InCommon®

InCommon Certificate Manager

RESTful API
TLS/SSL

InCommon
c/o Internet2
1000 Oakbrook Drive, Suite 300
Ann Arbor MI, 48104

Table of Contents

Version History........2
1 Introduction........3
1.1 HTTP Methods........3
1.2 HTTP Status Codes........3
1.3 Errors........3
2 Authentication........4
2.1 Authentication via Username and Password........4
2.2 Authentication via Username and a Client Certificate........4
3 API Resources (Functions)........5
3.1 TLS/SSL Certificates........5
3.1.1 Listing SSL Types........5
3.1.1.1 Response Attributes........5
3.1.1.2 Example Request........5
3.1.1.3 Example Response........5
3.1.2 TLS/SSL Enrollment........6
3.1.2.1 Request Attributes........6
3.1.2.1.1 Server Type........7
3.1.2.2 Example Request........9
3.1.2.3 Example Response........9
3.1.3 Collect TLS/SSL Certificate........10
3.1.3.1 Example Request........10
3.1.3.2 Example Response........10
3.1.4 TLS/SSL Certificate Revocation........10
3.1.4.1 Path Parameters........11
3.1.4.2 Request Attributes........11
3.1.4.3 Example Request........11
3.1.4.4 Example Response........11
3.1.5 TLS/SSL certificate renewal by RenewId........12
3.1.5.1 Path Parameters........12
3.1.5.2 Example Request........12
3.1.5.3 Example Response........12
3.1.6 TLS/SSL certificate renewal by SSL enrollment ID........13
3.1.6.1 Path Parameters........13
3.1.6.2 Example Request........13
3.1.6.3 Response fields........13
3.1.6.4 Example Response........13
3.1.7 TLS/SSL Certificate Replacement........14
3.1.7.1 Path Parameters........14
3.1.7.2 Request Attributes........14
3.1.7.3 Example Request........15
3.1.7.4 Example Response........15

Version History

Date	Description
28 January 2018	Initial Release; CCM version 6.0

1 Introduction

1.1 HTTP Methods

InCommon Certificate Manager (CM) tries to adhere as closely as possible to standard HTTP and REST conventions in its use of of HTTP Methods (verbs).

Method	Usage
GET	Used to retrieve a resource
POST	Used to create a new resource

1.2 HTTP Status Codes

InCommon Certificate Manager (CM) tries to adhere as closely as possible to standard HTTP and REST conventions in its use of of HTTP status codes.

Status Code	Usage
200 OK	The request completed successfully
204 No Content	An update to an existing resource has been applied successfully
400 Bad Request	The request was malformed. The response body will include an error providing further information
404 Not Found	The requested resource does not exist

1.3 Errors

Whenever an error response (status code >=400) is returned, the body will contain a JSON object that describes the problem. The error object has the following structure:

Path	Type	Description
code	Number	Error code
description	String	Error message

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
Content-Length: 41
{"code":-16,"description":"Unknown user"}
```

2 Authentication

In order to access InCommon APIs, you will need to authenticate yourself to the InCommon CM service. You can authenticate via username/password, or via username + client certificate.

- Authentication via Username and Password
- Authentication via Username and a Client Certificate

All API functions (methods) require HTTP header parameters for authorization.

Header Name	Description
login	Privileged User's Login Name
password	*Privileged User's Password; (if necessary)*
CustomerUri	Customer URI part of the URL e.g.s. *InCommon* or *InCommon_Test*

2.1 Authentication via Username and Password

Prerequisites

- Users should have InCommon CM login credentials.
- **Web API** access must be enabled for the Organization by InCommon.
    - Each department department will need to be enabled by their RAO through InCommon CM.

The URI for the username/password authentication schema for InCommon CM is:

- {api-endpoint} = https://cert-manager.com/api/{path}

2.2 Authentication via Username and a Client Certificate

Prerequisites

- Users should have InCommon CM login credentials.
- **Web API** access must be enabled for the Organization by InCommon.
  - Each department department will need to be enabled by their RAO through InCommon CM.
- Admins should have **Certificate Auth** enabled.
- The authentication certificate MUST BE requested and issued through InCommon CM and active at the moment of authentication.

The URI for the username/client certificate authentication is:

- {api-endpoint} = https://cert-manager.com/private/api/{path}

3 API Resources

3.1 TLS/SSL Certificates

All functions pertaining to the management of TLS/SSL certificates within InCommon CM.

Basic Auth URI endpoint – https://cert-manager.com/api/ssl/v1/{path}
Client Auth URI endpoint – https://cert-manager.com/private/api/ssl/v1{path}

3.1.1 Listing SSL Types

Path – /ssl/v1/types
HTTP Method – GET

3.1.1.1 Response Attributes

Attribute	Type	Description
[ ]	Array	An array of available SSL Types
[ ] id	String	The SSL Type unique identifier
[ ] name	String	The SSL Cert Type name
[ ] terms [ ]	Array	An array of available terms, in days, for the SSL type.

3.1.1.2 Example Request

