# Certificate View API

User Guide

Version 2.8.3

October 19, 2021

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100

# Table of Contents

# Preface

This user guide is intended for application developers who will use the Qualys Certificate View API.

## About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com.

## Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.

# Chapter 1 - Get Started

Qualys API Framework - Learn the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

Introduction to Certificate View API Paradigm - Get tips on using the Curl command-line tool to make API requests. Every API request must authenticate using a JSON Web Token (JWT) obtained from the Qualys Authentication API.

### Get API Notifications

Subscribe to our API Notifications RSS Feeds for announcements and latest news.

**From our Community**

Join our Community

API Notifications RSS Feeds

## Qualys API Framework

The Qualys Certificate View API uses the following framework.

### Request URL

The URL for making API requests respects the following structure:

https://<baseurl>/<module>/<object>/<object_id>/<operation>

where the components are described below.

| | |
|---|---|
| <baseurl> | The Qualys API server URL that you should use for API requests depends on the platform where your account is located. The base URL for Qualys US Platform 1 is: https:///gateway.qg1.apps.qualys.com |
| <module> | The API module. For the Certificate View API, the module is: “certview”. |
| <object> | The module specific object. |
| <object_id> | (Optional) The module specific object ID, if appropriate. |
| <operation> | The request operation, such as count. |

## Qualys API Gateway URL

The Qualys API URL you should use for API requests depends on the Qualys platform

Click here to identify your Qualys platform and get the API URL

This documentation uses the API gateway URL for Qualys US Platform 1 (https://gateway.qg1.apps.qualys.com) in sample API requests. If you're on another platform, please replace this URL with the appropriate gateway URL for your account.

# Introduction to Certificate View API Paradigm

## Authentication

You must authenticate to the Qualys Cloud Platform using Qualys account credentials (user name and password) and get the JSON Web Token (JWT) before you can start using the Certificate View APIs. Use the Qualys Authentication API to get the JWT.

For example,

```
curl -X POST https://gateway.qg1.apps.qualys.com/auth -d
"username=value1&password=passwordValue&token=true&permissions=tru
e" -H "ContentType: application/x-www-form-urlencoded"
```

where gateway.qg1.apps.qualys.com is the base URL to the Qualys API server where your account is located.

- **username** and **password** are the credentials of the user account for which you want to fetch Certificate View data

- **token** should be true

- **permissions** should be true

- **Content-Type** should be "application/x-www-form-urlencoded"

The Authentication API returns a JSON Web Token (JWT) which you can use for authentication during Certificate View API calls. The token expires in 4 hours. You must regenerate the token to continue using the Certificate View API.

## Using Curl

**Curl** is a multi-platform command-line tool used to transfer data using multiple protocols. This tool is supported on many systems, including Windows, Unix, Linux and Mac. In this document Curl is used in the examples to build Qualys API requests using the HTTP over SSL (https) protocol, which is required.

Want to learn more? Visit https://curl.haxx.se/

The following Curl options are used according to different situations:

| Option | Description |
|---|---|
| -X “POST” | The POST method is required for all Certificate View API requests. |
| -H “Authorization: Bearer <token>” | This option is used to provide a custom HTTP request header parameter for authentication. Provide the JSON Web Token (JWT) received from Qualys authentication API in the following format: `Authorization: Bearer <token>` For information about Qualys authentication API, see Authentication. |

The sample below shows a typical Curl request using options mentioned above and how they interact with each other.

```
curl -X POST 'https://gateway.qg1.apps.qualys.com/auth' -H 'Content-Type: application/x-www-form-urlencoded' -d 'username=john_doe&password=john_doe&token=true&permissions=true'
```

# Chapter 2 - Certificate API

Use these API functions to retrieve a list of certificates based on an input filter query (List Certview Certificates) and to retrieve asset details of a specific certificate having more than 1000 assets (List Assets for a Certificate).

## List Certview Certificates

**/certview/v1/certificates**

[POST]

Use these API functions to List Certview Certificates to retrieve a list of certificates based on an input filter query and list. The response contains certificate details including associated host information and SSL/TLS related vulnerabilities and grades.

### Input Parameters

| | |
|---|---|
| filter (String) | (Optional) Filter the events list by providing a query using Qualys syntax. Refer to the How to Search topic in the online help for assistance with creating your query.<br>For example - expiryGroup: Expired<br>Refer to the list of tokens you can use to build the query: Search tokens |
| pageNumber (Integer) | (Optional)The page to be returned. Starts from zero. |
| pageSize (Integer) | (Optional) Provide the number of records per page to be included in the response.<br>Default: 10.<br>Maximum: 200<br>For example, the total result set is 50 assets. If the page size is specified as 10, then the result is divided in 5 pages with 10 assets each. |
| sort (String) | (Optional)Sort the results using a Qualys token.<br>For example - [{"lastFound":"desc"}] |
| Authorization (String) | (Required) Authorization token to authenticate to the Qualys Cloud Platform.<br>Prepend token with "Bearer" and one space. For example - Bearer authToken |

| | |
|---|---|
| certificateDetails (String) | (Optional) Define the level of certificate attributes you want to list. Default value **basic** is used to fetch commonly used attributes. Use value **extended** to fetch these additional attributes:<br>- Serial number<br>- Auth Key Identifier<br>- Subject Key Identifier<br>- Key Usage<br>- Base64 certificate<br>- Enhanced Key Usage<br><br>The enhancedKeyUsage attribute returns a list of OIDs in the EKU attribute of the certificate.<br>Some of the most commonly used OIDs are:<br><br>1.3.6.1.5.5.7.3.1—Server Authentication<br>1.3.6.1.5.5.7.3.2—Client Authentication<br>1.3.6.1.5.5.7.3.3—Code Signing<br>1.3.6.1.5.5.7.3.4—Email Protection<br>1.3.6.1.5.5.7.3.8—Time Stamping<br>1.3.6.1.5.5.7.3.9—OCSP Signing<br><br>For more information refer to http://www.oid-info.com/ |

**Notes**:

-This API supports both new and old query format. For more details refer to Query Example.

- If you want to generate a CSV report for more than 10000 certificates, use scheduled reports from Qualys Cloud Platform.

- If the data you are looking for is not available in CSV reports, use additional filter instead of requesting all certificates.

