

COMODO
Creating Trust Online®



Comodo Certificate Manager

SSL Web Service API

Comodo CA Limited
3rd Floor, 26 Office Village, Exchange Quay,
Trafford Road, Salford,
Greater Manchester M5 3EQ,
United Kingdom

Table of Contents

Version History	4
1.Introduction	4
2.Authentication	4
2.1.Authentication via Username and Password.....	4
2.2.Authentication via Username and a Client Certificate.....	5
3.Remote Functions	5
3.1.Function for SSL Certificate Renewal.....	5
3.1.1.Arguments.....	6
3.1.2.Return value - 'status code' of operation.....	6
3.2.Function for SSL Certificate Renewal by SSL ID.....	6
3.2.1.Arguments.....	6
3.2.2.Return value - 'status code' of operation.....	6
3.3.Function for Collecting Renewed SSL Certificate.....	6
3.3.1.Arguments.....	7
3.3.2.Return value - SSLRenewResponse.....	7
3.4.Function for SSL Certificate Replacement.....	7
3.4.1.Arguments.....	7
3.4.2.Return value - 'status code' of operation.....	8
3.5.Function for SSL Certificate Enrollment.....	8
3.5.1.Arguments.....	9
3.5.1.1.AuthData type.....	10
3.5.1.2.Server Type.....	10
3.5.2.Return value - 'status code' of operation.....	11
3.6.Function for Checking if Certificate is Available.....	12
3.6.1.Arguments.....	12
3.6.2.Return value - status of certificate availability.....	12
3.7.Function for Collecting Enrolled SSL Certificate.....	13
3.7.1.Arguments.....	13
3.7.2.Return value - SSLCollectResponse.....	13
3.7.3.SSL type.....	14
3.8.Function for SSL Certificate Revocation.....	14
3.8.1.Arguments.....	14
3.8.2.Return value - 'status code' of operation.....	14
3.9.Function for Loading List of Available Certificate Types for Customer.....	15
3.9.1.Arguments.....	15
3.9.2.Return value.....	15
3.9.2.1.CustomerCertType - type for saving information about available customer certificate type.....	15
3.9.2.2.CustomerCertType5 - type for saving information about available customer certificate type.....	16
3.10.Function for changing SSL Certificate External Requester.....	16
3.10.1.Arguments.....	16
3.10.2.Return Value - 'status code' of Operation.....	16
3.11.Utility Function for Getting Short Information about Web Service (name, version, etc.).....	17

Version History

1. Initial Version

- 1.1. Removed extra error codes.
- 1.2. Added 'SSL' type with 'renewID' field. The 'SSLCollectReponse' type contains 'SSL' field now.
- 1.3. Fixed variable's name.
- 1.4. Added 'Invalid ID' return code with 'getCollectStatus' method.
- 1.5. Added 3 methods (enroll5, getCustomerCertTypes5, enrollWithDCV5)
Added 2 types (CustomerCertTypes5, CustomerCertTypeResponse5)
Changed type (CustomerCertTypes contains Integer array)

1. Introduction

Name : EPKIManagerSSLService

Service EPR : <http://cert-manager.com/ws/EPKIManagerSSL> OR
<http://hard.cert-manager.com/ws/EPKIManagerSSL>
<http://cert-manager.com/private/ws/EPKIManagerSSL> OR
<http://hard.cert-manager.com/private/ws/EPKIManagerSSL>

View WSDL : <http://cert-manager.com/ws/EPKIManagerSSL?wsdl> OR
<http://hard.cert-manager.com/ws/EPKIManagerSSL?wsdl> *
<http://cert-manager.com/private/ws/EPKIManagerSSL?wsdl> OR
<http://hard.cert-manager.com/private/ws/EPKIManagerSSL?wsdl> *

Service Description : The Service allows the Administrator to renew and collect renewed SSL certificates, request, collect, and revoke SSL certificates.

* Please use the sub-domain appropriate to your use case.

2. Authentication

To access CCM APIs, you first need to authenticate yourself to the CCM service. You can authenticate via username/password or via username + client certificate. The SSL Web Service API service uses the SOAP protocol.

- [Authentication via Username and Password](#)
- [Authentication via Username and a Client Certificate](#)

2.1. Authentication via Username and Password

Prerequisite

- Users should have CCM login credentials and the correct customer login URI
- For the Web Service API, access must be enabled for the customer by Comodo and for each org/dept by admins on the client side.

