



Certificates

A signed certificate is a digital identity document that enables both server and client to authenticate each other. Certificates are used with HTTPS protocol to encrypt secure information transmitted over the internet. A certificate can be generated or procured from a third party Certificate Authority (CA). Generated certificates can be self-signed or signed by a trusted third-party CA. A certificate contains information such as user name, expiration date, a unique serial number assigned to the certificate by a trusted CA, the public key, and the name of the CA that issued the certificate.

To Create a Certificate

URL: /v1/certificates

Method: POST

Description: Creates a self-signed certificate with the given values.

Parameter Name	Data Type	Mandatory	Description
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
key_type	String	Optional	The key/algorithm used in the certificate. The values include: <ul style="list-style-type: none"> rsa ecdsa <p>Note: By default, key_type is <i>rsa</i>. If the key used in the certificate is ECDSA, then specify <i>ecdsa</i> as key_type.</p>
common_name	Alphanumeric	Yes	The domain name (DN) of the web server for which you want to generate the certificate.
san_certificate	Alphanumeric	Yes	The Subject Alternative Names (SAN) that needs to be associated with the certificate. The values include: <ul style="list-style-type: none"> DNS Email URI IP <p>Example:</p> <ul style="list-style-type: none"> DNS: barracuda.yourdomain.com URI, Enter a Fully Qualified Domain Name (FQDN). Example: http://www.barracuda.com Email: rwatson@barracuda.com IP: 192.168.7.1
country_code	Alphabetic	Yes	The two-letter country code of the location of the organization.
state	Alphabetic	Optional	The full name of the state or province of the location of the organization.
city	Alphabetic	Optional	The full name of the locality (city) where the organization is located.
organization_name	Alphanumeric	Optional	The legally registered name of the organization or company.
organization_unit	Alphanumeric	Optional	The department or unit within the organization.
key_size	Enumeration	Yes	The private key size for the certificate in bits. The enumerated values include: <ul style="list-style-type: none"> 1024 2048 4096



Parameter	Type	Optional	Description
curve_type	Enumeration	Optional	<p>The elliptic curve. The enumerated values include:</p> <ul style="list-style-type: none"> • secp256r1 • secp384r1 • secp512r1 <p>Specify whether to lock the Private Key corresponding to this certificate or not. The values include:</p> <ul style="list-style-type: none"> • yes • no
allow_private_key_export	String	Yes	<p>Normally, certificates are downloaded in PKCS12 format which includes the Private Key and Certificate. When a key is locked, you can only download the certificate in PEM format. Also, you cannot take a backup when the Private Key is locked.</p> <p>Note: This option is valid only for created and uploaded (generated and signed by a trusted CA) certificates.</p>

Request:

```
curl http://10.11.28.179:8000/restapi/v1/certificates -
u'eyJldCI6IjEzNDg3MTYzNzkwIiwicGFzc3dvcmQiOiJkODANjU3ZWE0NjFIZDZjMmI0YjNiYjVm\nNmJkN2I0ZSIsInVzZX
liOiJhZG1pbij9\n:' -X POST -H Content-Type:application/json -
d'{"name":"Certificate13","common_name":"barracuda.yourdomain.com","country_code":"US","state":"Californi
a","city":"Campbell","organization_name":"BarracudaNetworks","organization_unit":"Engineering","key_size":"1
024","allow_private_key_export":"yes","san_certificate":["IP:10.11.19.76","DNS:mydomain","Email:sd@gmail.co
m"],"URI:https://example.org/absolute/URI/with/absolute/path/to/resource.txt"},"key_type":"ecdsa","curve_type":"
secp256r1"}'
```

Response:

```
{"id":"certificate1","token":"eyJldCI6IjE0ODcxNDQ5MjQiLCJwYXNzd29yZCI6IjFIMjMmI0ZGQzNzFhN2VjYTc1OGE2
OWY1\nYzM5OWJmIiwidXNlciI6ImFkbWluIn0=\n"}
```