You can use the following filters for better results:

-Last Found Date < 1 month/3months/6 months

-Expiration Date < 1 yr/ between 1yr and 2 yrs/etc

-Approved vs Unapproved CAs vs Self-signed CAs

## Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

## Sample with all parameters defined

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{ \"filter\" : \"subject.name:www.qualys.com\", \"pageNumber\":
```

```
0, \"pageSize\" : 1}" -H  "Authorization: Bearer <JWT Token>"
```

Response:


```
[
    {
        "keySize": 2048,
        "subject": {
            "organization": "Qualys, Inc.",
            "locality": "Foster City",
            "name": "www.qualys.com",
            "state": "California",
            "country": "US",
            "organizationUnit": []
        },
        "validFrom": 1596067200000,
        "signatureAlgorithm": "SHA256withRSA",
        "issuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
            "certhash":
"403e062a2653059113285baf80a0d4ae422c848c9f78fad01fc94bc5b87fef1a"
,
            "locality": ""
        },
        "rootissuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
            "certhash":
"7431e5f4c3c1ce4690774f0b61e05440883ba9a01ed00ba6abd7806ed3b118cf"
,
            "locality": ""
        },
        "instanceCount": 1,
        "dn": "CN=www.qualys.com, O=\"Qualys, Inc.\", L=Foster
City, ST=California, C=US, serialNumber=3152140,
StateOrProvince=Delaware, CountryName=US, businessCategory=Private
```

```
Organization",
        "certhash":
"61ffdf5ec74189b0f6f256fc42fe858bb04c1862c0f4cb9ec9d5f9bf4b2e0499"
,
        "assets": [
            {
                "netbiosName": "",
                "assetId": "a8999684-49c1-4492-87a9-8a9f77a1ef84",
                "name": "www.qualys.com",
                "operatingSystem": "NetScaler",
                "tags": [
                    {
                        "name": "Internet Facing Assets",
                        "uuid": "49af0a63-f5f2-4b2c-b942-
af22afd91243"
                    }
                ],
                "hostInstances": [
                    {
                        "protocol": "tcp",
                        "sslProtocols": [
                            "TLSv1.2"
                        ],
                        "port": 443,
                        "grade": "A",
                        "service": "http",
                        "fqdn": "",
                        "vulnerabilities": [
                            {
                                "title": "SSL Certificate -
Information",
                                "severity": 1,
                                "qid": 86002
                            },
                            {
                                "title": "SSL/TLS Protocol
Properties",
                                "severity": 1,
                                "qid": 38706
                            },
                            {
                                "title": "SSL Server Information
Retrieval",
                                "severity": 1,
                                "qid": 38116
                            },
```

```
                            {
                               "title": "SSL/TLS invalid protocol
version tolerance",
                                "severity": 1,
                                "qid": 38597
                            },
                            {
                                "title": "HTTP Strict Transport
Security (HSTS) Support Detected",
                                "severity": 1,
                                "qid": 86137
                            },
                            {
                               "title": "TLS Secure Renegotiation
Extension Support Information",
                                "severity": 1,
                                "qid": 42350
                            }
                        ],
                        "vulnCount": 6
                    }
                ],
                "created": 1568753271000,
                "updated": 1600191908000,
                "assetInterfaces": [
                    {
                        "hostname": "www.qualys.com",
                        "address": "64.39.96.133"
                    }
                ],
                "certificateCount": 0
            }
        ],
        "selfSigned": false,
        "validTo": 1628078400000,
        "issuerCategory": "unapproved",
        "subjectAlternativeNames": {
            "IP Address": null,
            "DNS Name": [
                "qualys.com",
                "www.qualys.com"
            ]
        },
        "lastFound": 1600191908000,
        "extendedValidation": true,
        "orderStatus": ""
```

```
}
```

## Sample with certificate Details parameter set to Basic

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{ \"filter\" : \"subject.name:www.qualys.com\", \"pageNumber\":
0, \"pageSize\" : 1, \"certificateDetails\": \"basic\"}" -H
"Authorization: Bearer <JWT Token>"
```

Response:


```
[{
        "keySize": 2048,
        "subject": {
            "organization": "Qualys, Inc.",
            "locality": "Foster City",
            "name": "www.qualys.com",
            "state": "California",
            "country": "US",
            "organizationUnit": []
        },
        "validFrom": 1596067200000,
        "signatureAlgorithm": "SHA256withRSA",
        "issuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
            "certhash":
"403e062a2653059113285baf80a0d4ae422c848c9f78fad01fc94bc5b87fef1a"
,
            "locality": ""
        },
        "rootissuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
```

```
            "certhash":
"7431e5f4c3c1ce4690774f0b61e05440883ba9a01ed00ba6abd7806ed3b118cf"
,
            "locality": ""
        },
        "instanceCount": 1,
        "dn": "CN=www.qualys.com, O=\"Qualys, Inc.\", L=Foster
City, ST=California, C=US, serialNumber=3152140,
StateOrProvince=Delaware, CountryName=US, businessCategory=Private
Organization",
        "certhash":
"61ffdf5ec74189b0f6f256fc42fe858bb04c1862c0f4cb9ec9d5f9bf4b2e0499"
,
        "assets": [
            {
                "netbiosName": "",
                "assetId": "a8999684-49c1-4492-87a9-8a9f77a1ef84",
                "name": "www.qualys.com",
                "operatingSystem": "NetScaler",
                "tags": [
                    {
                        "name": "Internet Facing Assets",
                        "uuid": "49af0a63-f5f2-4b2c-b942-
af22afd91243"
                    }
                ],
                "hostInstances": [
                    {
                        "protocol": "tcp",
                        "sslProtocols": [
                            "TLSv1.2"
                        ],
                        "port": 443,
                        "grade": "A",
                        "service": "http",
                        "fqdn": "",
                        "vulnerabilities": [
                            {
                                "title": "SSL Certificate -
Information",
                                "severity": 1,
                                "qid": 86002
                            },
                            {
                                "title": "SSL/TLS Protocol
Properties",
```

```
                                "severity": 1,
                                "qid": 38706
                            },
                            {
                                "title": "SSL Server Information
Retrieval",
                                "severity": 1,
                                "qid": 38116
                            },
                            {
                               "title": "SSL/TLS invalid protocol
version tolerance",
                                "severity": 1,
                                "qid": 38597
                            },
                            {
                                "title": "HTTP Strict Transport
Security (HSTS) Support Detected",
                                "severity": 1,
                                "qid": 86137
                            },
                            {
                               "title": "TLS Secure Renegotiation
Extension Support Information",
                                "severity": 1,
                                "qid": 42350
                            }
                        ],
                        "vulnCount": 6
                    }
                ],
                "created": 1568753271000,
                "updated": 1600191908000,
                "assetInterfaces": [
                    {
                        "hostname": "www.qualys.com",
                        "address": "64.39.96.133"
                    }
                ],
                "certificateCount": 0
            }
        ],
        "selfSigned": false,
        "validTo": 1628078400000,
        "issuerCategory": "unapproved",
        "subjectAlternativeNames": {
```

```
                "IP Address": null,
                "DNS Name": [
                    "qualys.com",
                    "www.qualys.com"
                ]
            },
            "lastFound": 1600191908000,
            "extendedValidation": true,
            "orderStatus": ""
        }
    ]
```