```
$ curl 'https://cert-manager.com/api/ssl/v1/types/' -i \
      -H 'Content-Type: application/json' \
      -H 'login: forest.gump@university.edu' \
      -H 'password: RollTide4Jenny' \
      -H 'customerUri: InCommon'
```

3.1.1.3 Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 38
[{"id":2018,"name":"InCommon SSL","terms":[365]}]
```

3.1.2 TLS/SSL Enrollment

Creation and submission of a request for a new TLS/SSL certificate.

Path – /ssl/v1/enroll?
HTTP Method – POST

3.1.2.1 Request Attributes

Attribute	Type	Description	Constraints
orgId	Number	Organization ID; can be found within InCommon CM on the organization's and or department's General tab.	MUST BE positive & not NULL
csr	String	Certificate Signing Request (Base-64 encoded, with or without the: -----BEGIN xxxxx----- and -----END xxxxx----- header and footer)	MUST not be empty or NULL. Max size: 32767 characters.
SubjAltNames	String	Comma-Separated list of DNS Subject Alternative Names (SANs)	A maximum of 100 SANs.
certType	Number	Certificate Type ID	Obtained from **/ssl/types** OR **/ssl/types{orgId}** function
numberServers	Number	Number of Server Licenses. Required for the Wildcard products	MUST BE positive & not NULL
serverType	Number	Server Software Identifier	MUST BE at least -1; *See Server Type table below for acceptable attribute values*
term	Number	Certificate validity period in	MUST BE positive.

		number of days.	
comments	String	Additional Comments/Notes for Enrollment Request	A maximum of 1024 characters accepted.
customFields []	Array	Custom fields to be applied to enrolling certificate.	MUST contain mandatory custom fields.
customFields[].name	String	The name of an enabled mandatory custom field.	
customFields[].value	String	The value of the custom field	

3.1.2.1.1 Server Type

Server Type	Description
1	AOL
2	Apache/ModSSL
3	Apache-SSL (Ben-SSL, not Stronghold)
4	C2Net Stronghold
33	Cisco 3000 Series VPN Concentrator
34	Citrix
5	Cobalt Raq
6	Covalent Server Software
7	IBM HTTP Server
8	IBM Internet Connection Server
9	iPlanet
10	Java Web Server (Javasoft / Sun)
11	Lotus Domino
12	Lotus Domino Go!
13	Microsoft IIS 1.x to 4.x

Server Type	Description
14	Microsoft IIS 5.x and later
15	Netscape Enterprise Server
16	Netscape FastTrack
17	Novell Web Server
18	Oracle
19	Quid Pro Quo
20	R3 SSL Server
21	Raven SSL
22	RedHat Linux
23	SAP Web Application Server
24	Tomcat
25	Website Professional
26	WebStar 4.x and later
27	WebTen (from Tenon)
28	Zeus Web Server
29	Ensim
30	Plesk
31	WHM/cPanel
32	H-Sphere
-1	OTHER