Request:

```
curl http://10.11.28.179:8000/restapi/v1/certificates -
u'eyJldCI6IjEzNDg3MTYzNzkwIiwicGFzc3dvcmQiOiJkODANjU3ZWE0NjFIZDZjMmI0YjNiYjVm\nNmJkN2I0ZSIsInVzZX
liOiJhZG1pbij9\n:' -X POST -H Content-Type:application/json -
d'{"name":"Certificate3","common_name":"barracuda.yourdomain.com","country_code":"US","state":"California
","city":"Campbell","organization_name":"BarracudaNetworks","organization_unit":"Engineering","key_size":"10
24","allow_private_key_export":"yes","san_certificate":["IP:10.11.19.76","DNS:mydomain","Email:sd@gmail.com
"],"URI:https://example.org/absolute/URI/with/absolute/path/to/resource.txt"},"key_type":"ecdsa","curve_type":"s
ecp256r1"}'
```

Response:

```
{"id":"Certificate3","token":"eyJldCI6IjE0ODcxNDQ5MjQiLCJwYXNzd29yZCI6IjFIMjMmI0ZGQzNzFhN2VjYTc1OGE2
OWY1\nYzM5OWJmIiwidXNlciI6ImFkbWluIn0=\n"}
```

To Upload a Signed Certificate

URL: /v1/certificates?upload=signed

Method: POST

Description: Uploads the given signed (pem or pkcs12) certificate.



Parameter Name	Data Type	Mandatory	Description
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
type	String	Yes	Select the certificate type. The values include: <ul style="list-style-type: none"> • pkcs12 • pem
key_type	String	Optional	The key/algorithm used in the certificate. The values include: <ul style="list-style-type: none"> • rsa • ecdsa Note: By default, key_type is <i>rsa</i> . If the key used in the certificate is ECDSA, then specify <i>ecdsa</i> as key_type .
signed_certificate	String	Yes	The path and name of the signed certificate file that needs to be uploaded. <p>The values include:</p> <ul style="list-style-type: none"> • yes - If the CSR corresponding to this certificate was generated on the Barracuda Web Application Firewall. • no - Upload the private key corresponding to this certificate in the "key" field.
assign_associated_key	String	Conditional	<p>Note: Required ONLY when the certificate being uploaded is in PEM format.</p> <p>The path and name of the corresponding private key for the signed certificate being uploaded.</p>
key	String	Conditional	<p>Note: Required ONLY when the certificate being uploaded is in PEM format.</p> <p>The path and name of the intermediary CA certificate file that needs to be uploaded.</p> <p>Note: If your certificate is signed by a trusted CA, the certificate should be uploaded in the following order:</p> <ul style="list-style-type: none"> • Leaf certificate • Intermediate certificate(s) • Root CA certificate
intermediary_certificate	String	Conditional	<p>This is required ONLY when the certificate being uploaded is in PEM format.</p> <p>Determines whether to export the private key corresponding to the certificate or not. The values include:</p> <ul style="list-style-type: none"> • yes - To export the private key corresponding to the certificate. • no - To lock the private key. In this case, the certificate can be downloaded only in PEM format, and backup of system configuration cannot be taken.
allow_private_key_export	String	Yes	<p>The password used to generate the PKCS #12 token for the signed certificate being uploaded.</p>
password	Alphanumeric	Conditional	<p>Note: Required ONLY when the certificate being uploaded is PKCS12 Token.</p>

Example: Uploading a Signed Certificate in PEM Format

Request:

```
curl -i -F name=cert10 -F signed_certificate=@/home/gireesh/RestAPI/abc_bc_com.crt -F
key=@/home/gireesh/RestAPI/abc_bc_com_key.pem -F assign_associated_key=no -F key_type=rsa -F type=pem
```



```
-F allow_private_key_export=yes http://10.11.25.108:8000/restapi/v1/certificates?upload=signed -u
'eyJldCI6IjE0NzQwMTg5NjciLCJwYXNzd29yZCI6IjJhMWVlMDhmNTdlOTY2NjRiZTE4Y2VhOWRh\nMWJmZjA5IiwidXNl
cil6ImFkbWluln0=\n:'
```

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Fri, 03 Jul 2015 10:46:10 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

```
{"id":"cert10","token":"eyJldCI6IjE0NzQwMTg5NjciLCJwYXNzd29yZCI6IjJhMWVlMDhmNTdlOTY2NjRiZTE4Y2VhOW
Rh\nMWJmZjA5IiwidXNlci6ImFkbWluln0=\n"}
```