## Sample with certificate Details parameter set to Extended

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{ \"filter\" : \"subject.name:www.qualys.com\", \"pageNumber\":
0, \"pageSize\" : 1, \"certificateDetails\": \"extended\"}" -H
"Authorization: Bearer <JWT Token>"
```

Response:


```
[{
        "keySize": 2048,
        "subject": {
            "organization": "Qualys, Inc.",
            "locality": "Foster City",
            "name": "www.qualys.com",
            "state": "California",
            "country": "US",
            "organizationUnit": []
        },
        "validFrom": 1596067200000,
        "signatureAlgorithm": "SHA256withRSA",
        "issuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
            "certhash":
"403e062a2653059113285baf80a0d4ae422c848c9f78fad01fc94bc5b87fef1a"
,
```

```
            "locality": ""
        },
        "rootissuer": {
            "organization": "DigiCert Inc",
            "organizationUnit": [
                "www.digicert.com"
            ],
            "name": "DigiCert SHA2 Extended Validation Server CA",
            "country": "US",
            "state": "",
            "certhash":
"7431e5f4c3c1ce4690774f0b61e05440883ba9a01ed00ba6abd7806ed3b118cf"
,
            "locality": ""
        },
        "instanceCount": 1,
        "dn": "CN=www.qualys.com, O=\"Qualys, Inc.\", L=Foster
City, ST=California, C=US, serialNumber=3152140,
StateOrProvince=Delaware, CountryName=US, businessCategory=Private
Organization",
        "certhash":
"61ffdf5ec74189b0f6f256fc42fe858bb04c1862c0f4cb9ec9d5f9bf4b2e0499"
,
        "assets": [
            {
                "netbiosName": "",
                "assetId": "a8999684-49c1-4492-87a9-8a9f77a1ef84",
                "name": "www.qualys.com",
                "operatingSystem": "NetScaler",
                "tags": [
                    {
                        "name": "Internet Facing Assets",
                        "uuid": "49af0a63-f5f2-4b2c-b942-
af22afd91243"
                    }
                ],
                "hostInstances": [
                    {
                        "protocol": "tcp",
                        "sslProtocols": [
                            "TLSv1.2"
                        ],
                        "port": 443,
                        "grade": "A",
                        "service": "http",
                        "fqdn": "",
```

```
                        "vulnerabilities": [
                            {
                                "title": "SSL Certificate -
Information",
                                "severity": 1,
                                "qid": 86002
                            },
                            {
                                "title": "SSL/TLS Protocol
Properties",
                                "severity": 1,
                                "qid": 38706
                            },
                            {
                                "title": "SSL Server Information
Retrieval",
                                "severity": 1,
                                "qid": 38116
                            },
                            {
                               "title": "SSL/TLS invalid protocol
version tolerance",
                                "severity": 1,
                                "qid": 38597
                            },
                            {
                                "title": "HTTP Strict Transport
Security (HSTS) Support Detected",
                                "severity": 1,
                                "qid": 86137
                            },
                            {
                               "title": "TLS Secure Renegotiation
Extension Support Information",
                                "severity": 1,
                                "qid": 42350
                            }
                        ],
                        "vulnCount": 6
                    }
                ],
                "created": 1568753271000,
                "updated": 1600191908000,
                "assetInterfaces": [
                    {
                        "hostname": "www.qualys.com",
```

```
                        "address": "64.39.96.133"
                    }
                ],
                "certificateCount": 0
            }
        ],
        "selfSigned": false,
        "validTo": 1628078400000,
        "issuerCategory": "unapproved",
        "serialNumber": "0e66f3475fd186c97dbd7fc274b0ddca",
        "subjectAlternativeNames": {
            "DNS Name": [
                "qualys.com",
                "www.qualys.com"
            ],
            "IP Address": null
        },
        "lastFound": 1600191908000,
        "extendedValidation": true,
        "orderStatus": "",
        "keyUsage": [
            "Digital signature",
            "Key encipherment"
        ],
        "rawData": "-----BEGIN CERTIFICATE-----
\nMIIGyjCCBbKgAwIBAgIQDmbzR1/Rhsl9vX/CdLDdyjANBgkqhkiG9w0BAQsFADB1
\nMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
\nd3cuZGlnaWNlcnQuY29tMTQwMgYDVQQDEytEaWdpQ2VydCBTSEEyIEV4dGVuZGVk
\nIFZhbGlkYXRpb24gU2VydmVyIENBMB4XDTIwMDczMDAwMDAwMFoXDTIxMDgwNDEy
\nMDAwMFowgckxHTAbBgNVBA8MFFByaXZhdGUgT3JnYW5pemF0aW9uMRMwEQYLKwYB
\nBAGCNzwCAQMTAlVTMRkwFwYLKwYBBAGCNzwCAQITCERlbGF3YXJlMRAwDgYDVQQF
\nEwczMTUyMTQwMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIG
\nA1UEBxMLRm9zdGVyIENpdHkxFTATBgNVBAoTDFF1YWx5cywgSW5jLjEXMBUGA1UE
\nAxMOd3d3LnF1YWx5cy5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
\nAQDOXOJ/OXDMh0rWLUE5zE13422k8QFDUTP07LrrThoAED/7Q8VAwIYlTK5I4pVl
\n8F7xA0l/PUA1fNYGru3b8IEfDPV8D3zvTg14gTbtQAx8UDXbnJISVBI6H40+F3fT
\niP7H+X9VV/LBcUVeETiiGTQsA5Mzqn5AxVELrqUrbN2cFRa8RrNS4ho2w5XFNV+D
\n5ae+H0ztoyfNmpuDcV4fGD6e/b9ImkSwW+Q2I0Vm8pXqTYZ3Cfp0/eFtnD2LWYKq
\npL42ejF2r0H9EpqYdQ2OQ3xG4GMYEFZmq744q7cQ8MGQQZ1mQTEpIDbfy4lou/9p
\nXnBkRS9x6VBFDTuAVfmY644zAgMBAAGjggL/MIIC+zAfBgNVHSMEGDAWgBQ901Cl
\n1qCt7vNKYApl0yHU+PjWDzAdBgNVHQ4EFgQUSpkxgc8NwL4vkE3yAtWk7obFjtww
\nJQYDVR0RBB4wHIIKcXVhbHlzLmNvbYIOd3d3LnF1YWx5cy5jb20wDgYDVR0PAQH/
\nBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB1BgNVHR8EbjBs
\nMDSgMqAwhi5odHRwOi8vY3JsMy5kaWdpY2VydC5jb20vc2hhMi1ldi1zZXJ2ZXIt
\nZzIuY3JsMDSgMqAwhi5odHRwOi8vY3JsNC5kaWdpY2VydC5jb20vc2hhMi1ldi1z
\nZXJ2ZXItZzIuY3JsMEsGA1UdIAREMEIwNwYJYIZIAYb9bAIBMCowKAYIKwYBBQUH
\nAgEWHGh0dHBzOi8vd3d3LmRpZ2ljZXJ0LmNvbS9DUFMwBwYFZ4EMAQEwgYgGCCsG
```