3.1.2.2 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/enroll/' -i -X POST \
-H 'Content-Type: application/json' \
-H 'login: admin_customer1453' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1453' \
-d '{"orgId":766,"csr":"-----BEGIN CERTIFICATE REQUEST-----\
nMIIC4jCCAcoCAQAwdDELMAkGA1UEBhMCVUExDTALBgNVBAgTBHRlc3QxDTALBgNV\
nBAcTBHRlc3QxDTALBgNVBAoTBHRlc3QxDTALBgNVBAsTBHRlc3QxEjAQBgNVBAMT\
nCWNjbXFhLmNvbTEVMBMGCSqGSIb3DQEJARYGdGVzdEB0MIIBIjANBgkqhkiG9w0B\
nAQEFAAOCAQ8AMIIBCgKCAQEAul8SGkicOnrMjJDvgG8P2j1Ee5hY6ww+qSoe0oI2\
ntvRcLBknPHMMAkxTjW9fy80wD8hyrnc+IGlQcq2R/tEMIJHRsJD603M+2FjAwlP9\
n8xtiqv0hMyHO4fEt+HMyy8Q367aTBmnZCuAxJZJapfFW9wH5jGZxuX8mnrXVsBTD\
n4ZBO4UFd9P4u8P0nJx80CiuDt4COSDl6Br4pNLciPVqfwj7LQ5/skwPkNCggk3/G\nxoQX/
3FV7O4fC6WCxVP1uYjJVQjlD1Tf06hPNfonVfThVuP20OL3QAlnIF3lZiyY\nJ5etdFtu+BKcPNMdQDJOS/
O4Zz0YJn6K2HdAXSc1YxYniwIDAQABoCkwJwYJKoZI\
nhvcNAQkOMRowGDAJBgNVHRMEAjAAMAsGA1UdDwQEAwIF4DANBgkqhkiG9w0BAQsF\
nAAOCAQEAVJVTTELGHWoRh8JZt+kx/zO0VnibBq/D6uB405L+Ir80X48Ei9hTLB11\
nAqhSBE+AbEgBhRnEIDBjiXEDcWvC532Omex721kc17ZTzowuD8lOjfQkTHbAmjIi\nnCQNFAPf0D/
zpi6Eync5pi2P//Uj/Yn7oDYYli1t61EZwuQyEu4mbQ1efUnU/SOl\
nAAQtDPhNwATZPmfefjM8+YuzhG70dQvmFAClcFayKM92Zx9khDd/VnLT85YzDULJ\
n8iiHW8dZNblaTsUjKrc73iX2hONZIxw6B3tGCFs8mH9lZlExV7Y2er3t/lO1pdxe\
nSUohEELWcttIxyWnYgxvwaWX4lfx9A\u003d\u003d\n-----END CERTIFICATE
REQUEST-----","subjAltNames":"ccmqa.com","certType":500,"numberServers":0,"serverType":-
1,"term":365,"comments":"test","customFields":[{"name":"custom field","value":"custom field
value"}]}'
```

3.1.2.3 Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 46

{"renewId":"rqSqUt19WVhzmM6-hagZ","sslId":583}
```

3.1.3 Collect TLS/SSL Certificate

Certificate retrieval (collection) of an issued certificate from InCommon CM

Path – /ssl/v1/collect/{sslId}/{formatType}
HTTP Method – POST

Parameter	Description
sslId	Certificate ID; positive Number value; Minimum value = 1 Displayed in InCommon CM as 'Self Enrollment Certificate ID'
formatType	Format Type for certificate collection. **Allowed values:** • '**x509**' - for X509, Base64 encoded • '**x509CO**' - for X509 Certificate only, Base64 encoded • '**x509IO**' - for X509 Intermediates/root only, Base64 encoded • '**x509IOR**' - for X509 Intermediates/root only Reverse, Base64 encoded • '**base64**' - for PKCS#7 Base64 encoded • '**bin**' - for PKCS#7 Bin encoded

3.1.3.1 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/collect/582/base64' -i \
-H 'login: admin_customer1447' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1447'
```

3.1.3.2 Example Response

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
content-disposition: attachment; filename=test.cert
```

3.1.4 TLS/SSL Certificate Revocation

Generate a request to InCommon CM to revoke a specific TLS/SSL certificate.

Path – /ssl/v1/revoke/{sslId}
HTTP Method – POST

3.1.4.1 Path Parameters

Parameter	Description
sslId	Certificate ID; positive Number value; Minimum value = 1 Displayed in InCommon CM as 'Self Enrollment Certificate ID'

3.1.4.2 Request Attributes

Attribute	Type	Description	Constraints
reason	String	A short comment as to why the certificate needs to be revoked.	MUST NOT be empty. Maximum characters = 512

3.1.4.3 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/revoke/587' -i -X POST \
-H 'Content-Type: application/json' \
-H 'login: admin_customer1489' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1489' \
-d '{"reason": "test"}'
```

3.1.4.4 Example Response

```
HTTP/1.1 204 No Content
```

3.1.5 TLS/SSL certificate renewal by RenewId

A function to initiate the renewal of a given certificate using the same CSR and parameters of the existing certificate.

PATH – /ssl/v1/renew/{renewId}

HTTP Method – POST

3.1.5.1 Path Parameters

Parameter	Type	Max. Length (chars)	Description
renewId	String	20	Returned via the enrollment API call. It is also found in every Enrollment Successful email that InCommon CM sends.

3.1.5.2 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/renew/10' -i -X POST \
-H 'Content-Type: application/json' \
-H 'login: admin_customer1479' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1479' \
-d '{"renewId": "rqSqUt19WVhzmM6-hagZ"}'
```

3.1.5.3 Example Response

```
HTTP/1.1 204 No Content
```

3.1.6 TLS/SSL certificate renewal by SSL enrollment ID

A function to initiate the renewal of a given certificate using the same CSR and parameters of the existing TLS/SSL certificate.