The URL for the username/password authentication is: `https://<CCM Server>:<port>/ws/EPKIManagerSSL`

Parameter	Description
<CCM Server>	The address of the CCM server you use. For example, ' <i>cert-manager.com</i> ' or ' <i>hard.cert-manager.com</i> '.
<port>	The default port number is 443.

Example:

`https://cert-manager.com:443/ws/EPKIManagerSSL`

Authentication is performed by sending the AuthData parameter to the web service API. This includes the username, password and Customer URI. After successful authentication, the admin can proceed to the CCM management interface. If authentication is not successful (login and/or password are incorrect, password has expired), the admin will see an error and will be denied access to the SSL Web Service API. The same admin could, however, still authenticate themselves via a client certificate (refer to the [next section](#)).

2.2. Authentication via Username and a Client Certificate

Prerequisite

- Admins should have the Customer URI
- For the Web Service API, access must be enabled for the customer by Comodo and for each org/dept by admins on the client side.
- Admins should have 'Certificate Auth' enabled. The authentication certificate must requested and issued via CCM and active at the moment of authentication.

The URLs for the username/client certificate authentication is: `https://<CCM Server>:<port>/private/ws/EPKIManager.SSL`

Parameter	Description
<CCM Server>	The address of the CCM server you use. For example, ' <i>cert-manager.com</i> ' or ' <i>hard.cert-manager.com</i> '.
<port>	The default port number is 443.

Example:

`https://cert-manager.com:443/private/ws/EPKIManagerSSL`

The certificate must be provided by the admin's client at the time of login. After receiving the authdata parameter (customer URI and username), CCM will verify that the certificate matches the one specified in the 'Certificate Auth' area of the admin's profile. After successful authentication, the admin can proceed to the CCM management interface. If authentication is not successful (username is incorrect, certificate is not correct/revoked), the admin will see an error and will be denied access to the SSL Web Service API. The same admin could, however, still authenticate themselves using the username and password method (see [previous section](#)).

3. Remote Functions

3.1. Function for SSL Certificate Renewal

```
int renew(String renewId)
```

3.1.1. Arguments

Variable Name	Type	Max. Length (chars)	Description
renewId	String	20	Given by Comodo Certificate Manager in notification letter when SSL certificate was issued.

3.1.2. Return value - 'status code' of operation

Status code	Possible Value(s)
If 'status code' < 0	-3 = internal error; -4 = invalid renewId.
If 'status code' = 0	0 = success.

3.2. Function for SSL Certificate Renewal by SSL ID

```
int renewByld(String renewByld)
```

3.2.1. Arguments

Variable Name	Type	Max. Length (chars)	Description
authData	AuthData		Authentication data for access. See section 3.3.1.1.AuthData type
id	String		This is the SSL identifier previously returned by <i>enroll/renewByld</i> functions Displayed in SSL Grid as 'Self Enrollment Certificate ID'

3.2.2. Return value - 'status code' of operation

Status code	Possible Value(s)
If 'status code' < 0	-14 = An unknown error occurred; -16 = Permission denied; -40 = Invalid ID; -100 = Invalid auth data; -110 = The certificate type is invalid.
If 'status code' = 0	0 = success.

3.3. Function for Collecting Renewed SSL Certificate

```
SSLRenewResponse collectRenewed(String renewId, int formatType)
```

3.3.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
renewId	String	20		Given by CCM in notification letter when SSL certificate was issued.
formatType	int	1	0 = X509 PEM Bundle; 1 = X509 PEM Certificate only; 2 = X509 PEM Intermediate certificate only; 3 = PKCS#7 PEM Bundle; 4 = PKCS#7 DER Bundle.	Format of SSL to be returned.

3.3.2. Return value - SSLRenewResponse

SSLRenewResponse - Object that contains collect operation status and SSL Certificate in Byte array if succeed.

Method Name	Possible value(s)
int getErrorCode()	0 = issued; -1 = applied; -2 = certificate error, invalid state; -3 = internal error; -4 = SSL Certificate not exists; -5 = waiting for approval by admin; -6 = admin has declined request.
byte[] getData()	If status code = 0, then certificate in the form of byte array if succeed, <i>null</i> otherwise.