```
\nAQUFBwEBBHwwejAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AuZGlnaWNlcnQuY29t
\nMFIGCCsGAQUFBzAChkZodHRwOi8vY2FjZXJ0cy5kaWdpY2VydC5jb20vRGlnaUNl
\ncnRTSEEyRXh0ZW5kZWRWYWxpZGF0aW9uU2VydmVyQ0EuY3J0MAwGA1UdEwEB/wQC
\nMAAwggEEBgorBgEEAdZ5AgQCBIH1BIHyAPAAdgD2XJQv0XcwIhRUGAgwlFaO400T
\nGTO/3wwvIAvMTvFk4wAAAXOg8dw9AAAEAwBHMEUCIB9GUjDgcZrgO42Z97jUglBk
\nx0PwlZvhBADnnPEthAqiAiEA1XkIrIz8Cri9JqJws5OMBcLu3MxSepQz3183kiDZ
\nGuAAdgBc3EOS/uarRUSxXprUVuYQN/vV+kfcoXOUsl7m9scOygAAAXOg8dxpAAAE
\nAwBHMEUCIAk0qPCGzpKhgO4JXpJG4HGNrWJwxfuFLR8MEcVsQVVIAiEAqSmtKbR8
\nVnC/SdPhqjvvYsh4hf9/LvYh6EwVrYiM7b0wDQYJKoZIhvcNAQELBQADggEBALEX
\nR1BiVmMmHWiQvlY9wKzmab5y4dg6+QtylR9ycPvItgz8QfOw45xBT6celK0d7Qmy
\ndDG+EbhojYjT382zRjwWHizDmr2BKtURojc2zWIwwNpSbtLtBnSwRUJQ7Y+q70mM
\nkKZ8xzzjtVS82ayvVYLUkSJm+zPcj7w22IOOryMCzv1QDhQmo0kNJPghQNxqEKaq
\nuk/XqX5LTj0p8Z9V6YTb6uHVbucgJaDwxjIBrVGq8S1cIvKd++c2QslZUS5tWyJz
\nijn1RRJ0twr/Go3H5sDzNVXN1Sf04+DnjFKdIAKwUvKNLLVmcE+ws2iQpcL2Ita6
\nDN+frzrJn9/MAz4tlG0=\n-----END CERTIFICATE-----",
        "enhancedKeyUsage": [
            "1.3.6.1.5.5.7.3.1",
            "1.3.6.1.5.5.7.3.2"
        ],
        "subjectKeyIdentifier":
"4a993181cf0dc0be2f904df202d5a4ee86c58edc",
        "authKeyIdentifier":
"3dd350a5d6a0adeef34a600a65d321d4f8f8d60f"
    }
  ]
```

# List Assets for a Certificate

**/certview/v1/certificates/{certhash}/assets**

[POST]

Use this API function to retrieve a list of the assets of the specific certificate. Use cert hash as an input query. The response contains all the asset details of the specific certificate. View details of assets like tags and instances. Use this API to get all the assets when the certificate has more than 1000 assets. If the total count is greater than the number of results returned, you can fetch the next page by calling the API again with the next page number.

## Input Parameters

| | |
|---|---|
| certhash (string) | Query parameter- Provide hash of the certificate. |
| attributes | (Optional) Provide specific attributes to display additional asset details based on<br>-**tags**: View certificate list of assets with specified tags.<br>-**hostInstances**: View the certificate list which contains list of instances on which this certificate was found.<br>-**tags and hostInstances**: View the certificate list which contains all the primitive details of the assets along with the asset tags and instances on which this certificate was found. |
| filter (String) | (Optional) Filter the events list by providing a query using Qualys syntax. Refer to the How to Search topic in the online help for assistance with creating your query.<br>For example - expiryGroup: Expired<br>Refer to the list of tokens you can use to build the query: Search tokens |
| pageNumber (Integer) | (Optional) The page to be returned. Starts from zero. |
| pageSize (Integer) | (Optional) Provide the number of records per page to be included in the response.<br>Default: 10.<br>Maximum: 200<br>For example, the total result set is 8000 assets. If the page size is specified as 80, then the result is divided in 100 pages with 80 assets each. |
| Note: Use combination of pageNumber and pageSize parameters to ensure that the results returned are less than 10000 records. If it exceeds 10000 records then an error message is displayed. | |
| sort (String) | (Optional) Sort the results ascending or descending order.<br>By default the result is sorted by {updated: desc} |

**Note**: This API supports only new query format. For more details refer to Query Example.

## Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

## Query Example

| Example | If you want to view the certificate with asset name server1. |
|---|---|
| Old Format | asset.name:server1 |
| New Format | asset:(name:server1) |

## Sample with all parameters defined

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/<certhash>/
assets" --header "Accept: application/json" -H "Authorization: Bearer <JWT
Token>" -d "{\"certificateDetails\":\"basic\"
```

Response:


```
[
   {
      "netbiosName":"",
      "assetId":"8d6d19b5-9201-445b-87c2-b61aeb3f4fa5",
      "name":"ABC.COM",
      "operatingSystem":"NetScaler",
      "created":1587464966000,
      "updated":1625213136000,
      "assetInterfaces":[
         {
            "hostname":"ABC.COM",
            "address":"10.XXX.X.XX"
         }
      ],
      "certificateCount":0
   },
   {
      "netbiosName":"",
      "assetId":"2a9428e4-9130-4979-9f8c-dcfe86579c39",
      "name":"Server1",
      "operatingSystem":"NetScaler",
      "created":1591703972000,
      "updated":1624014415000,
      "assetInterfaces":[
         {
            "hostname":"ABC.COM",
            "address":"10.XXX.X.XX"
         }
      ],
      "certificateCount":0
   }
]
```

# Chapter 3 - Analyze Certificate API

Use these API functions to analyze information based on host or IP.

