by: parthiban_cainter

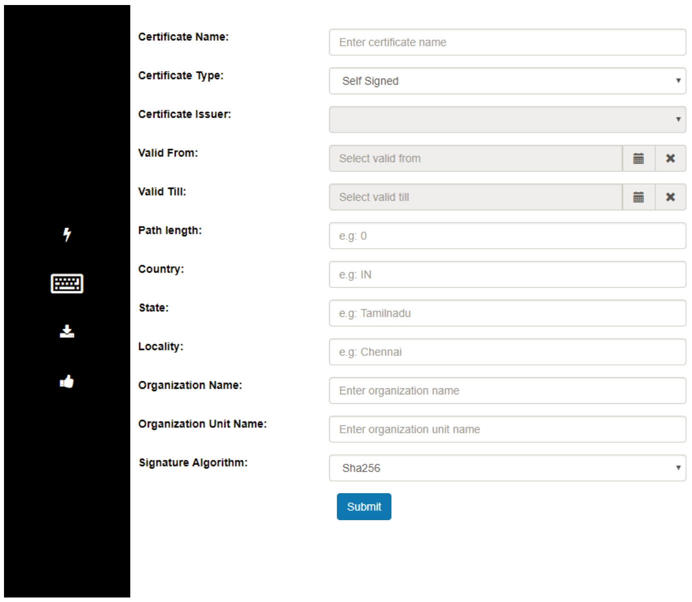
Valid from 5/22/2018 **to** 8/21/2019

UI:

Landing page:

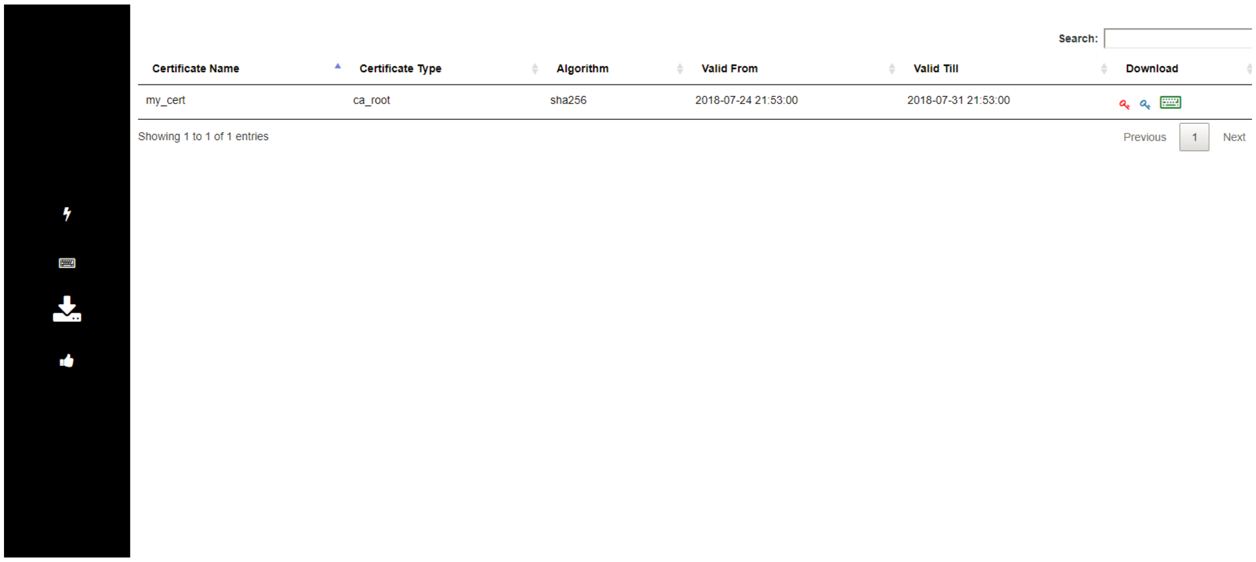

Shield
Create keys, organizational certificate chain and SSL certificates

Certificate creation page:



Certificate Name:	Enter certificate name
Certificate Type:	Self Signed
Certificate Issuer:	
Valid From:	Select valid from
Valid Till:	Select valid till
Path length:	e.g: 0
Country:	e.g: IN
State:	e.g: Tamilnadu
Locality:	e.g: Chennai
Organization Name:	Enter organization name
Organization Unit Name:	Enter organization unit name
Signature Algorithm:	Sha256

Certificate downloads page:

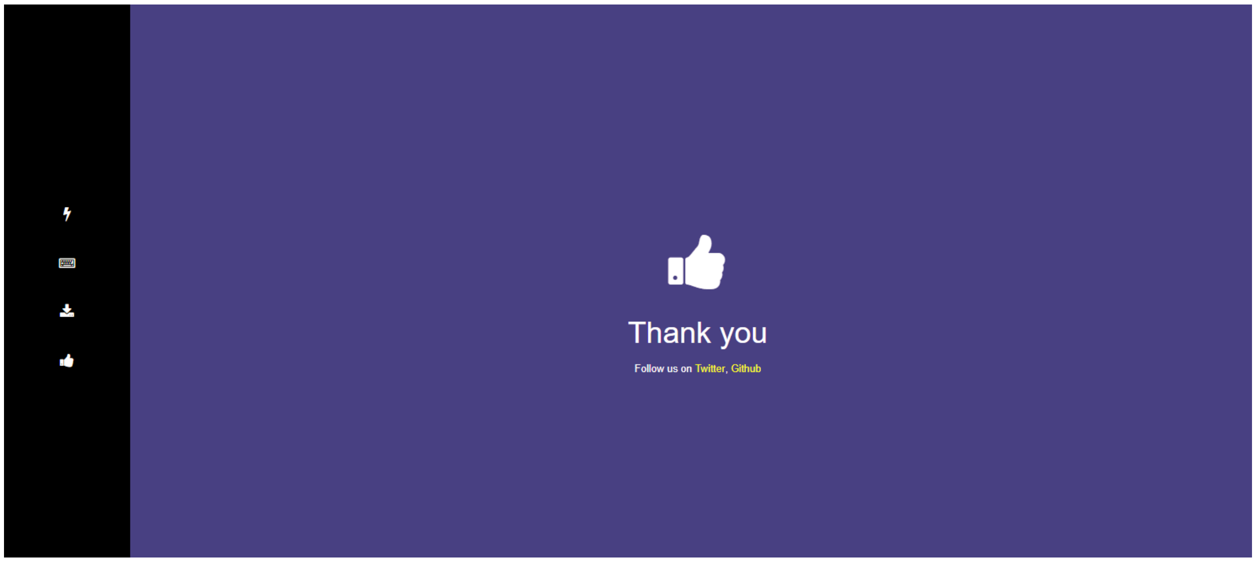


Search:

Certificate Name	Certificate Type	Algorithm	Valid From	Valid Till	Download
my_cert	ca_root	sha256	2018-07-24 21:53:00	2018-07-31 21:53:00	

Showing 1 to 1 of 1 entries

Thank you:



Thank you

Follow us on Twitter, Github