Example 1: Uploading a Signed Certificate in PKCS12 Token Format

Request:

```
curl -i -F name=Cert3 -F signed_certificate=@/home/gireesh/RestAPI/Barracuda.p12 -F type=pkcs12 -F
key_type=rsa -F allow_private_key_export=yes -F password='password1231'
http://10.11.25.108:8000/restapi/v1/certificates?upload=signed -u
'eyJldCI6IjE0NzQwMTgyNzEiLCJwYXNzd29yZCI6IjU5NmI5MThkZDNlNzMyNzdmZmQ2NmY3ZWZh\nMmE2Y2QyIiwidXNlci6ImFkbWluln0=\n:'
```

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Tue, 19 Nov 2013 12:31:56 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

```
{"id":"Cert3","token":"eyJldCI6IjE0NzQwMTgyNzEiLCJwYXNzd29yZCI6IjU5NmI5MThkZDNlNzMyNzdmZmQ2NmY3ZWZh\nMmE2Y2QyIiwidXNlci6ImFkbWluln0=\n"}
```

Example 2: Uploading a Signed Certificate in PKCS12 Token Format

Request:

```
curl -i -F name=cedr -F type=pkcs12 -F signed_certificate=@/root/raj_ssl/cert/ecdsa1.p12 -F key_type=ecdsa -F
password=123456 -F allow_private_key_export=yes
http://10.11.25.107:8000/restapi/v1/certificates?upload=signed -u
```

Certificates



'eyJldCI6IjE0Mzg5MzU5NzAiLCJwYXNzd29yZCI6Ijg0YTg0YzRkMDIhYWlzcZmEwOGEyNmU1ZDg4\nYzRjMTNkliwidXNlcil6ImFkbWluln0=\n:'

Response:

HTTP/1.1 201
Server: BarracudaHTTP 4.0
Date: Fri, 24 Jul 2015 11:21:04 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
{\"id\":\"cedr\",\"token\":\"eyJldCI6IjE0Mzg5MzY4NjliLCJwYXNzd29yZCI6ImQxYjYxMGRIZGI1OGRiYzY1MTJiYzcxYmM2\nMDI4MDFiliwidXNlcil6ImFkbWluln0=\n\"}

To Upload a Trusted (CA) Certificate

Use this API to upload a Certificate Authority's (CA) certificate, a trusted certificate that acts as a root CA certificate for authenticating the client certificates. Any client certificate signed by the trusted certificate is valid and allowed access without further validation.

URL: /v1/certificates?upload=trusted

Method: POST

Description: Uploads the given trusted CA certificate.

Parameter Name	Data Type	Mandatory	Description
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
trusted_certificate	String	Yes	The path and name of the trusted CA certificate that needs to be uploaded.

Example:

Request:

curl -i -F name=Trusted_Cert -F trusted_certificate=@/home/certs/rootca.pem
http://192.168.0.1:8000/restapi/v1/certificates?upload=trusted -u
'eyJldCI6IjEzODQyOTQyMzU1LCJwYXNzd29yZCI6IjQyZW50NjIjMTc5MjIhMDZkMzZmZmI5NjYz\nODMyOTk0liwidXNlcil6ImFkbWluln0=\n:'

Response:

HTTP/1.1 201
Server: BarracudaHTTP 4.0
Date: Tue, 12 Nov 2013 06:46:11 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
{\"id\":\"Trusted_Cert\",\"token\":\"eyJldCI6IjEzODQyOTU3MDgiLCJwYXNzd29yZCI6ImRhNTU0OTFINDY5Y2U0NDA4Njc



xOTMzZGFj\nNzlyYWZkliwidXNlciI6ImFkbWluIn0=\n"}

To Upload a Trusted Server Certificate

Use this API to upload a Certificate Authority's (CA) certificate, a trusted certificate that acts as a root CA certificate for authenticating back-end server certificates. Any back-end server certificate signed by the uploaded trusted certificate is valid and allowed access without further validation.

URL: /v1/certificates?upload=trusted_server

Method: POST

Description: Uploads the given trusted server certificate.

Parameter Name	Data Type	Mandatory	Description
Input Parameters:			
name	Alphanumeric	Yes	The name of the certificate.
trusted_server_certificate	String	Yes	The path and name of the trusted server certificate that needs to be uploaded.