Use this API to retrieve the list of endpoints that are associated with an FQDN in the CertView inventory

## Analyze Certificate Information

**/certview/v1/analyze**

[POST]

### Input Parameters

| | |
|---|---|
| host (String) | (Required) Host on which scan is executed, it can be IP<br>For example - www.ssllabs.com, 10.10.10.10 |
| Authorization (String) | (Required) Authorization token to authenticate to the Qualys Cloud Platform.<br>Prepend token with "Bearer" and one space. For example - Bearer authToken |

### Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

### Sample - Host is IP

Request:

```
curl -X POST "gateway.qg1.apps.qualys.com/certview/v1/analyze" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{\"host\":\"XXX.XXX.XXX.XXX\"}" -H "Authorization: Bearer <JWT
Token>"
```

Response:

```
{
    "host": "XXX.XXX.XXX.XXX",
    "endpoints": [
        {
            "ipAddress": "XXX.XXX.XXX.XXX",
            "port": 443,
            "service": "http",
            "serverName": "",
            "grade": "A",
            "gradeTrustIgnored": "A",
            "hasWarnings": false,
            "exceptional": false
```

```
            }
        ]
    }
```

## Sample - Host is FQDN

Request:

```
curl -X POST "gateway.qg1.apps.qualys.com/certview/v1/analyze" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{\"host\":\"www.qualys.com\"}" -H "Authorization: Bearer <JWT
Token>"
```

Response:

```
{
    "host": "www.qualys.com",
    "endpoints": [
        {
            "ipAddress": "XXX.XXX.XXX.XXX",
            "port": 443,
            "service": "http",
            "serverName": "",
            "grade": "A",
            "gradeTrustIgnored": "A",
            "hasWarnings": false,
            "exceptional": false
        }
    ]
}
```

# Chapter 4 - Endpoint API

Use these API function to retrieve detailed endpoint information.

## List Endpoints

**/certview/v1/getEndpointData**

[POST]

### Input Parameters

| | |
|---|---|
| Authorization (String) | (Required) Authorization token to authenticate to the Qualys Cloud Platform.<br>Prepend token with "Bearer" and one space. For example - Bearer authToken |
| ip (String) | (Required) Host IP for which the endpoint details are required. |
| port (Integer) | Used to filter the endpoint details based on port.<br><br>In Certview scan, we can scan multiple ports as certificates can be found on multiple ports. Define the port number to filter the endpoint data based on port. |
| fqdn (String) | Used to filter the endpoint details based on FQDN.<br><br>Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the service or protocol parameter is specified. |
| service (String) | Used to filter the endpoint details based on service.<br><br>Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the fqdn or protocol parameter is specified. |
| protocol (String) | Used to filter the endpoint details based on protocol.<br><br>Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the service or protocol parameter is specified. |

### Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

## Sample with all parameters defined

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/getEndpointData"
-H "Accept: application/json" -H "Content-Type: application/json"
-d "{ \"ip\": \"XXX.XXX.XXX.XXX\", \"port\": 443, \"fqdn\": \"\",
\"service\": \"tcp\"}" -H "Authorization: Bearer <JWT Token>"
```

Response:


```
[
    {
        "ipAddress": "XXX.XXX.XXX.XXX",
        "port": 443,
        "lastFound": 1600191908000,
        "service": "http",
        "grade": "A",
        "gradeTrustIgnored": "A",
        "hasWarnings": false,
        "isExceptional": false,
        "details": {
            "certChains": [
                {
                    "certIds": [

"61ffdf5ec74189b0f6f256fc42fe858bb04c1862c0f4cb9ec9d5f9bf4b2e0499"
,

"403e062a2653059113285baf80a0d4ae422c848c9f78fad01fc94bc5b87fef1a"
,

"7431e5f4c3c1ce4690774f0b61e05440883ba9a01ed00ba6abd7806ed3b118cf"
                    ],
                    "trustPaths": [
                        {
                            "certIds": [

"61ffdf5ec74189b0f6f256fc42fe858bb04c1862c0f4cb9ec9d5f9bf4b2e0499"
,

"403e062a2653059113285baf80a0d4ae422c848c9f78fad01fc94bc5b87fef1a"
,

"7431e5f4c3c1ce4690774f0b61e05440883ba9a01ed00ba6abd7806ed3b118cf"
                            ],
                            "trust": [
```

```
                    {
                        "rootStore": "Mozilla",
                        "isTrusted": true
                    }
                ]
            }
        ],
        "noSni": false
    }
],
"protocols": [
    {
        "id": 771,
        "name": "TLS",
        "version": "1.2"
    }
],
"suites": [
    {
        "protocol": 771,
        "list": [
            {
                "id": 103,
                "name": "DHE-RSA-AES128-SHA256",
                "cipherStrength": 128,
                "kxType": "DH"
            },
            {
                "id": 107,
                "name": "DHE-RSA-AES256-SHA256",
                "cipherStrength": 256,
                "kxType": "DH"
            },
            {
                "id": 158,
                "name": "DHE-RSA-AES128-GCM-SHA256",
                "cipherStrength": 128,
                "kxType": "DH"
            },
            {
                "id": 159,
                "name": "DHE-RSA-AES256-GCM-SHA384",
                "cipherStrength": 256,
                "kxType": "DH"
            },
```