Path – /ssl/v1/renewById/{sslId}

HTTP Method – POST

3.1.6.1 Path Parameters

Parameter	Possible value(s)
sslId	Certificate ID; positive Number value; Minimum value = 1 Displayed in InCommon CM as 'Self Enrollment Certificate ID'

3.1.6.2 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/renew/10' -i -X POST \
-H 'Content-Type: application/json' \
-H 'login: admin_customer1479' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1479' \
-d '{"sslId":5150}'
```

3.1.6.3 Response fields

Attribute	Type	Description
sslId	Number	Renewed certificate’s identifier.

3.1.6.4 Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 13

{"sslId":8675309}
```

3.1.7 TLS/SSL Certificate Replacement

A function to initiate the replacement (or re-issuance) of a given certificate, based on a unique identifier and a new CSR.

3.1.7.1 Path Parameters

Parameter	Description
sslId	Certificate ID; positive Number value; Minimum value = 1 Displayed in InCommon CM as 'Self Enrollment Certificate ID'

3.1.7.2 Request Attributes

Attribute	Type	Description	Constraints
csr	String	Certificate Signing Request (Base-64 encoded, with or without the: -----BEGIN xxxxx----- and -----END xxxxx----- header and footer)	MUST not be empty or NULL. Max size: 32767 characters.
reason	String	A short comment as to why the certificate needs to be replaced (or re-issued).	MUST NOT be empty. Maximum characters = 512

3.1.7.3 Example Request

```
$ curl 'https://ccm.com/api/ssl/v1/replace/586' -i -X POST \
-H 'Content-Type: application/json' \
-H 'login: admin_customer1483' \
-H 'password: mLZxWzJh1+DZAPjHgnwzxaU/KVo=' \
-H 'customerUri: cst1483' \
-d '{"csr":"-----BEGIN CERTIFICATE REQUEST-----\
nMIIC4jCCAcoCAQAwdDELMAkGA1UEBhMCVUExDTALBgNVBAgTBHRlc3QxDTALBgNV\
nBAcTBHRlc3QxDTALBgNVBAoTBHRlc3QxDTALBgNVBAsTBHRlc3QxEjAQBgNVBAMT\
nCWNjbXFhLmNvbTEVMBMGCSqGSIb3DQEJARYGdGVzdEB0MIIBIjANBgkqhkiG9w0B\
nAQEFAAOCAQ8AMIIBCgKCAQEAul8SGkicOnrMjJDvgG8P2j1Ee5hY6ww+qSoe0oI2\
ntvRcLBknPHMMAkxTjW9fy80wD8hyrnc+IGlQcq2R/tEMIJHRsJD603M+2FjAwlP9\
n8xtiqv0hMyHO4fEt+HMyy8Q367aTBmnZCuAxJZJapfFW9wH5jGZxuX8mnrXVsBTD\
n4ZBO4UFd9P4u8P0nJx80CiuDt4COSDl6Br4pNLciPVqfwj7LQ5/skwPkNCggk3/G\nxoQX/
3FV7O4fC6WCxVP1uYjJVQjlD1Tf06hPNfonVfThVuP20OL3QAlnIF3lZiyY\nJ5etdFtu+BKcPNMdQDJOS/
O4Zz0YJn6K2HdAXSc1YxYniwIDAQABoCkwJwYJKoZI\
nhvcNAQkOMRowGDAJBgNVHRMEAjAAMAsGA1UdDwQEAwIF4DANBgkqhkiG9w0BAQsF\
nAAOCAQEAVJVTTELGHWoRh8JZt+kx/zO0VnibBq/D6uB405L+Ir80X48Ei9hTLB11\
nAqhSBE+AbEgBhRnEIDBjiXEDcWvC532Omex721kc17ZTzowuD8lOjfQkTHbAmjIi\nnCQNFAPf0D/
zpi6Eync5pi2P//Uj/Yn7oDYYli1t61EZwuQyEu4mbQ1efUnU/SOl\
nAAQtDPhNwATZPmfefjM8+YuzhG70dQvmFAClcFayKM92Zx9khDd/VnLT85YzDULJ\
n8iiHW8dZNblaTsUjKrc73iX2hONZIxw6B3tGCFs8mH9lZlExV7Y2er3t/lO1pdxe\
nSUohEELWcttIxyWnYgxvwaWX4lfx9A\u003d\u003d\n-----END CERTIFICATE
REQUEST-----","reason":"test"}'
```

3.1.7.4 Example Response

```
HTTP/1.1 204 No Content
```