Example:

Request:

```
curl -i -F name=Server_cert1 -F trusted_server_certificate=@/home/certs/rootca.pem
http://192.168.0.1:8000/restapi/v1/certificates?upload=trusted_server -u
'eyJldCI6IjEzODQyOTQyMzUjLCJwYXNzd29yZCI6IjQyZW5jYzIjMTc5MjIhMDZkMzZmI5NjYz\nODMyOTk0IiwidXNlciI6ImFkbWluIn0=\n:'
```

Response:

HTTP/1.1 201

Server: BarracudaHTTP 4.0

Date: Tue, 12 Nov 2013 06:49:45 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

```
{"id":"Server_cert1","token":"eyJldCI6IjEzODQyOTU5NjEiLCJwYXNzd29yZCI6ImNjN2ZjOWNiNWQ3NTJlNDM1MGJiNjk2YmQz\nNzZlOGU0IiwidXNlciI6ImFkbWluIn0=\n"}
```

To Download a Signed Certificate

Use this API to download a signed certificate. For more information on certificates, refer to [Certificate Management](#).

In the web interface of the Barracuda Web Application Firewall, the certificate is saved as a PKCS12 token (p12). Therefore, it is recommended to append **.p12** extension next to the certificate in the API call.



URL: /v1/certificates/{certificate_name}

Method: GET

Description: Downloads the given certificate.

Parameter Name	Data Type	Mandatory	Description
----------------	-----------	-----------	-------------

Input Parameters:

download	Binary	Yes	Determines whether the certificate needs to be downloaded or not. One (1) - to download the certificate.
encrypt_password	Alphanumeric	Yes	The password to save the certificate.

Example:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u
'eyJldCI6IjEzOTM1MDE3MTAiLCJwYXNzd29yZCI6IjU2YjliNGY2MzFIZjg5ZmU5Y2ZkNGZINTYy\nNDIzODM5liwidXNlci
I6ImFkbWluIn0=\n:' -H Content-Type:application/json -X GET -o rft.p12 -G -d download=1 -d
encrypt_password=123456
```

Response:

```
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 2485 0 2485 0 0 7102 0 699 0 --:-- --:-- --:-- 7223
```

To Download a Trusted (CA) Certificate or Trusted Server Certificate

Use this API to download a trusted (CA) certificate or trusted server certificate.

In the web interface of the Barracuda Web Application Firewall, a trusted (CA) certificate or trusted sever certificate is saved in PEM format. Therefore, it is recommended to append **.pem** extension next to the certificate in the API call.

URL: /v1/certificates/{certificate_name}

Method: GET

Description: Downloads the given certificate.

Parameter Name	Data Type	Mandatory	Description
----------------	-----------	-----------	-------------

Input Parameters:

download	Binary	Yes	Determines whether the certificate needs to be downloaded or not. One (1) - to download the certificate.
----------	--------	-----	---

Example:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates/server_cert1 -u
'eyJldCI6IjEzOTM1MDM1NDYiLCJwYXNzd29yZCI6IjU2YjliNGY2MzFIZjg5ZmU5Y2ZkNGZINTYy\nNDIzODM5liwidXNlci
I6ImFkbWluIn0=\n:' -H Content-Type:application/json -X GET -o raj.pem -G -d download=1
```

Response:

Certificates
7 / 10



% Total	% Received	% Xferd	Average	Speed	Time	Time	Time	Current
Dload	Upload	Total	Spent	Left	Speed			

100	1334	0	1334	0	0	7102	0	1537	0	--:--	--:--	--:--	1543
-----	------	---	------	---	---	------	---	------	---	-------	-------	-------	------

To Retrieve Certificates

URL: /v1/certificates
/v1/certificates/{certificate_id}

Method: GET

Description: Lists all certificates if "certificate_id" is not specified.

Parameter Name	Data Type	Mandatory	Description
----------------	-----------	-----------	-------------

Input Parameters:

parameters	Alphanumeric	Optional	Any specific parameter name that needs to be retrieved. See <i>Example 3</i> .
------------	--------------	----------	--

Example 1:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates -u
'eyJldCI6IjEzODYxNzAzNTliLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTly\nODljMzRjIiwidXNlciI6I
mFkbWluln0=\n:' -X GET
```

Response:

```
{ "parameters": null, "object": "Certificates", "data": [ { "expiry": "Dec 1 06:06:16 2023
GMT\n", "common_name": "barracuda.yourdomain.com", "services": "No
Service", "private_key": "exportable", "name": "Cert_cr_1", "type": "created_certificate" }, { "expiry": "Dec 1 06:06:25
2023
GMT\n", "common_name": "waf4.bc.com", "services": "ss1", "private_key": "exportable", "name": "cert_cr_2", "type": "
created_certificate" }, { "expiry": "Dec 1 06:06:34 2023 GMT\n", "common_name": "waf.bc.com", "services": "No
Service", "private_key": "exportable", "name": "cert_cr_3", "type": "created_certificate" }, { "expiry": "Dec 1 06:07:02
2023 GMT\n", "common_name": "adc.bc.com", "services": "No
Service", "private_key": "not_exportable", "name": "cert_cr_4", "type": "created_certificate" }, { "expiry": "Dec 31
23:59:59 2013 GMT\n", "common_name": "gdfews-globalenergy-stg.gdfsuez.com", "services": "No
Service", "private_key": "not_exportable", "name": "chained_6", "type": "uploaded_certificate" }, { "expiry": "Jul 25
12:04:51 2014 GMT\n", "common_name": "wafqa.net", "services": "No
Service", "private_key": "not_exportable", "name": "cert9", "type": "uploaded_certificate" }, { "expiry": "Dec 31
23:59:59 2013 GMT\n", "common_name": "gdfews-globalenergy-stg.gdfsuez.com", "services": "No
Service", "private_key": "exportable", "name": "chained_68", "type": "uploaded_certificate" }, { "expiry": "Jul 25
11:57:11 2014 GMT\n", "common_name": "CN", "services": "No
Service", "name": "ca2", "type": "trusted_certificates" }, { "expiry": "Jan 22 13:22:28 2016
GMT", "common_name": "wafqa-1", "services": "N/A", "name": "svr_cert2", "type": "trusted_server_certificates" } ], "tok
en": "eyJldCI6IjEzODYxNzAzNTliLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTly\nODljMzRjIiwidXNlciI6I
dXNlciI6ImFkbWluln0=\n" }
```

Example 2:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u
'eyJldCI6IjEzODYxNzAzNTliLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTly\nODljMzRjIiwidXNlciI6I
dXNlciI6ImFkbWluln0=\n'
```




```
mFkbWluln0=\n:' -X GET
```

Response:

```
{"expiry":"Dec 1 06:06:16 2023 GMT\n","common_name":"barracuda.yourdomain.com","services":"No Service","private_key":"exportable","name":"Cert_cr_1","type":"created_certificate","token":"eyJldCI6IjEzODYyNzk0MDYiLCJwYXNzd29yZCI6IjRiYmYyMDQyNTQ5M2I2Yjc3MDU1ZjY3MWE3\nZDFhMTQ0IiwidXNlciI6ImFkbWluln0=\n"}\n"}
```

Example 3:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates -u 'eyJldCI6IjEzODYxNzAzNTliLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTYiY\nODIjMzRjIiwidXNlciI6ImFkbWluln0=\n:' -X GET -G -d parameters="name,type"
```

Response:

```
{"parameters":{"name,type","object":"Certificates","data":[{"name":"Cert_cr_1","type":"created_certificate"}, {"name":"cert_cr_2","type":"created_certificate"}, {"name":"cert_cr_3","type":"created_certificate"}, {"name":"cert_cr_4","type":"created_certificate"}, {"name":"chained_6","type":"uploaded_certificate"}, {"name":"cert9","type":"uploaded_certificate"}, {"name":"chained_68","type":"uploaded_certificate"}, {"name":"ca2","type":"trusted_certificates"}, {"name":"svr_cert2","type":"trusted_server_certificates"}], "token":"eyJldCI6IjEzODYyNzk0MDQlLCJwYXNzd29yZCI6IjFIYTg3MjIjZGQ3NGIwZWlzMjhhY2E1MDJj\nMmYxMWUyIiwidXNlciI6ImFkbWluln0=\n"}
```

To Delete a Certificate

URL: /v1/certificates/{certificate_id}

Method: DELETE

Description: Deletes the given certificate.

Example:

Request:

```
curl http://192.168.0.1:8000/restapi/v1/certificates/Cert1 -u 'eyJldCI6IjEzODYxNzAzNTliLCJwYXNzd29yZCI6IjZiNTc5NDZiNWU0YjM3NTNhZDZhM2RjYTYiY\nODIjMzRjIiwidXNlciI6ImFkbWluln0=\n:' -X DELETE
```

Response:

```
{"msg":"Successfully deleted","token":"eyJldCI6IjEzODYyNzk0NzciLCJwYXNzd29yZCI6ImRiODM5NDE4NGE1YmVIMWY5NDE3ZDM5OTI5\nYjExZTE4IiwidXNlciI6ImFkbWluln0=\n"}
```