```
                {
                    "id": 49171,
                    "name": "ECDHE-RSA-AES128-SHA",
                    "cipherStrength": 128,
                    "kxType": "ECDH"
                },
                {
                    "id": 49172,
                    "name": "ECDHE-RSA-AES256-SHA",
                    "cipherStrength": 256,
                    "kxType": "ECDH"
                },
                {
                    "id": 49191,
                    "name": "ECDHE-RSA-AES128-SHA256",
                    "cipherStrength": 128,
                    "kxType": "ECDH"
                },
                {
                    "id": 49192,
                    "name": "ECDHE-RSA-AES256-SHA384",
                    "cipherStrength": 256,
                    "kxType": "ECDH"
                },
                {
                    "id": 49199,
                    "name": "ECDHE-RSA-AES128-GCM-SHA256",
                    "cipherStrength": 128,
                    "kxType": "ECDH"
                },
                {
                    "id": 49200,
                    "name": "ECDHE-RSA-AES256-GCM-SHA384",
                    "cipherStrength": 256,
                    "kxType": "ECDH"
                }
            ]
        }
    ],
    "vulnBeast": false,
    "renegSupport": 2,
    "compressionMethods": 0,
    "supportsRc4": false,
    "rc4WithModern": false,
    "rc4Only": false,
```

```
            "forwardSecrecy": 4,
            "supportsAead": true,
            "protocolIntolerance": 48,
            "heartbleed": false,
            "heartbeat": false,
            "openSslCcs": 1,
            "openSSLLuckyMinus20": 1,
            "ticketbleed": 1,
            "bleichenbacher": 1,
            "poodle": false,
            "poodleTls": 1,
            "fallbackScsv": false,
            "freak": false,
            "hasSct": 1,
            "logjam": false,
            "drownVulnerable": false,
            "zombiePoodle": 1,
            "goldenDoodle": 1,
            "supportsCBC": true,
            "zeroLengthPaddingOracle": 1,
            "sleepingPoodle": 1
        }
    }
]
```

# Chapter 5 - Enroll and Renew Certificate APIs

Use these API functions to enroll new or renew existing certificates using the new APIs. The following APIs enable you to complete the end-to-end enrollment or renewal workflow:

- Create Enrollment/Renewal Certificate Request
- Update Certificate Request
- Update Status of Certificate Request
- View Certificate Request
- List DigiCert Organizations
- List DigiCert Products
- List Digicert EV Approvers

## Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

# Create Enrollment/Renewal Certificate Request

Use this API to enroll or renew certificates

| | |
|---|---|
| APIs affected | /certview/rest/public/v1/certificates/enrollment/digicert/orders |
| Method | POST |
| New or Updated APIs | New |

## Input Parameters

Input parameters for Create and Update APIs

| | |
|---|---|
| approverUserNames (array) | (Required) Array of valid user names. User roles must be manager, PKI admin or approvers. |
| intermediateCA (object) | (Required) Intermediate Certificate Authority information. Make sure:<br>- At least one of the params certhash, commonName or serialNumber is required.<br>- CA is DigiCert's public intermediate CA<br>- CA is configured with CA API key |
| certhash (string) | (Optional) Provide hash of the certificate. |
| commonName (string) | (Optional) Fully qualified domain name of the Web server that will receive the certificate |
| serialNumber (string) | (Optional) A short, unique identifier for each certificate generated by the certificate issuer |
| certificate (object) | (Required) |
| commonName (string) | (Required) Provide a wildcard character if the product name is of wildcard type. |
| csr (object) | (Required) Certificate Signing Request Information |
| autoGenerateCSR (boolean) | (Optional) If this flag is set to True then Qualys will generate csr value and return private key info in the response of the API. If this field is set to True then encoded_csr field can not be set By default the value is set to False. |
| encodedCSR (string) | (Optional) A valid Encoded Certificate Signing Request |
| organizationUnits (array) | (Optional) provide value for the OU field for the certificate. |
| signatureHash (string) | Required) Certificate's signing algorithm hash. Accepted values: SHA-256, SHA-384, SHA-512 |
| renewal (object) | (Optional) Required for certificate renewal request. |
| digicertPreviousOrderId (integer) | (Required) If the request is a renewal of a previous request then add the previous request id. |

| | |
|---|---|
| renewalOfCertificate (string) | (Required) Provide certhash of the old certificate for which this renewal request is required.<br>Make sure:<br>- certificate is in customer's account<br>- certificate is leaf certificate<br>- certificate is not in IN_RENEWAL status |
| validity (object) | (Required) Provide any one of the following values: customExpirationDate, validityYears and validityDays params<br>Make sure only one value is provided in a request. |
| customExpirationDate (date) | (Optional) Expiry date of the certificate. |
| validityYears (integer) | (Optional) Number of years that the certificate is valid. |
| validityDays (integer) | (Optional) Number of days that the certificate is valid. |
| digicertOrganizationId (integer) | (Required) Get organization id using List DigiCert Organizations API |
| digicertProductNameId (integer) | (Required) Get product name id using List DigiCert Products API |
| digicertEVApproverUserIds (array) | (Optional) Required when product name is of EV type. Get EV Approvers user id using List Digicert EV Approvers API |
| comment (string) | (Optional) Any additional comments. |

### Sample to Submit Certificate Enrollment Request

API request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/orders" -H "Accept: application/json" -H "Content-Type:
application/json" -d '{ "approverUserNames": [ "quays_sd" ],
"certificate": { "commonName": "ABC.com", "csr": {
"autoGenerateCSR": true }, "organizationUnits": [ "QA" ],
"signatureHash": "SHA-256" }, "comment": "api test",
"digicertOrganizationId": 525858, "digicertProductNameId":
"private_ssl_plus", "intermediateCA": { "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
}, "validity": { "validityYears": 1 } }' -H "Authorization: Bearer
<jwt token>"
```

Response:

```
{
    "uuid": "cb95d100-ec30-11ea-920d-eb66140967e3",
    "intermediateCA": {
        "name": "DigiCert Test SHA2 Intermediate CA-1",
        "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
    },
```

```
    "approverUserNames": [
        "quays_sd"
    ],
    "requesterUserName": "quays_sd",
    "certificate": {
        "commonName": "ABC.com",
        "organizationUnits": [
            "QA"
        ],
        "signatureHash": "SHA-256",
        "encodedCSR": "<csr>",
        "privateKey": "<private key>",
        "dnsNames": null
    },
    "renewal": null,
    "validity": {
        "validityYears": 1,
        "validityDays": null,
        "customExpirationDate": null
    },
    "digicertOrganization": {
        "id": 525858
    },
    "digicertProductNameId": "private_ssl_plus",
    "digicertEVApproverUserIds": null,
    "status": "SUBMITTED",
    "caStatus": "",
    "created": "2020-09-01T08:54:50.473+0000",
    "updated": "2020-09-01T08:54:50.473+0000"
}
```