3.4. Function for SSL Certificate Replacement

`int replace (AuthData data, Integer id, string csr, string reason)`

3.4.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type
id	String			This is the SSL identifier previously returned by function <i>enroll</i> . Displayed in SSL Grid as 'Self Enrollment Certificate ID'

csr	String	32767	<p>Subject:</p> <p>The fields may be in any order, (although multiple street addresses, if present, should be in the correct order).</p> <p>Algorithm OID = rsaEncryption (PKCS#1). Size = 512 to 8192 bits.</p> <p>Attributes:</p> <p>Any attributes MAY be present, but they will be ignored if the subject_ fields are used.</p> <p>Signature Algorithm: md5WithRSAEncryption (PKCS#1).</p>	Certificate Signing Request (Base-64 encoded with or without the -----BEGIN xxxxx----- and -----END xxxxx----- header and footer). Allowed values:
reason	string	256		Reason for the Replacement. The reason will be used for audit logging.

3.4.2. Return value - 'status code' of operation

Status code	Possible Value(s)
If 'status code' < 0	-9 = The CSR is not valid Base-64 data; -10 = The CSR cannot be decoded; -11 = The CSR uses an unsupported algorithm; -12 = The CSR has an invalid signature; -13 = The CSR uses an unsupported key size; -14 = An unknown error occurred; -15 = Reason cannot be empty; -16 = Permission denied; -24 = Auth data argument is invalid; -40 = Invalid ID; -100 = Invalid auth data; -101 = Invalid organization auth data; -103 = The type of certificate status is invalid; -105 = Person not found; -106 = EULA is not accepted.
If 'status code' = 0	Operation was successful.

3.5. Function for SSL Certificate Enrollment

Integer enroll (AuthData data, Integer orgId, String secretKey, String csr, String phrase, String

subjAltNames, CustomerCertType certType, Integer numberServers, Integer serverType, Integer term, String comments)

Integer enroll5 (AuthData data, Integer orgId, String secretKey, String csr, String phrase, String subjAltNames, CustomerCertType certType, Integer numberServers, Integer serverType, Integer term, String comments)

Integer enrollWithDCV5 (AuthData data, Integer orgId, String secretKey, String csr, String phrase, String subjAltNames, CustomerCertType certType, Integer numberServers, Integer serverType, Integer term, String comments, String dcv_email)

Integer enrollWithDCV (AuthData data, Integer orgId, String secretKey, String csr, String phrase, String subjAltNames, CustomerCertType certType, Integer numberServers, Integer serverType, Integer term, String comments, String dcv_email)

3.5.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type
orgId	Integer			Organization identifier. Can be obtained from Admin UI - Organizations tab.
secretKey	String	20		Secret Key for SSL is setting in Client Admin UI 'Organization' properties, 'SSL' tab.
csr	String	32767	Subject: The fields may be in any order (although multiple street addresses, if present, should be in the correct order). Algorithm OID = rsaEncryption (PKCS#1). Size = 512 to 8192 bits. Attributes: Any attributes MAY be present, but will be ignored if the subject_ fields are used. Signature Algorithm: md5WithRSAEncryption (PKCS#1)	Certificate Signing Request (Base-64 encoded with or without the -----BEGIN xxxxx----- and -----END xxxxx----- header and footer)
phrase	String	64		Pass phrase for revocation.
subjAltNames	String		Subject Alternative Names separated by ", "	List of Subject Alternative Names.

certType	CustomerCertificateType			Certificate types available for the ordering customer. See description in section 3.7.2 or more details.
numberServers	Integer			Number of servers.
serverType	Integer			Server type of the SSL certificate. See description below in section 3.3.1.2 Server Type .
term(for enroll and enrollWithDCV))	Integer		Term in years.	Term of the SSL certificate.
term(for enroll5 and enrollWithDCV5)	Integer		Terms in days.	Term of the SSL certificate.
comments	String	256		The message that will be attached to the certificate.
dcv_email	String	320		Email string is for DCV validation email.

3.5.1.1. AuthData type

Name	Description
setLogin(String value)	Set login name for account within CCM. This is the login of the Admin with role 'SSL Admin' within CCM account.
setPassword(String value)	Set password for account within CCM. This is the password of the Admin with role 'SSL Admin' within CCM account.
setURI(String value)	URI for logging into account within CCM.

3.5.1.2. 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 (Javasoftware / Sun)
11	Lotus Domino
12	Lotus Domino Go!
13	Microsoft IIS 1.x to 4.x
14	Microsoft IIS 5.x and later
15	Netscape Enterprise Server
16	Netscape FastTrac
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

3.5.2. Return value - 'status code' of operation

Status code	Possible Value(s)
If 'status code' < 0	-3 = The 'User name' argument is invalid. -7 = Country is not a valid ISO-3166 country! -9 = The CSR is not valid Base-64 data!

	<ul style="list-style-type: none"> -10 = The CSR cannot be decoded! -11 = The CSR uses an unsupported algorithm! -12 = The CSR has an invalid signature! -13 = The CSR uses an unsupported key size! -14 = An unknown error occurred! -16 = Permission denied! -31 = The email is not a valid email. -32 = The two phrase should be the same! -33 = The Comodo certificate type is invalid! -34 = The secret key is invalid! -35 = The server type is invalid! -36 = The term is invalid for customer type! - 100 = Invalid authentication data for customer - 101 = Invalid authentication data for customer Organization - 110 = Domain is not allowed for customer - 111 = Domain is not allowed for customer Organization - 120 = Customer configuration is not allowed the requested action
If 'status code' > 0	<p>SSL identifier.</p> <p>It will be used for certificate collecting/revoking.</p>

3.6. Function for Checking if Certificate is Available

Integer getCollectStatus(AuthData data, Integer id)

3.6.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type .
id	Integer		Any SSL identifier previously returned to your account.	This is the SSL identifier previously returned by function <i>enroll</i> .

3.6.2. Return value - status of certificate availability

Possible Value(s)
<ul style="list-style-type: none"> 1 = Certificate available 0 = Certificate being processed by Comodo -14 = An unknown error occurred! -16 = Permission denied! -40 = Invalid ID

- 100 = Invalid authentication data for customer
- 101 = Invalid authentication data for customer Organization
- 110 = Domain is not allowed for customer
- 111 = Domain is not allowed for customer Organization
- 120 = Customer configuration is not allowed the requested action

3.7. Function for Collecting Enrolled SSL Certificate

`SSLCollectResponse collect(AuthData data, Integer id, int formatType)`

3.7.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type
id	Integer		Any SSL identifier previously returned to your account.	This is the SSL identifier previously returned by function <i>enroll</i> .
formatType	int	1	0 = X509 PEM Bundle; 1 = X509 PEM Certificate only; 2 = X509 PEM Intermediate certificate only; 3 = PKCS#7 PEM Bundle; 4 = PKCS#7 DER Bundle.	Allowed formats for downloading of SSL.

3.7.2. Return value - SSLCollectResponse

SSLCollectResponse - Object that contains collect operation status and SSL Certificate in Base-64 if succeed.

Method Name	Possible Value(s)
int getStatusCode()	1 = Certificate Available 2 = Certificates Attached 0 = Being processed by Comodo -14 = An unknown error occurred! -16 = Permission denied! -20 = The certificate request has been rejected! -21 = The certificate has been revoked! -22 = Still awaiting payment! - 100 = Invalid authentication data for customer - 101 = Invalid authentication data for customer Organization - 110 = Domain is not allowed for customer

	<ul style="list-style-type: none"> - 111 = Domain is not allowed for customer Organization - 120 = Customer configuration is not allowed the requested action
SSL getSSL()	If status code = 2, then the special object with the certificate in Base-64 if succeed, <i>null</i> otherwise. See section 3.5.3.SSL type

3.7.3. SSL type

Method Name	Description
String getRenewID()	Given by CCM when SSL certificate was issued. This code may be used for renewing the certificate.
String getCertificate()	The certificate in Base-64

3.8. Function for SSL Certificate Revocation

`Integer revoke(AuthData data, Integer id, String reason)`

3.8.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type .
id	Integer		Any SSL identifier previously returned to your account.	This is the SSL identifier previously returned by function <i>enroll</i> .
reason	String	256		Revocation reason for audit logging. Empty String allowed.