# Update Certificate Request

Use this API to edit an existing enrollment/renewal request

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/orders/{certificate_order_uuid} |
| Method | PUT |
| New or Updated APIs | New |

Refer to Input Parameters

### Sample to Update Certificate Request

API request:

```
curl -X PUT
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/orders/cb95d100-ec30-11ea-920d-eb66140967e3" -H "Accept:
application/json" -H "Content-Type: application/json" -d '{
"approverUserNames": [ "quays_sd" ], "certificate": {
"commonName": "ABC.com", "csr": { "autoGenerateCSR": true },
"organizationUnits": [ "QA" ], "signatureHash": "SHA-256" },
"comment": "Updated api test comment", "digicertOrganizationId":
525858, "digicertProductNameId": "private_ssl_plus",
"intermediateCA": { "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
}, "validity": { "validityYears": 1 } }' -H "Authorization: Bearer
<jwt token>"
```

Response:

```
{
    "uuid": "cb95d100-ec30-11ea-920d-eb66140967e3",
    "intermediateCA": {
        "name": "DigiCert Test SHA2 Intermediate CA-1",
        "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
    },
    "approverUserNames": [
        "quays_sd"
    ],
    "requesterUserName": "quays_sd",
    "certificate": {
        "commonName": "ABC.com",
        "organizationUnits": [
            "QA"
        ],
        "signatureHash": "SHA-256",
        "encodedCSR": "<csr>",
```

```
        "privateKey": "<private key>",
        "dnsNames": null
    },
    "renewal": null,
    "validity": {
        "validityYears": 1,
        "validityDays": null,
        "customExpirationDate": null
    },
    "digicertOrganization": {
        "id": 525858
    },
    "digicertProductNameId": "private_ssl_plus",
    "digicertEVApproverUserIds": null,
    "status": "SUBMITTED",
    "caStatus": "",
    "created": "2020-09-01T08:54:50.473+0000",
    "updated": "2020-09-01T08:58:58.138+0000"
}
```

# Update Status of Certificate Request

Use this API to approve, reject, or cancel an existing enrollment/renewal request

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/orders/{certificate_order_uuid}/status |
| Method | PUT |
| New or Updated APIs | New |

## Input Parameters

Input parameters for Status update API

| | |
|---|---|
| status (string) | (Required) Provide one of the following: APPROVED, CANCELLED, REJECTED<br>Make sure:<br>- Only one of approvers, pki or manager can approve, reject or cancel<br>- Once approved request can not rejected or canceled |
| comment (string) | (Required) Comments about status change. |

### Sample to Update Status of Certificate Request

API request:

```
curl -X PUT
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/orders/cb95d100-ec30-11ea-920d-eb66140967e3/status" -H
"Accept: application/json" -H "Content-Type: application/json" -d
'{
"comment": "API request Cancelled",
"status": "CANCELLED"
}' -H "Authorization: Bearer <jwt token>"
```

Response:

```
No Content
Response Code: 204
```

# View Certificate Request

Use this API to get details for specified request

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/orders/{certificate_order_uuid} |
| Method | GET |
| New or Updated APIs | New |

## Input Parameters

Input parameters for View certificate request API

| | |
|---|---|
| uuid (string) | (Required) UUID of the certificate |
| approverUserNames (array) | (Required) Array of valid user names. User roles must be manager, PKI admin or approvers. |
| requesterUserName (string) | Requester user name. |
| intermediateCA (object) | (Required) Intermediate Certificate Authority information. Make sure:<br>- At least one of the params certhash, commonName or serialNumber is required.<br>- CA is DigiCert's public intermediate CA<br>- CA is configured with CA API key |
| certhash (string) | (Optional) Provide hash of the certificate. |
| commonName (string) | (Optional) Fully qualified domain name of the Web server that will receive the certificate |
| serialNumber (string) | (Optional) A short, unique identifier for each certificate generated by the certificate issuer |
| certificate (object) | (Required) |
| commonName (string) | (Required) Provide a wildcard character if the product name is of wildcard type. |
| csr (object) | (Required) Certificate Signing Request Information |
| autoGenerateCSR (boolean) | (Optional) If this flag is set to True then Qualys will generate csr value and return private key info in the response of the API. If this field is set to True then encoded_csr field can not be set By default the value is set to False. |
| encodedCSR (string) | (Optional) A valid Encoded Certificate Signing Request |
| organizationUnits (array) | (Optional) provide value for the OU field for the certificate. |
| signatureHash (string) | Required) Certificate's signing algorithm hash. Accepted values: SHA-256, SHA-384, SHA-512 |
| renewal (object) | (Optional) Required for certificate renewal request. |
| digicertPreviousOrderId (integer) | (Required) If the request is a renewal of a previous request then add the previous request id. |

| | |
|---|---|
| renewalOfCertificate (string) | (Required) Provide certhash of the old certificate for which this renewal request is required.<br>Make sure:<br>- certificate is in customer's account<br>- certificate is leaf certificate<br>- certificate is not in IN_RENEWAL status |
| validity (object) | (Required) Provide any one of the following values: customExpirationDate, validityYears and validityDays params<br>Make sure only one value is provided in a request. |
| customExpirationDate (date) | (Optional) Expiry date of the certificate |
| validityYears (integer) | (Optional) Number of years that the certificate is valid |
| validityDays (integer) | (Optional) Number of days that the certificate is valid |
| digicertOrganizationId (integer) | (Required) Get organization id using List DigiCert Organizations API |
| digicertProductNameId (integer) | (Required) Get product name id using List DigiCert Products API |
| digicertEVApproverUserIds (array) | (Optional) Required when product name is of EV type. Get EV Approvers user id using List Digicert EV Approvers API |
| status (string) | (Optional) Provide any of the following values: CANCELLED, APPROVED, SUBMITTED, ISSUED |
| caStatus (string) | (Optional) Status from the Certificate Authority |
| created (date) | (Optional) Date the request was created |
| updated (date) | (Optionsl) Date the request was updated |

### Sample to View Certificate Request

API request:

```
curl -X GET
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/orders/cb95d100-ec30-11ea-920d-eb66140967e3" -H "Accept:
application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer <jwt token>"
```