3.8.2. Return value - 'status code' of operation

Possible Value(s)
<ul style="list-style-type: none"> 0 = Successful -14 = An unknown error occurred! -16 = Permission denied! - 100 = Invalid authentication data for customer - 101 = Invalid authentication data for customer Organization - 110 = Domain is not allowed for customer - 111 = Domain is not allowed for customer Organization - 120 = Customer configuration is not allowed the requested action

3.9. Function for Loading List of Available Certificate Types for Customer

CustomerCertTypeResponse getCustomerCertTypes (AuthData authData)

CustomerCertTypeResponse5 getCustomerCertTypes5 (AuthData authData)

CustomerCertTypeResponse getCustomerCertTypesByOrg (AuthData authData, Integer orgId)

CustomerCertTypeResponse5 getCustomerCertTypesByOrg5 (AuthData authData, Integer orgId)

3.9.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type
orgId	Integer			The function returns object that contains array available certificate types for the specific Organization

3.9.2. Return value

CustomerCertTypeResponse - Object that contains array available customer certificate types (see description of CustomerCertType below).

CustomerCertTypeResponse5 - Object that contains array available customer certificate types (see description of CustomerCertType5 below).

Field Name	Possible Value(s)
CustomerCertType[] getTypes()	If customer does not have available certificate type - result array will be empty.
CustomerCertType5[] getTypes()	If customer does not have available certificate type - result array will be empty.
int getStatusCode()	0 = Successful -14 = An unknown error occurred! -16 = Permission denied!

3.9.2.1. CustomerCertType - type for saving information about available customer certificate type

Variable Name	Description
int getId()	The service customer cert type identifier.
String getName()	Name of this certificate type. For example "InstantSSL"
String[] getTerms	List of available terms for this customer certificate type (In Years).

For backward compatibility during the certificate enrollment process, one can specify the number of years instead of the number of days.

3.9.2.2. CustomerCertType5 - type for saving information about available customer certificate type

Variable Name	Description
String[] getTerms	List of available terms for this customer certificate type (In Days).

3.10. Function for changing SSL Certificate External Requester

Integer updateRequesterExt(AuthData data, Integer id, String[] requesterExt)

3.10.1. Arguments

Variable Name	Type	Max. Length (chars)	Allowed Values	Description
authData	AuthData			Authentication data for access. See section 3.3.1.1.AuthData type
id	Integer		Any SSL identifier previously returned to your account.	This is the SSL identifier previously returned by function <i>enroll</i> .
requesterExt	String[]		Array of email addresses	Array of email addresses for external requester(s)

3.10.2. Return Value - 'status code' of Operation

Possible Value(s)
0 = SUCCESSFUL; -14 = An unknown error occurred; -16 = Permission denied; -31 = External Requester is invalid; -100 = Invalid auth data; -101 = Invalid organization auth data; -106 = EULA is not accepted; -110 = Domain is not allowed for customer;

-111 = Domain is not allowed for organization;
-112 = KU/EKU template is not allowed for customer;
-113 = KU/EKU template is not allowed any more;
-114 = Client Cert Type is not available for organization;
-115 = Domain is not DCV validated (while 'Enforce DCV for S/MIME' is ON);
-120 = Customer configuration is not allowed the desired action.

3.11. Utility Function for Getting Short Information about Web Service (name, version, etc.)

`String getWebServiceInfo()`

About Comodo

The Comodo organization is a global innovator of cybersecurity solutions, protecting critical information across the digital landscape. Building on its unique position as the world's largest certificate authority, Comodo authenticates, validates and secures networks and infrastructures from individuals to mid-sized companies to the world's largest enterprises. Comodo provides complete end-to-end security solutions across the boundary, internal network and endpoint with innovative technologies solving the most advanced malware threats, both known and unknown. With global headquarters in Clifton, New Jersey, and branch offices in Silicon Valley, Comodo has international offices in China, India, the Philippines, Romania, Turkey, Ukraine and the United Kingdom. For more information, visit comodo.com.

Comodo CA Limited

3rd Floor, 26 Office Village, Exchange Quay, Trafford
Road, Salford, Greater Manchester M5 3EQ,
United Kingdom.

Tel : +44 (0) 161 874 7070

Fax : +44 (0) 161 877 1767

Email: EnterpriseSolutions@Comodo.com

Comodo Security Solutions, Inc.

1255 Broad Street
Clifton,

NJ 07013

United States

Tel: +1.877.712.1309

Tel: +1.888.256.2608

For additional information on Comodo - visit <http://www.comodo.com>.