Response:

```
{
    "uuid": "cb95d100-ec30-11ea-920d-eb66140967e3",
    "intermediateCA": {
        "name": "DigiCert Test SHA2 Intermediate CA-1",
        "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
    },
    "approverUserNames": [
        "quays_sd"
    ],
    "requesterUserName": "quays_sd",
```

```
    "certificate": {
        "commonName": "ABC.com",
        "organizationUnits": [
            "QA"
        ],
        "signatureHash": "SHA-256",
        "encodedCSR": "<csr>",
        "privateKey": null,
        "dnsNames": null
    },
    "renewal": null,
    "validity": {
        "validityYears": 1,
        "validityDays": null,
        "customExpirationDate": null
    },
    "digicertOrganization": {
        "id": 525858
    },
    "digicertProductNameId": "private_ssl_plus",
    "digicertEVApproverUserIds": null,
    "status": "SUBMITTED",
    "caStatus": "",
    "created": "2020-09-01T08:54:50.473+0000",
    "updated": "2020-09-01T08:58:58.138+0000"
}
```

# List DigiCert Organizations

Use this API to list Organizations registered with DigiCert

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/organizations |
| Method | POST |
| New or Updated APIs | New |

## Input Parameters

Input parameters for DigiCert APIs. It is required to provide at least one of the params certhash, commonName or serialNumber.

| | |
|---|---|
| certhash (string) | (Optional) Secure hash of the certificate |
| commonName (string) | (Optional) Fully qualified domain name of the Web server that will receive the certificate |
| serialNumber (string) | (Optional) A short, unique identifier for each certificate generated by the certificate issuer |

### Sample to List DigiCert Organizations

API request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/organizations" -H "Accept: application/json" -H "Content-
Type: application/json" -d '{
"certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
}' -H "Authorization: Bearer <jwt token>"
```

Response:

```
{
    "organizations": [
        {
            "id": 525858,
            "status": "active",
            "name": "Qualys, Inc",
            "assumedName": null,
            "displayName": "Qualys, Inc",
            "active": true
        }
    ]
}
```

# List DigiCert Products

Use this API to list DigiCert products for your account

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/products |
| Method | POST |
| New or Updated APIs | New |

Refer to Input Parameters

### Sample to List DigiCert Products

API request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/products" -H "Accept: application/json" -H "Content-Type:
application/json" -d '{
"certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
}' -H "Authorization: Bearer <jwt token>"
```

Response:

```
{
    "products": [
        {
            "groupName": "securesite_ssl_certificate",
            "nameId": "ssl_ev_securesite",
            "name": "Secure Site EV SSL",
            "type": "ssl_certificate",
            "sslCertificateType": null
        },
        {
            "groupName": "securesite_ssl_certificate",
            "nameId": "ssl_ev_securesite_multi_domain",
            "name": "Secure Site EV Multi-Domain SSL",
            "type": "ssl_certificate",
            "sslCertificateType": null
        }
    ]
}
```

# List Digicert EV Approvers

Use this API to list EVInput Parameters approvers registered with DigiCert

| | |
|---|---|
| APIs affected | certview/rest/public/v1/certificates/enrollment/digicert/evApprovers |
| Method | POST |
| New or Updated APIs | New |

Refer to Input Parameters

### Sample to List DigiCert Products

API request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates/digi
cert/evApprovers" -H "Accept: application/json" -H "Content-Type:
application/json" -d '{
"certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
}' -H "Authorization: Bearer <jwt token>"
```

Response:

```
{
    "evApprovers": [
        {
            "userId": "1541521",
            "name": "John White",
            "firstName": "John",
            "lastName": "White"
        },
        {
            "userId": "1551253",
            "name": "Kelly Smith",
            "firstName": "Kelly",
            "lastName": "Smith"
        }
    ]
}
```

# Appendix A - Error codes/Descriptions

This appendix lists the Certificate View API error codes along with a description of what each code means. For an API request that had an error, you'll find the error code and text in the XML response.

| HTTP Status | Error Code | Error Text | Meaning |
|---|---|---|---|
| 400 Bad Request | 1903 | Missing required parameter(s):... | The API request did not contain one or more parameters which are required. |
| 400 Bad Request | 1904 | Please specify only one of these parameters:... | The API request contained 2 or more parameters from a group from which at most one may be specified. |
| 400 Bad Request | 1905 | parameter ... has invalid value ... | The API request contained a valid parameter specified with an invalid value. |
| 400 Bad Request | 1907 | The following combination of key=value pairs is not supported:... | The API request contained an invalid or unsupported combination of parameters. Invalid value for following param. autoGenerateCSR: true and encodedCSR is not null. |
| 400 Bad Request | 140001 | Malformed json | The json request is not properly formed. |
| 400 (Bad Request) | 140002 | Field is not editable | The requested field can not be edited. |
| 400 (Bad Request) | 140004 | Enrollment is not supported for CA | Enrollment/renewal of certificates by the specified CA is currently not supported. |
| 400 (Bad Request) | 140005 | API key is not configured | Incorrect API details, please verify the API key in the Configuration tab. |
| 400 (Bad Request) | 140006 | Invalid renewal certificate | Renewal failed due to one of the following reasons:<br><br>- certificate not found in inventory<br>- certificate is not a leaf certificate<br>- certificate is already in the process of being renewed<br>- certificate is not going to expire in next 60 days |
| 400 (Bad Request) | 140007 | Certificate order type is not editable | Cannot change an enrollment request to renewal request or vice versa |
| 403(Forbidden) | 2012 | User license is not authorized to run this API. | The API request failed because the user's subscription does not have API access enabled. |
| 403(Forbidden) | 148100 | User does not have required permissions | The API request failed because the user does not have the required permissions.Check user permissions in Admin module |

| HTTP Status | Error Code | Error Text | Meaning |
|---|---|---|---|
| 403(Forbidden) | 148101 | User has exhausted the allocated number of licenses | The API request failed because the order exceeds the allocated license count.Contact your Technical Account Manager for additional licenses |
| 404(Not Found) | 148200 | Invalid certificate order | Verify the order id |
| 409 Conflict | 1920 | API resource is not editable | Certificate request can not be updated once it is in the POSTED status. |
| 400 Bad Request | 1922 | Please specify at least one of the following parameters:... | The API request was missing some required information (but not necessarily a single specific parameter). |